

Trabajo Fin de Grado
Grado en Ingeniería Aeroespacial

Fundamentos de programación de VBA en Catia V5:
Aplicación al diseño paramétrico de perfiles y alas
NACA de 4 dígitos

Autor: Francisco Javier Jiménez Calvo

Tutora: Cristina Torrecillas Lozano

Depto. de Ingeniería Gráfica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2015



Trabajo Fin de Grado
Grado en Ingeniería Aeroespacial

**Fundamentos de programación de VBA en Catia
V5: Aplicación al diseño paramétrico de perfiles y
alas NACA de 4 dígitos**

Autor:

Francisco Javier Jiménez Calvo

Tutora:

Cristina Torrecillas Lozano

Contratada Doctora

Depto. de Ingeniería Gráfica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2015

Trabajo Fin de Grado: Fundamentos de programación de VBA en Catia V5: Aplicación al diseño paramétrico de perfiles y alas NACA de 4 dígitos

Autor: Francisco Javier Jiménez Calvo

Tutora: Cristina Torrecillas Lozano

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2015

El Secretario del Tribunal

A mis maestros y amigos

A mi familia

Agradecimientos

Me gustaría agradecer en primer lugar a mi tutora del proyecto, Cristina Torrecillas, por darme la oportunidad de poder realizar este trabajo con ella y por ayudarme durante todo el transcurso del mismo.

En segundo lugar quiero dar las gracias a una gran amiga, María del Carmen Rodríguez Suárez, por aguantar mis quejas diaramente y darme siempre animos. Gracias por tu ayuda amiga.

Y por último y más importante me gustaría mencionar a mis padres, Silverio y Lola, ya que si no fuese por ellos, nunca hubiese sido posible llegar a este momento de mi vida. No tengo palabras para agradecer todo lo que han luchado para poder proporcionarme un futuro. Muchísimas gracias.

Francisco Javier Jiménez Calvo

Sevilla, 2015

Resumen

El contenido del presente proyecto estudia alguno de los posibles campos de aplicación del lenguaje de macros de *Microsoft, Visual Basic for Applications* (VBA), al programa de diseño informático asistido por ordenador CATIA V5. En concreto éste Trabajo Fin de Grado está enfocado en el diseño de una aplicación para recrear en CATIA perfiles y alas parametrizadas que utilizan la nomenclatura NACA de 4 dígitos.

En primer lugar se realiza una introducción a la familia de perfiles NACA de 4 dígitos y al módulo de diseño paramétrico *Knowledgeware* de CATIA. Además, se definirá una descripción detallada del proceso seguido para conseguir parametrizar los perfiles y las alas.

En segundo lugar, se define qué es el VBA además de explicar cómo se accede desde CATIA V5, el entorno de éste, el modo en que el usuario puede desarrollar interfaces para interactuar con el programa así como los objetos más importantes del módulo *Mechanical Design* (*Sketcher, Part Design, Wireframe and Surface Design* y *Knowledgeware*) y las herramientas que ofrecen para trabajar.

Posteriormente, se desarrolla el núcleo principal de este trabajo que es el diseño de una aplicación para elaborar perfiles y alas parametrizadas con nomenclatura NACA de 4 dígitos a partir de diferentes parámetros que definen la geometría de los perfiles y las alas.

Para terminar, se ofrece el manual de usuario donde se explica a la persona que quiera hacer uso de la aplicación cómo interactuar con la interfaz para obtener lo que necesite. Además, al ser un diseño parametrizado, se realiza algunos comentarios sobre el uso después de la aplicación.

Abstract

The contents of this project is studying some of the possible fields of application of macro language *Microsoft, Visual Basic for Applications* (VBA), computer design program CATIA V5 computer assisted. In particular this Final Project is focused on the design of an application to recreate in CATIA profiles and wings parameterized using NACA 4 digit nomenclature.

In the first time, an introduction is made to the family of 4 digit NACA profiles and parametric design module of CATIA Knowledgeware. In addition, a detailed description of the process followed for parameterized profiles and wings will be defined.

In the second time, it defines what is the VBA also explains how it is accessed from CATIA V5, the setting of this, the way in which the user can develop interfaces to interactuar with the program and the most important objects of the module *Mechanical Design* (*Sketcher, Part Design, Wireframe and Surface Design and Knowledgeware*) and tools that offer to work.

After, the core of this work is nothing more nor less than the design of an application for profiling and parameterized wings NACA 4 digit nomenclature based on different parameters that define the geometry of the profiles and develops wings.

Finally, the manual where the person who wants to use the issue of how to interact with the interface you need to get explained is offered. Also, being a parameterized design, some comments on the use after the application is made.

Objetivos y alcance

El objetivo del presente proyecto es la interacción y manejo del lenguaje de programación de macros VBA para desarrollar una aplicación basado en dicha tecnología con la que poder obtener rápidamente en CATIA V5 perfiles NACA de 4 dígitos parametrizados y alas parametrizadas que usan la nomenclatura NACA de 4 dígitos en sus perfiles.

La idea de este trabajo surgió después de realizar numerosos trabajos de diseño de alas en los cuales se requería el uso tanto de alas sólidas como en superficie, ya que dependiendo del estudio que se quería realizar después del diseño geométrico, se necesitaba de una forma u otra. Además, durante todo el proceso de diseño, la geometría de las alas se veían continuamente sujetas a modificaciones por lo que se pensó que se podía implementar una geometría parametrizada para agilizar el trabajo de diseño permitiendo de esta forma realizar modificaciones en la geometría de forma instantánea.

Así, el alcance de este Trabajo Fin de Grado en Ingeniería Aeroespacial es familiarizarse y controlar tanto las herramientas de parametrización de CATIA como los conocimientos de programación necesarios para elaborar esta o cualquier otra aplicación y facilitar a cualquier usuario el acceso al software realizado para cumplir sus necesidades. De este modo, el alcance del proyecto no es un estudio intensivo de la geometría de perfiles y alas y tampoco la exploración de las posibles aplicaciones de éste.

Índice general

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Objetivos y alcance	xv
Índice general	xvii
Índice de tablas	xix
Índice de Figuras	xxi
Índice de Códigos	xxiii
Notación	xxv
1 Introducción	1
1.1 <i>Evolución histórica de la nomenclatura NACA de 4 dígitos</i>	1
1.2 <i>Perfiles aerodinámicos con espesor finito: familia NACA de 4 dígitos</i>	2
1.3 <i>Aviones que utilizan la nomenclatura NACA.</i>	5
1.4 <i>CATIA</i>	6
1.5 <i>Módulo Knowledge Advisor de CATIA</i>	7
1.5.1 Entorno	8
1.5.2 Parámetros	9
1.5.3 Fórmulas	11
1.5.4 Reglas	12
1.6 <i>Parametrización de un ala con perfil NACA de 4 dígitos en CATIA</i>	14
2 El entorno de programación VBA en CATIA V5	27
2.1 <i>Visual Basic for Application</i>	27
2.1.1 Entorno	28
2.1.2 Librerías de las macros	28
2.1.3 Macro Recording	29
2.2 <i>Visual Basic Editor</i>	30
2.3 <i>Iniciación a la programación</i>	32
2.3.1 Declaración de estamentos	33
2.3.2 Estamentos	34
2.3.3 Estamentos ejecutables	34
2.3.4 Funciones y subfunciones	34
2.3.5 Estructuras condicionales e iterativas	34
2.3.5.1 Condicionales	34
2.3.5.2 Iterativas	35
2.3.6 Objetos orientados a la programación	36
2.3.7 Cómo definir un objeto	36
2.4 <i>Interfaz con el usuario</i>	37
3 Herramientas de programación de perfiles y alas	41

3.1	<i>Arranque</i>	41
3.2	<i>SKETCHER</i>	44
3.2.1	Crear un punto	45
3.2.2	Crear una recta	46
3.2.3	Crear un Spline	47
3.3	<i>Restricciones</i>	48
3.3.1	CatCstTypeDistance	50
3.3.2	CatCstTypeOn	50
3.3.3	CatCstTypePerpendicularity	51
3.3.4	CatCstTypeAngle	51
3.3.5	CatCstTypeSymetry	51
3.4	<i>CATPART</i>	52
3.4.1	<i>PARTDESIGN</i>	52
3.4.1.1	Close Surface	56
3.4.1.2	Mirror	56
3.4.1.3	Symmetry	57
3.4.2	<i>WIREFRAME AND SURFACE DESIGN</i>	57
3.4.2.1	Fill	59
3.4.2.2	Multi-sections Surface	60
3.4.2.3	Join	61
3.4.2.4	Symmetry	61
3.5	<i>KNOWLEDGEWARE</i>	62
3.5.1	Parámetros	62
3.5.2	Fórmulas	65
3.5.3	Reglas	66
4	Aplicación: entorno de programación	69
4.1	<i>Parámetros de entrada en los UserForms</i>	71
4.2	<i>Estructura de la Aplicación</i>	72
4.3	<i>Estructura del Código</i>	73
5	Aplicación perfiles y alas con nomenclatura NACA de 4 dígitos: manual de usuario	79
5.1	<i>Uso posterior a la aplicación</i>	81
	Conclusiones	83
	Bibliografía	85
	Glosario	87
	ANEXOS	89
	<i>ANEXO 1. ACTIVACIÓN DE PARÁMETROS Y RELACIONES EN CATIA</i>	89
	<i>ANEXO 2. CÓDIGO PARA DIBUJAR UN PERFIL NACA DE 4 DÍGITOS</i>	92
	<i>ANEXO 3. CÓDIGO DE LA ESTRUCTURA PRINCIPAL PARA DIBUJAR UN ALA</i>	126
	<i>ANEXO 4. CÓDIGO ALA IZQUIERDA SUPERFICIE</i>	198
	<i>ANEXO 5. CÓDIGO ALA DERECHA SUPERFICIE</i>	200
	<i>ANEXO 6. CÓDIGO ALAS SUPERFICIES</i>	203
	<i>ANEXO 7. CÓDIGO ALA IZQUIERDA SÓLIDA</i>	206
	<i>ANEXO 8. CÓDIGO ALA DERECHA SÓLIDA</i>	208
	<i>ANEXO 9. CÓDIGO ALAS SÓLIDAS</i>	211

Índice de tablas

Tabla 1. Herramientas del módulo <i>Knowledge Advisor</i>	8
Tabla 2. Conceptos básicos de la programación	33
Tabla 3. Diferencias entre clase y objeto	37
Tabla 4. Opciones de la barra de herramientas	38
Tabla 5. Objetos del módulo <i>Sketcher</i>	45
Tabla 6. Restricciones	49
Tabla 7. Objetos del <i>PartDesign</i>	54
Tabla 8. Herramientas del <i>ShapeFactory</i>	55
Tabla 9. Herramientas del <i>HybridShapefactory</i>	58
Tabla 10. Herramientas de superficie	58
Tabla 11. Estructura jerarquizada del programa	72

Índice de Figuras

Figura 1. Emblema NACA	2
Figura 2. Definición Geométrica Perfil NACA. Referencia [2]	3
Figura 3. Cessna 150	5
Figura 4. Fairchild A-10	6
Figura 5. Herramientas del módulo <i>Knowledge Advisor</i>	8
Figura 6. Orden <i>Formula</i>	9
Figura 7. Cuadro de diálogo orden <i>Formula</i>	10
Figura 8. Rama <i>Parameters</i>	10
Figura 9. Cuadro de edición <i>Constraint Definition</i>	11
Figura 10. Cuadro de edición <i>Formula Editor</i>	11
Figura 11. Cuadro de edición <i>Constraint Definition</i> editado	12
Figura 12. Rama <i>Relations</i>	12
Figura 13. Cuadro de diálogo <i>Rule Editor</i>	13
Figura 14. Cuadro de edición <i>Rule Editor</i>	13
Figura 15. <i>Constraint</i> punto borde de ataque perfil encastre	15
Figura 16. Líneas de referencia perfil encastre	16
Figura 17. <i>Constraints Sketch.1</i>	16
Figura 18. Perfil NACA con incidencia nula	18
Figura 19. Perfil NACA con ángulo de incidencia	19
Figura 20. Vista frontal ala, ángulo de diedro	19
Figura 21. Vista en planta ala, ángulo de flecha	19
Figura 22. <i>Constraint</i> punto borde de ataque perfil punta ala	20
Figura 23. Líneas de referencia del perfil de la punta del ala	20
Figura 24. Estrechamiento	23
Figura 25. Perfiles distintos	23
Figura 26. Incidencia y Torsión	23
Figura 27. Diedro y Flecha	24
Figura 28. Perfiles y línea de borde de ataque	24
Figura 29. Ejemplo resultado final del diseño de un ala cualquiera	25
Figura 30. Acceso a macros	28
Figura 31. Entorno VBA	30

Figura 32. Ventana <i>Project Explorer</i> (Rojo) y <i>Properties Window</i> (Amarillo)	30
Figura 33. Ventana <i>Object Browser</i>	31
Figura 34. Formulario de trabajo	37
Figura 35. Herramientas de diseño	37
Figura 36. Editor	39
Figura 37. Comandos para hacer funcionar la aplicación	39
Figura 38. Estructura del módulo <i>Sketcher</i>	44
Figura 39. Estructura del módulo <i>PartDesign</i>	53
Figura 40. Estructura interna del <i>PartDocument</i>	54
Figura 41. Icono <i>HybridBody</i>	59
Figura 42. Esquema de parámetros	63
Figura 43. Esquema <i>Relations</i>	65
Figura 44. Vista del módulo <i>Relations</i> en el árbol	67
Figura 45. <i>Rule Editor</i>	68
Figura 46. Interfaz principal del programa	69
Figura 47. Flujo <i>UserForms</i>	70
Figura 48. Parámetros formulario Crear Perfil	71
Figura 49. Parámetros formulario Crear Ala	71
Figura 50. Mensaje de error	72
Figura 51. Ejemplo del resultado final de crear un perfil	73
Figura 52. Resultado de ejecutar el ANEXO 3	74
Figura 53. Ejemplo del resultado final de crear una semiala izquierda	74
Figura 54. Ejemplo del resultado final de crear una semiala derecha	75
Figura 55. Ejemplo del resultado final de crear un ala	76
Figura 56. Diagrama de flujo de la aplicación	78
Figura 57. Interfaz principal de la aplicación	80
Figura 58. Interfaz PerfilesNACA4Digitos	80
Figura 59. Interfaz AlaNACA4Digitos	81
Figura 60. Botón actualizar de CATIA	82
Figura 61. Localización <i>Options</i>	89
Figura 62. Configuración <i>Tree Customization</i>	90
Figura 63. Configuración <i>Part Infrastructure, General</i>	90
Figura 64. Configuración <i>Part Infrastructure, Display</i>	91
Figura 65. Configuración <i>Parameters and Measure</i>	91

Índice de Códigos

Código 1. Cotas en longitud de los puntos del extradós del perfil del encastre	17
Código 2. Cotas en altura de los puntos del extradós del perfil del encastre	17
Código 3. Cotas en longitud de los puntos del intradós del perfil del encastre	18
Código 4. Cotas en altura de los puntos del intradós del perfil del encastre	18
Código 5. Cotas en altura de los puntos del extradós del perfil de la punta del ala	21
Código 6. Cotas en longitud de los puntos del extradós del perfil de la punta del ala	21
Código 7. Cotas en longitud de los puntos del intradós del perfil de la punta del ala	22
Código 8. Cotas en altura de los puntos del intradós del perfil de la punta del ala	22
Código 9. Declaración de estamento	33
Código 10. Estamento	34
Código 11. Estamento ejecutable	34
Código 12. Subfunción	34
Código 13. Estructura condicional	35
Código 14. Estructura <i>For ... Next</i>	35
Código 15. Estructura <i>While ... Wend</i>	35
Código 16. Estructura <i>Do ... Loop</i>	36
Código 17. Arranque de documentos	42
Código 18. Árbol de trabajo	42
Código 19. <i>Sketches</i>	42
Código 20. Vector de coordenadas	43
Código 21. Establecimiento lugar de trabajo	43
Código 22. Objeto <i>Factory 2D</i>	44
Código 23. Creación de un punto	46
Código 24. Creación de una recta	46
Código 25. Línea de construcción	47
Código 26. Creación de un punto de control	47
Código 27. Creación de un spline	48
Código 28. Creación constraints de forma general	48
Código 29. Modo <i>Constraint</i>	49
Código 30. Modo <i>Measurement</i>	49
Código 31. <i>CatCstTypeDistance</i>	50

Código 32. <i>CatCstTypeOn</i>	51
Código 33. <i>CatCstTypePerpendicularity</i>	51
Código 34. <i>CatCstTypeAngle</i>	51
Código 35. <i>CatCstTypeSymmetry</i>	52
Código 36. <i>Close Surface</i>	56
Código 37. <i>Mirror</i>	56
Código 38. <i>Symmetry</i>	57
Código 39. Generar un cuerpo híbrido	59
Código 40. Herramienta <i>HybridShapeFill</i>	59
Código 41. <i>Multi-sections Surface</i>	60
Código 42. Unión de dos superficies	61
Código 43. Añadir superficies al <i>Join</i>	61
Código 44. <i>Symmetry</i>	62
Código 45. <i>Parameters</i>	62
Código 46. Creación de parámetros	64
Código 47. Asignación de un valor a un parámetro	64
Código 48. Comprobación del valor de los parámetros	64
Código 49. <i>Relations</i>	65
Código 50. Crear fórmula	66
Código 51. Crear una <i>Rule</i>	67
Código 52. Orden <i>Modify</i>	67
Código 53. Llamada a los <i>UserForms</i> con un <i>CommandButton</i>	70
Código 54. Salir del programa con un <i>CommandButton</i>	70

Notación

L	Fuerza de sustentación
α	Ángulo de ataque
x_U	Posición en el eje horizontal de los puntos del perfil en el extradós
y_U	Posición en el eje vertical de los puntos del perfil en el extradós
x_L	Posición en el eje horizontal de los puntos del perfil en el intradós
y_L	Posición en el eje vertical de los puntos del perfil en el intradós
x	Variable independiente en la dirección del eje horizontal que va de 0 a cuerda
t	Dos últimos dígitos del perfil NACA partidos por 100
c	Cuerda del perfil
m	Primer dígito del perfil NACA partido por 100
p	Segundo dígito del perfil NACA partido por 10
sen	Función seno
cos	Función coseno
arctan	Función arcotangente
atan	Función arcotangente
\leq	Menor o igual
\leqslant	Menor o igual
\geq	Mayor o igual
\geqslant	Mayor o igual
$**$	Función exponente

1 INTRODUCCIÓN

Piensa, cree, sueña y atrévete.

- Walt Disney -

Según la Real Academia Española (RAE), la palabra ala tiene la siguiente definición:

“Cada una de las partes que a ambos lados del avión presentan al aire una superficie plana y sirven para sustentar el aparato en vuelo.”

El ala es un elemento estructural de la aeronave que proporciona la fuerza de sustentación, L , necesaria para poder volar (*Referencia [1]*). Dichas alas, no pueden tener un perfil cualquiera ya que debido a la geometría del perfil somos capaces de generar la fuerza de sustentación. Aquí es donde entra en juego la nomenclatura NACA de 4 dígitos que es un tipo de perfil muy usado históricamente y que aún continúa siéndolo.

1.1 Evolución histórica de la nomenclatura NACA de 4 dígitos

Antes de la era de la aerodinámica computacional, la industria aeronáutica dependía de los estudios realizados por dos agencias a nivel mundial para la selección de perfiles aerodinámicos en sus diseños. En el Reino Unido estaba la RAE (*Royal Aircraft Establishment*) y en Estados Unidos la institución precursora del desarrollo racional de perfiles alares denominada NACA (*National Advisory Committee for Aeronautics*). En la Figura 1 se puede ver el emblema de NACA.



Figura 1. Emblema NACA

Este comité estaba encargado de realizar pruebas y diseños empíricos de perfiles alares. La NACA fue una agencia federal de Estados Unidos que se fundó el 3 de marzo de 1915 para emprender, fomentar e institucionalizar las investigaciones aeronáuticas. La agencia se disolvió el 1 de octubre de 1958 para dar paso a la creación de la NASA (*National Aeronautics and Space Administration*).

Estas dos instituciones a nivel mundial formalizaron trabajos y estudios en túneles de viento durante el periodo comprendido entre 1920 y 1960. Como resultado de este trabajo surgió la creación de familias de perfiles alares y el desarrollo racional de una nomenclatura para la elección de perfiles basado en su aplicación posterior. El estudio tiene tanta trascendencia e importancia que muchos de los perfiles siguen siendo utilizados en aeronaves actuales.

Cabe resaltar que antes de 1930 los diseñadores no tenían ningún tipo de herramienta para seleccionar perfiles alares, por este motivo, el diseño y elección no obedecía a un análisis dimensionado.

Esta situación cambio radicalmente cuando NACA realizó un estudio racional y sistemático para el diseño de los perfiles alares aeronáuticos. Este estudio se basó en pruebas exhaustivas en túneles de viento para el análisis de las características y propiedades inherentes de cada perfil alar. El estudio requería de una nomenclatura efectiva para lograr su clasificación de manera ordenada y detallada con el fin de suministrar al diseñador una validación completa del perfil alar acorde a su misión. *Referencia [2]*

Las superficies de sustentación de NACA son todavía utiles en muchas aplicaciones de diseño. Además, proporciona un medio para obtener una geometría final, que no tiene por que ser NACA, para obtener el rendimiento deseado en nuestra superficie sustentadora, es decir, como un primer paso y/o paso intermedio durante el proceso de diseño. *Referencia [3]*

1.2 Perfiles aerodinámicos con espesor finito: familia NACA de 4 dígitos

NACA decidió generar una nomenclatura para la primera familia de perfiles alares a principios de 1930. La primera familia de los NACA *airfoils* se conoce como la familia NACA de 4 dígitos. Esta familia de perfiles se divide en dos tipos:

1. Perfiles simétricos. La línea media del perfil es una recta que une el borde de ataque con el borde de salida .

2. Perfiles con curvatura. La línea media del perfil está constituida por dos parábolas entre sí.

La razón de por qué se utilizan perfiles con curvatura es porque el valor de la fuerza de sustentación es mayor que en el caso de perfiles simétricos para un ángulo de ataque, α , dado. *Referencia [1]*

La línea de curvatura media es la característica más importante del diseño de un perfil alar. Ésta se define mediante puntos posicionados de forma equidistante entre el extradós y el intradós medidos perpendicularmente desde la cuerda.

El punto de inicio y el punto de finalización de la línea de curvatura media están definidos como el borde de ataque y el borde de salida. La línea recta que conecta los puntos anteriores es llamada como la línea de cuerda del perfil alar y la distancia precisa medida en esta línea se conoce como cuerda. Todas estas características de las que se habla se pueden ver en la Figura 2. *Referencia [3]*

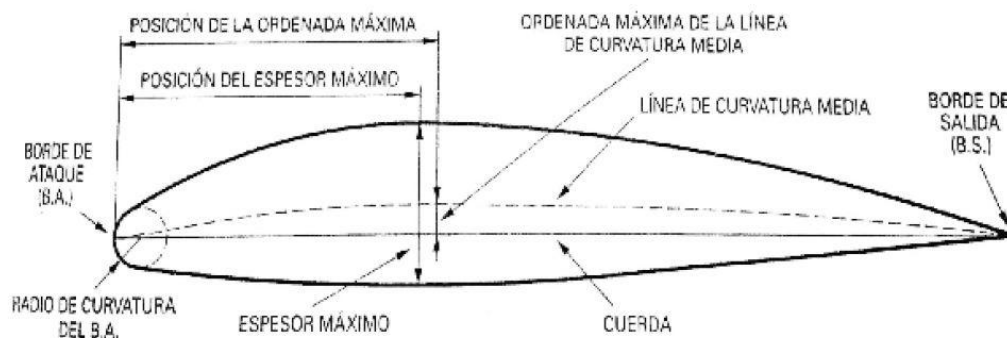


Figura 2. Definición Geométrica Perfil NACA. *Referencia [2]*

El significado de cada uno de los cuatro dígitos que definen un perfil NACA de 4 dígitos empezando por el de más a la izquierda, utilizando como ejemplo el perfil NACA 6410, es el siguiente (*Referencia [1]* y [4]):

- El primer dígito, 6, indica la máxima cota de la línea media del perfil en unidades de tanto por ciento de la cuerda. El valor de este máximo recime el nombre de máxima curvatura del perfil.
- El segundo dígito, 4, indica la posición del punto de máxima curvatura con respecto al borde de ataque en unidades de decenas de tanto por ciento de la cuerda.
- Los dos últimos dígitos, 10, definen el máximo espesor del perfil en unidades de tanto por ciento de la cuerda.

Por último, comentar que la distribución del espesor del perfil alar tiene efectos aerodinámicos importantes relacionados con el coeficiente máximo de sustentación y la capa límite. Al incrementar el espesor, el coeficiente de sustentación máximo aumenta y si disminuimos el espesor, ocurre el efecto contrario. Además existe un fenómeno de separación de la capa límite, denominado desprendimiento, asociado al incremento del espesor, ya que se incrementa el gradiente adverso de presiones que debe soportar el flujo alrededor del perfil. Por este motivo, es necesario tener en cuenta hasta qué punto es bueno incrementar el espesor del perfil con el objetivo de incrementar el coeficiente de sustentación máximo y de igual forma minimizar los efectos

de desprendimiento de la capa límite. *Referencia [2]*

Un perfil NACA de 4 dígitos viene definido por las ecuaciones (1) y (2):

$$x_U = x - y_t \text{sen}\theta, \quad y_U = y_c + y_t \text{cos}\theta \quad (1)$$

$$x_L = x + y_t \text{sen}\theta, \quad y_L = y_c - y_t \text{cos}\theta \quad (2)$$

Donde x_U es la posición en el eje horizontal de los puntos del perfil en el extradós, y_U es la posición en el eje vertical de los puntos del perfil en el extradós, x_L es la posición en el eje horizontal de los puntos del perfil en el intradós, y_L es la posición en el eje vertical de los puntos del perfil en el intradós, x es una variable independiente en la dirección del eje horizontal que va de 0 a la cuerda, y_t que viene dado por la ecuación (3), θ que viene dado por la ecuación (5) e y_c que viene dado por la ecuación (4).

$$y_t = \frac{t}{0.2} c \left[0.2969 \sqrt{\frac{x}{c}} - 0.1260 \frac{x}{c} - 0.3516 \left(\frac{x}{c}\right)^2 + 0.2843 \left(\frac{x}{c}\right)^3 - 0.1015 \left(\frac{x}{c}\right)^4 \right] \quad (3)$$

donde t es igual a los dos últimos dígitos del perfil partidos por 100 y c es la cuerda.

$$y_c = \begin{cases} m \frac{x}{p^2} \left(2p - \frac{x}{c}\right) & 0 \leq x \leq pc \\ m \frac{c-x}{(1-p)^2} \left(1 + \frac{x}{c} - 2p\right) & pc \leq x \leq c \end{cases} \quad (4)$$

donde m es el primer dígito partido por 100 y p es el segundo dígito partido por 10.

$$\theta = \arctan\left(\frac{dy_c}{dx}\right) \quad (5)$$

Por lo tanto, las expresiones finales que definen un perfil NACA de 4 dígitos son las correspondientes a las ecuaciones (6), (7), (8) y

(9).

$$x_U = \begin{cases} x - \frac{t}{0.2} c \left[0.2969 \sqrt{\frac{x}{c}} - 0.1260 \frac{x}{c} - 0.3516 \left(\frac{x}{c}\right)^2 + 0.2843 \left(\frac{x}{c}\right)^3 - 0.1015 \left(\frac{x}{c}\right)^4 \right] \text{sen} \left[\arctan\left(\frac{2m}{p^2} \left(p - \frac{x}{c}\right)\right) \right] & 0 \leq x \leq pc \\ x - \frac{t}{0.2} c \left[0.2969 \sqrt{\frac{x}{c}} - 0.1260 \frac{x}{c} - 0.3516 \left(\frac{x}{c}\right)^2 + 0.2843 \left(\frac{x}{c}\right)^3 - 0.1015 \left(\frac{x}{c}\right)^4 \right] \text{sen} \left[\arctan\left(\frac{2m}{(1-p)^2} \left(p - \frac{x}{c}\right)\right) \right] & pc \leq x \leq c \end{cases} \quad (6)$$

$$y_U = \begin{cases} m \frac{x}{p^2} \left(2p - \frac{x}{c}\right) + \frac{t}{0.2} c \left[0.2969 \sqrt{\frac{x}{c}} - 0.1260 \frac{x}{c} - 0.3516 \left(\frac{x}{c}\right)^2 + 0.2843 \left(\frac{x}{c}\right)^3 - 0.1015 \left(\frac{x}{c}\right)^4 \right] \text{cos} \left[\arctan\left(\frac{2m}{p^2} \left(p - \frac{x}{c}\right)\right) \right] & 0 \leq x \leq pc \\ m \frac{c-x}{(1-p)^2} \left(1 + \frac{x}{c} - 2p\right) + \frac{t}{0.2} c \left[0.2969 \sqrt{\frac{x}{c}} - 0.1260 \frac{x}{c} - 0.3516 \left(\frac{x}{c}\right)^2 + 0.2843 \left(\frac{x}{c}\right)^3 - 0.1015 \left(\frac{x}{c}\right)^4 \right] \text{cos} \left[\arctan\left(\frac{2m}{(1-p)^2} \left(p - \frac{x}{c}\right)\right) \right] & pc \leq x \leq c \end{cases} \quad (7)$$

$$x_L = \begin{cases} x + \frac{t}{0.2}c \left[0.2969\sqrt{\frac{x}{c}} - 0.1260\frac{x}{c} - 0.3516\left(\frac{x}{c}\right)^2 + 0.2843\left(\frac{x}{c}\right)^3 - 0.1015\left(\frac{x}{c}\right)^4 \right] \text{sen} \left[\arctan \left(\frac{2m}{p^2} \left(p - \frac{x}{c} \right) \right) \right] & 0 \leq x \leq pc \\ x + \frac{t}{0.2}c \left[0.2969\sqrt{\frac{x}{c}} - 0.1260\frac{x}{c} - 0.3516\left(\frac{x}{c}\right)^2 + 0.2843\left(\frac{x}{c}\right)^3 - 0.1015\left(\frac{x}{c}\right)^4 \right] \text{sen} \left[\arctan \left(\frac{2m}{(1-p)^2} \left(p - \frac{x}{c} \right) \right) \right] & pc \leq x \leq c \end{cases} \quad (8)$$

$$y_L = \begin{cases} m \frac{x}{p^2} \left(2p - \frac{x}{c} \right) - \frac{t}{0.2}c \left[0.2969\sqrt{\frac{x}{c}} - 0.1260\frac{x}{c} - 0.3516\left(\frac{x}{c}\right)^2 + 0.2843\left(\frac{x}{c}\right)^3 - 0.1015\left(\frac{x}{c}\right)^4 \right] \text{cos} \left[\arctan \left(\frac{2m}{p^2} \left(p - \frac{x}{c} \right) \right) \right] & 0 \leq x \leq pc \\ m \frac{c-x}{(1-p)^2} \left(1 + \frac{x}{c} - 2p \right) - \frac{t}{0.2}c \left[0.2969\sqrt{\frac{x}{c}} - 0.1260\frac{x}{c} - 0.3516\left(\frac{x}{c}\right)^2 + 0.2843\left(\frac{x}{c}\right)^3 - 0.1015\left(\frac{x}{c}\right)^4 \right] \text{cos} \left[\arctan \left(\frac{2m}{(1-p)^2} \left(p - \frac{x}{c} \right) \right) \right] & pc \leq x \leq c \end{cases} \quad (9)$$

Estas ecuaciones son las que se programan en CATIA haciendo uso del módulo *Knowledge Advisor*. Dicha implementación se explicará en apartados posteriores. *Referencia [5]*

1.3 Aviones que utilizan la nomenclatura NACA.

Muchas de las aeronaves actuales utilizan perfiles alares NACA de todas las familias y divisiones. Estos se han beneficiado de los estudios realizados en los distintos perfiles alares por NACA donde se pueden encontrar estudios de perfiles para reducir la resistencia lo máximo posible o para incrementar el coeficiente de sustentación lo máximo posible entre otros. A modo ilustrativo, a continuación se muestra una serie de aviones que utilizan la nomenclatura NACA en sus alas (*Referencia [2]*):

- Cessna 150: utiliza el perfil NACA 2412.



Figura 3. Cessna 150

- Fairchild A-10: utiliza el perfil NACA 6716.



Figura 4. Fairchild A-10

1.4 CATIA

CATIA (*Computer-Aided Three dimensional Interactive Application*) es mucho más que un software de diseño, es una herramienta de integración del proceso desarrollo del producto realizado por Dassault Systèmes. CATIA es la solución líder en todo el mundo para la experiencia y el diseño de productos. Organizaciones líderes de distintos sectores la utilizan para desarrollar los productos que vemos y usamos en nuestra vida cotidiana.

Actualmente cuenta con más de ciento cincuenta módulos de las más diversas aplicaciones y ofrece la posibilidad única no solo de modelar cualquier producto, sino de hacerlo en el contexto de su comportamiento en la vida real: *diseño en la era de la experiencia*. Los arquitectos de sistemas, los ingenieros, los diseñadores y todos sus colaboradores pueden definir el mundo que nos conecta, imaginarlo y darle forma. Con CATIA se puede diseñar cualquier cosa por muy compleja que pueda ser. Sólo hay que encontrar la mejor forma de hacerlo. Además, las posibilidades en cuanto a desarrollo paramétrico sorprenden, sin olvidar la extremada precisión de trabajo, y a pesar de lo complejo que es, su sencillez de uso, se agradece.

CATIA tiene más de 140 módulos o soluciones de proceso. Cada solución engloba un conjunto de herramientas específicas según la aplicación que se necesite. Y cada solución funciona bajo una licencia de uso concreta. CATIA es muy potente y ofrece múltiples soluciones que se distribuyen bajo sus correspondientes licencias de uso. Se basa en la plataforma 3DEXPERIENCE de Dassault Systèmes, ofrece:

- Un entorno de diseño social en una fuente única de autenticidad, al que se accede mediante potentes paneles en 3D que impulsan la inteligencia empresarial, el diseño simultáneo en tiempo real y la

colaboración de todas las partes interesadas, incluidos los trabajadores.

- 3DEXPERIENCE ofrece una experiencia intuitiva con funcionalidades de modelado y simulación en 3D de primer nivel que optimizan la eficacia de todos los usuarios.
- Se trata de una plataforma inclusiva de desarrollo de productos, que resulta fácil de integrar con los procesos y las herramientas existentes. Esto permite que varias disciplinas aprovechen las eficaces e integradas aplicaciones especializadas en todas las fases del proceso de desarrollo de los productos.

Las aplicaciones de diseño, ingeniería e ingeniería de sistemas de CATIA constituyen el núcleo de las soluciones basadas en experiencias de *Dassault Systèmes* destinadas a cubrir necesidades específicas de cada sector. Esto supone una revolución en la manera de concebir, desarrollar y hacer realidad productos de las organizaciones, cuyas innovadoras experiencias del cliente, además, aportan una ventaja frente a la competencia. *Referencia [6] y [7]*

1.5 Módulo Knowledge Advisor de CATIA

Tanto para la parametrización de los perfiles NACA de 4 dígitos como de las alas es necesaria la utilización del módulo *Knowledge Advisor* de CATIA. Este módulo permite a los usuarios incluir el conocimiento dentro de un plan de trabajo e influenciar en él para ayudar al diseñador a tomar decisiones en el orden de reducir errores o automatizar dicho plan, para poder obtener la máxima productividad.

Los usuarios pueden aportar conocimiento en un plan de diseño tales como conductas, fórmulas, reglas y chequeos y acudir a ellos cuando se requiera en cualquier momento.

Durante la ejecución de este Trabajo Fin de Grado se hizo uso continuamente de fórmulas y reglas para que el usuario final de la aplicación pudiese, una vez ejecutada la aplicación, desentenderse de ésta y realizar todas las modificaciones que se requieran con tan solo dar un nuevo valor a los parámetros de la geometría.

De forma abreviada, el módulo *Knowledge Advisor* ayuda a:

- Capturar el conocimiento de la ingeniería como especificaciones que permiten dar a la aplicación una consistencia completa.
- Definir y compartir fácilmente la habilidad de diseño entre todos los usuarios.
- Automatizar la definición del producto.
- Aumentar la productividad.
- Permitir dar una atención temprana a especificaciones del plan de diseño que previenen un costoso rediseño.
- Guiar y ayudar a los usuarios a través de sus tareas del plan de diseño.

Referencia [8]

1.5.1 Entorno

Dentro del módulo *Knowledge Advisor* podemos encontrar todas las herramientas que visualizamos en la Figura 5.



Figura 5. Herramientas del módulo *Knowledge Advisor*

En la Tabla 1 se detallan las diferentes herramientas así como una breve explicación de para que se utilizan.










Símbolo	Etiqueta	Descripción
	<i>Formula</i>	Se utiliza para crear parámetros nuevos y relaciones entre ellos.
	<i>Design Table</i>	Proporcionan la capacidad de manejar familias de componentes.
	<i>Knowledge Inspector</i>	Pregunta designaciones para determinar y preveer los resultados al cambiar algunos parámetros.
	<i>Rule</i>	Permite crear reglas, escribir códigos y textos con sintaxis y aplicarlas a nuestro documento.
	<i>Check</i>	Al igual que <i>Rule</i> se crean con un código y se aplican al documento.
	<i>Reactions</i>	Permite crear una escritura específica cómo cambiar algún rasgo que atribuye cuando un evento ocurre.
	<i>Add Set of Parameters</i>	Permite crear juegos de parámetros. Estos juegos de parámetros son agrupados debajo del nodo de los parámetros.
	<i>Macros with arguments</i>	Permite que se lancen unos argumentos de ingenio Donde se ha creado una macro.
	<i>Actions</i>	Ermite crear una escritura para la macro.
	<i>List</i>	Permite crear una lista de rasgos que se localizará bajo el nodo de los parámetros en el árbol de la especificación.

Tabla 1. Herramientas del módulo *Knowledge Advisor*

En este Trabajo Fin de Grado solamente utilizaremos las herramientas “*Formula*” tanto para la creación de parámetros como de fórmulas y “*Rule*” para la programación de las fórmulas del perfil NACA de 4 dígitos ya que dichas fórmulas tienen diferentes casos dependiendo de la posición de la cuerda en la que nos

encontremos.

Se recomienda que antes de empezar a trabajar con estas herramientas se tenga activada la visualización de los parámetros y las relaciones, si aún no lo está, en el ANEXO 1 viene detallado paso a paso como realizarla. A continuación, para una mejor comprensión de este módulo, se procede a explicar las diferentes herramientas utilizadas en la ejecución del Trabajo. *Referencia [8]*

1.5.2 Parámetros

En CATIA, los parámetros son elementos que define un valor numérico. Estos elementos no son elementos geométricos, sino un almacenen de valores de diferentes tipos que después pueden ser utilizados en distintas aplicaciones, es decir, no es más que un dato o valor que se toma para una determinada función y con criterios específicos.

Por ejemplo, en la antigua Grecia, con solo conocer un parámetro, la longitud de la cabeza de cualquier individuo, se podría reconstruir el cuerpo entero repetando unos cánones o proporciones observadas, aunque esto rápidamente fue descartado (por las infinitas diferencias entre cada cuerpo humano).

En la Ingeniería se siguen obteniendo proporciones basadas en análisis físicos y con ellas construyendo infinidad de cosas. A continuación se va a ver cómo se crea un parámetro.

En primer lugar se debe acceder a la orden *Formula* de la barra *Knowledge*, la ubicación de esta barra varia según el usuario ya que en CATIA cada persona personaliza su entorno de trabajo a su gusto. La orden que debemos buscar la podemos visualizar en la Figura 6.



Figura 6. Orden *Formula*

Al pinchar en la orden “Formula” aparece un cuadro de diálogo como el de la Figura 7. Una vez desplegado, se debe realizar tres sencillos pasos para crear un parámetro.

1. Primero: definir el tipo de unidades del parámetro a crear. Esto es importante, ya que los parámetros se utilizan en combinación con valores de elementos, y estos en general llevan asociadas unas unidades de trabajo.
2. Se establece el tipo de valores que contendrá, uno o múltiples.
3. Se crea el parámetro con la opción *New Parameter of Type*.

Los pasos se encuentran señalizados sobre la Figura 7. Para cambiar el nombre del parámetro, primero hay que crearlo. Una vez creado, se marca y renombra.

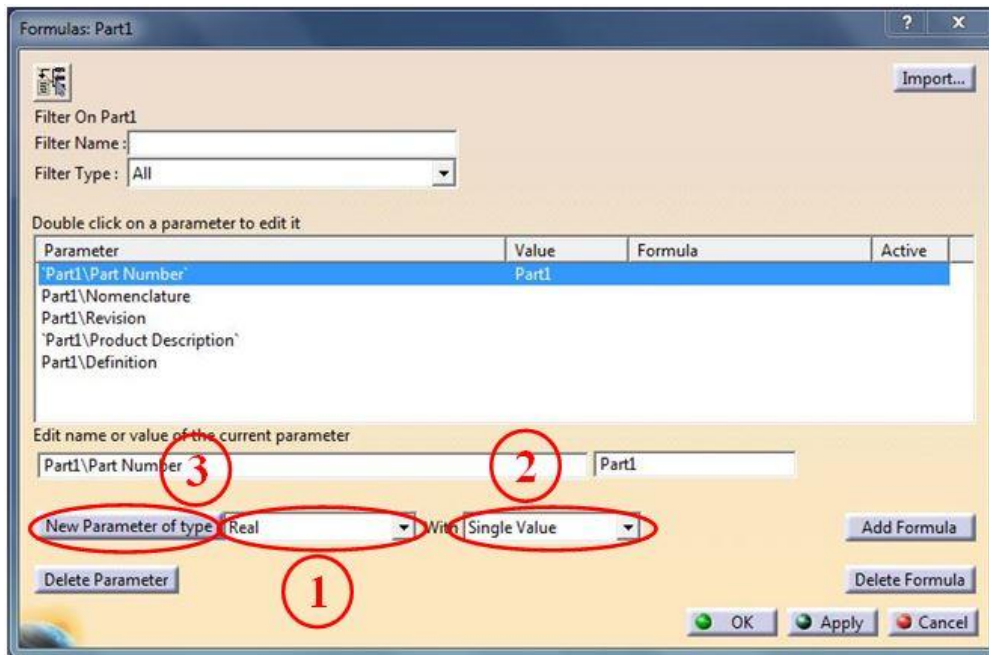


Figura 7. Cuadro de diálogo orden *Formula*

Una vez creado el parámetro, este se visualizan en el árbol agrupado bajo la rama *Parameters*. En la Figura 8 se puede ver los parámetros utilizados para parametrizar el ala completa usando la nomenclatura NACA de 4 dígitos. Además, podemos observar las unidades de cada parámetro en el caso de que este tenga asignada una.

Referencia [6]

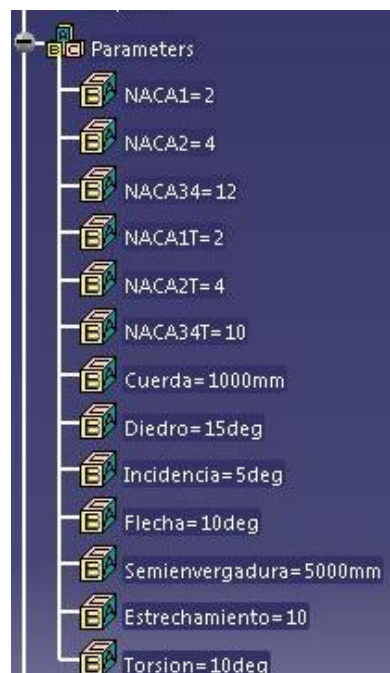


Figura 8. Rama *Parameters*

1.5.3 Fórmulas

Las fórmulas o también llamadas ecuaciones o relaciones, son una relación que se puede añadir para sincronizar o relacionar valores o parámetros entre sí, es decir, con ellas se puede llevar diferentes parámetros a otros nuevos, manteniendo entre ellos no sólo una proporción, sino una ecuación de relación.

En este Trabajo Fin de Grado se han utilizado fórmulas para que se modifiquen las *Constraint*, asociando éstas a uno o varios parámetros, de forma que al modificar algún parámetro, se actualice automáticamente el resto de la geometría con tan sólo un click.

Para crear una fórmula en primer lugar se debe crear una *Constraint* y una vez creada modificar su valor editando una fórmula para ello debemos hacer click sobre el botón derecho de nuestro ratón sobre el valor de la *Constraint*, desplegándose un menú donde aparecerá dicha opción tal y como se observa en la Figura 9.

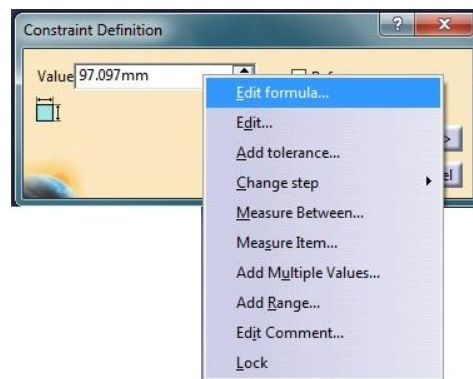


Figura 9. Cuadro de edición *Constraint Definition*

Una vez pinchemos en *Edit formula* nos aparece un nuevo cuadro de diálogo donde se puede escribir dicha fórmula. Este cuadro se puede visualizar en la Figura 10.

Hay que tener mucho cuidado cuando creamos una fórmula ya que las unidades de los parámetros que incluimos deben ser correctas.

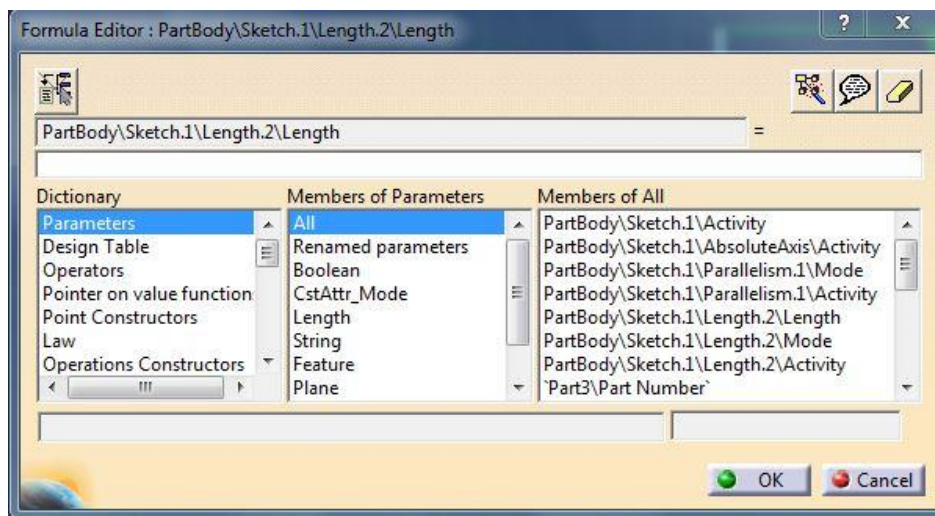


Figura 10. Cuadro de edición *Formula Editor*

Una vez creada la fórmula, ya no se puede escribir ningún valor en el cuadro de edición. Éste queda bloqueado en función de la relación impuesta. Además, en el área gráfica se indica junto al valor que éste lleva implícita una fórmula como se observa en la Figura 11.



Figura 11. Cuadro de edición *Constraint Definition* editado

Por último, mencionar que en el árbol de trabajo se puede ver la relación impuesta en la rama *Relations* tal y como se observa en la Figura 12. Referencia [6]



Figura 12. Rama *Relations*

1.5.4 Reglas

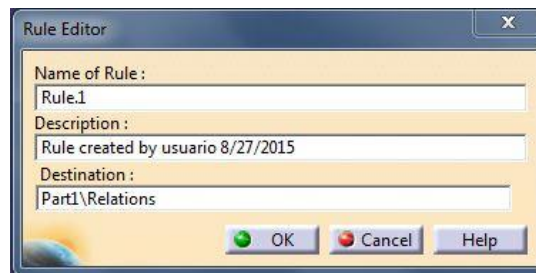
Las *Rule* dan la posibilidad de obligar al modelo que siga unas determinadas pautas. Éstas son fundamentalmente de dos tipos:

- Agrupaciones de fórmulas: consiguiendo tener un árbol más ordenado y con menos elementos.
- Reglas creadas por el propio usuario con un lenguaje intrínseco de CATIA y que sigue la estructura de un lenguaje sencillo de programación, VBA.

Normalmente es conveniente crear una Regla cuando tenemos un modelo que va a seguir unas pautas si cumple unas condiciones, y otras distintas si cumple otras condiciones. Para acciones de más relevancia habría que crear una Macro que es el objeto de este Trabajo Fin de Grado.

La Macro desarrollada en este trabajo hace uso de la herramienta *Rule* para incluir las fórmulas que dan lugar al perfil NACA de 4 dígitos.

Para crear una regla se debe pulsar sobre el icono de *Rule*, apareciendo de esta forma la ventana que aparece en la Figura 13.

Figura 13. Cuadro de diálogo *Rule Editor*

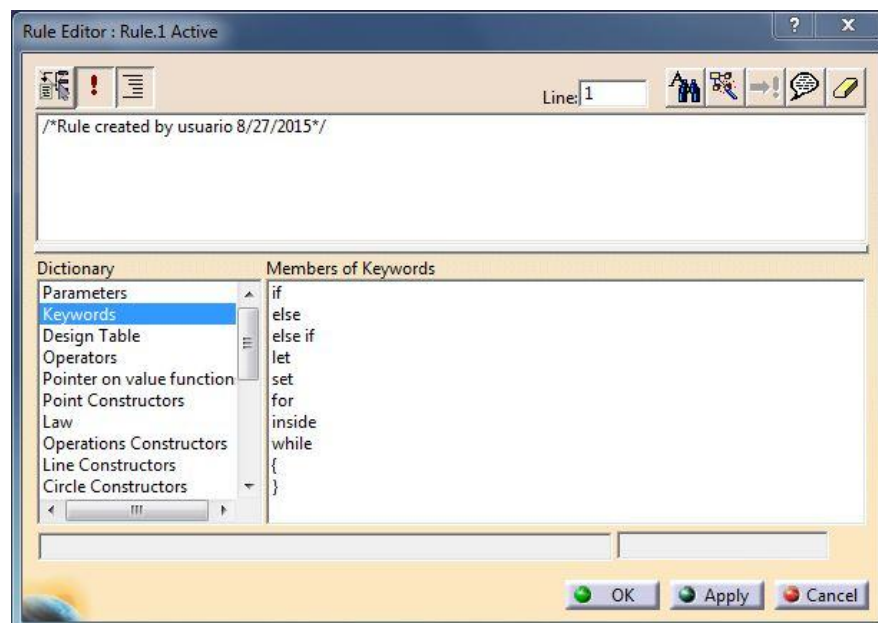
En la ventana *Rule Editor* podemos ver:

- El nombre de la regla: *Rule.1*. La primera regla creada en un documento es por defecto *Rule.1*, este nombre es el que aparecerá en el árbol a menos que se modifique.
- El usuario y la fecha de creación.
- El destino: una regla puede hacer referencias a especificaciones de fórmulas que lleve agregado el *Part* en su raíz o puede incluirse dentro de un nivel más bajo como un *Body* que esté insertado en el *Part* principal.

Si pinchamos en *OK* se desplegará el cuadro de edición *Rule Editor* el cual podemos observar en la Figura 14.

En este cuadro es donde se escribe la regla, pudiendo hacer uso de:

- Los *Parameters* que son los parámetros.
- Las *Keywords* que son las estructuras básicas de las que disponemos para construir la regla.
- Los *Operators* que son los operadores booleanos.

Figura 14. Cuadro de edición *Rule Editor*

Referencia [8]

1.6 Parametrización de un ala con perfil NACA de 4 dígitos en CATIA

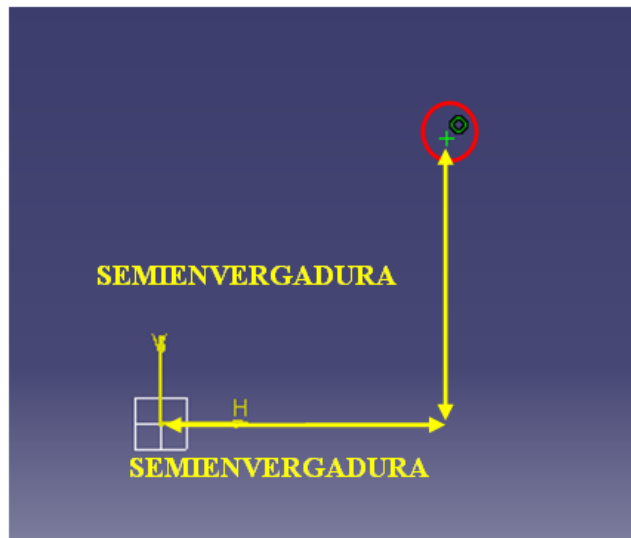
En este apartado se va a explicar el procedimiento que se sigue a la hora de parametrizar un perfil NACA de 4 dígitos y una ala que utiliza dichos perfiles. Con esto se pretende dos cosas, que el usuario final o el lector entienda como afecta cada parámetro sobre la geometría y justificar la decisión de programar dicho procedimiento.

Para empezar, lo que se debe hacer es crear todos los parámetros que se van a utilizar siguiendo el procedimiento explicado en el apartado 1.5.2. Estos son:

1. NACA1: primer dígito del perfil NACA situado en el encastre del ala.
2. NACA2: segundo dígito del perfil NACA situado en el encastre del ala.
3. NACA34: tercer y cuarto dígito del perfil NACA situado en el encastre del ala.
4. NACA1T: primer dígito del perfil NACA situado en la punta del ala.
5. NACA2T: segundo dígito del perfil NACA situado en la punta del ala.
6. NACA34T: tercer y cuarto dígito del perfil NACA situado en la punta del ala.
7. Cuerda: longitud (en milímetros) entre borde de ataque y borde de salida del perfil del encastre del ala.
8. Diedro: ángulo (en grados) que forma la línea de borde de ataque del ala respecto al suelo
9. Incidencia: ángulo (en grados) que forma el perfil del encastre del ala respecto al suelo.
10. Flecha: ángulo (en grados) que forma la línea de borde de ataque del ala respecto a un plano que es perpendicular al suelo y al plano que contiene la cuerda del perfil del encastre.
11. Semienvergadura: longitud (en milímetros) entre el perfil del encastre y el de la punta del ala.
12. Estrechamiento: tanto por ciento de estrechamiento de la cuerda del perfil de la punta del ala respecto a la cuerda del perfil del encastre del ala.
13. Torsión: ángulo (en grados) que forma el perfil de la punta del ala respecto al suelo.

Más adelante en este mismo apartado, se visualizará como afecta cada parámetro a la geometría del perfil y/o ala. Una vez creado todos los parámetros, el siguiente paso es dibujar en un *Sketch* el perfil del encastre. Para ello, dentro del *Sketch* se crearán dos líneas de puntos, que serán el extradós y el intradós del perfil las cuales constarán de 60 puntos en total (sin contar el punto del borde de ataque) para poder representar de manera fiel el contorno del perfil. Posteriormente, se creará un punto más, el del borde de ataque.

El siguiente paso es definir dos líneas de referencia para poder incluir el ángulo de incidencia del perfil. Para ello, en primer lugar, se debe acotar el punto creado para el borde de ataque respecto al sistema de referencia del *Sketch*. Una vez creada las *Constraints* se editará una fórmula en cada una de éstas de forma que dicho punto tenga una distancia en longitud y en altura igual a la semienvergadura tal y como se ve en la Figura 15.

Figura 15. *Constraint* punto borde de ataque perfil encastre

El motivo de situar el punto del borde de ataque a una longitud y altura igual a la semienvergadura es porque las cotas en CATIA no entienden de valores negativos. Algunas de las coordenadas de los puntos del perfil tienen valores negativos en su coordenada vertical y si además le introducimos un ángulo de incidencia al perfil, puede que también alcancen valores negativos en su coordenada horizontal. Por lo que a la hora de asignar los valores a las cotas de los puntos del perfil le sumaremos un *offset* a sus coordenadas para que siempre sean positivas.

El valor del *offset* debe ser además alto ya que los puntos de los *Sketch* siempre deben estar contenidos dentro del primer cuadrante del sistema de referencia global del *Sketch* y si no se encuentra lo suficientemente alejado del eje horizontal y vertical puede que al modificar algún parámetro se sobrepase dicho límite empezando a salir avisos de error en CATIA. Por este motivo se decide imponer que dicha distancia sea igual a Semienvergadura en el caso de que estemos diseñando alas e igual a Cuerda si se está diseñando perfiles ya que normalmente serán las distancias más grandes que se manejen en los respectivos casos.

Por último comentar que dicho punto se debe acotar sobre el sistema de referencia global del *Sketch* el cual permanece fijo ya que se requiere que el punto del borde de ataque permanezca “fijo” en el espacio ya que a través de él se define, además de las líneas de referencia a través de las cuales se acotarán el resto de puntos del perfil, la línea que unirá todos los puntos de los bordes de ataque de todos los perfiles la cual se utiliza como eje de giro para implementar la incidencia y/o torsión de los perfiles y como guía para la extrusión del ala.

Una vez tengamos el punto del borde de ataque acotado con sus respectivas fórmulas, se procede a crear dos líneas de construcción las cuales, tal y como se comentó anteriormente, ejercerán la función de sistema de referencia local para el resto de puntos que conforman el perfil. La primera de ellas tendrá un ángulo igual al ángulo de incidencia respecto al eje vertical y además, será coincidente con el punto del borde de ataque; y la segunda será siempre perpendicular a la primera. Al ser líneas de construcción nos da igual la longitud de las mismas. En la Figura 16 se puede ver un ejemplo del aspecto final que pueden tener dichas líneas ya que estas van cambiando al modificar los parámetros que definen el punto del borde de ataque o el ángulo que forma

con la vertical.

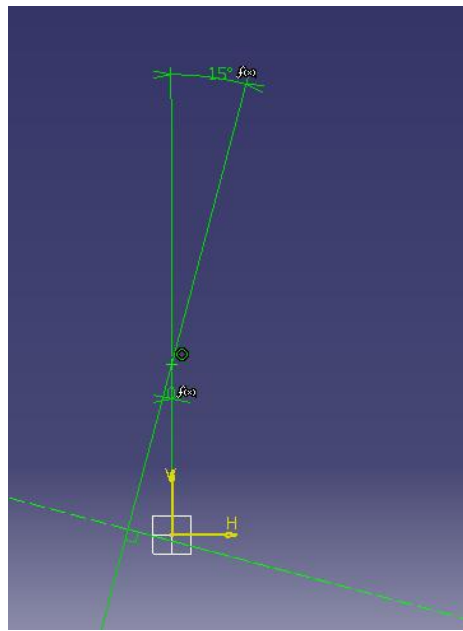


Figura 16. Líneas de referencia perfil encastre

A continuación se acotan las distancias de los 60 puntos creados al principio con respecto cada una de las líneas creadas anteriormente, la primera línea hace de eje vertical y la segunda de horizontal. Al acotar los puntos se debe seguir una lógica, ya que posteriormente se deben realizar operaciones sobre dichas cotas por lo que se debe saber el orden de acotación de los puntos en longitud y altura. El orden utilizado para el presente proyecto ha sido el siguiente: de la 6 a la 35 las cotas en longitud de los puntos del extradós, de la 36 a la 65 las cotas en altura de los puntos del extradós, de la 66 a la 95 las cotas en longitud de los puntos del intradós y de la 96 a la 125 las cotas en altura de los puntos del intradós. Las cotas no empiecan en el número uno debido a que previamente se han acotado el punto de control y las líneas de referencias tal y como se puede ver en la Figura 17.

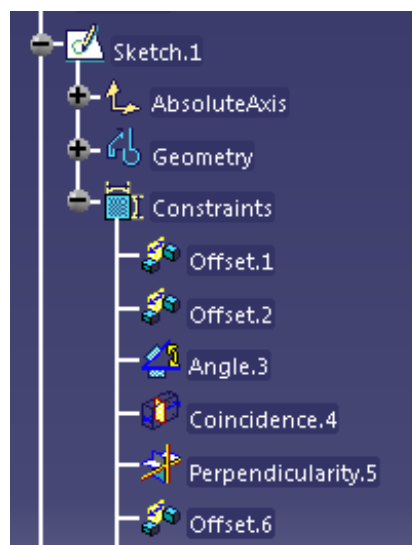


Figura 17. Constraints Sketch.1

Para finalizar el dibujo del perfil, se accede al módulo *Knowledge Advisor* y creamos una *Rule* donde se debe incluir las ecuaciones (6), (7), (8) y

(9) para sus correspondientes puntos, es decir, las ecuaciones (6) y (7) para cada punto del extradós y las ecuaciones (8) y

(9) para cada punto del intradós.

Antes de ponerse a escribir las ecuaciones en el *Rule Editor* hay que tener en cuenta que en CATIA, la sintaxis de la sentencia condicional es : If {condición} {instrucciones} else {instrucciones}, y que la potencia se escribe como **.

La línea de código que se debe implementar para las cotas de la 6 a la 35 es la del Código 1, para las cotas de la 36 a la 65 la del Código 2, para las cotas de la 66 a la 95 la del Código 3 y para las cotas de la 96 a la 125 la del Código 4. Como se ha mencionado antes, dichas líneas se deben repetir para todas las cotas por lo que se deberá cambiar el número de la cota, es decir, del *offset*. Además, la coordenada x debe ir de 20 a 1000 la cual esta adimensionalizada con la cuerda de forma que $x/1000 * \text{Cuerda}$ va de $20/1000 * \text{Cuerda}$ hasta $1000/1000 * \text{Cuerda}$. Se debe dividir los elementos por 1000 ya que las unidades de los parámetros tipo variable real están en el SI y las cotas en milímetros.

```

if (x/1000*Cuerda) <= (Cuerda*(NACA2/10)) PartBody\Sketch.1\Offset.6\Offset = (x/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(x/1000)-0.1260*(x/1000)-
0.3516*(x/1000)**2+0.2843*(x/1000)**3-
0.1015*(x/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(x/1000)))) else
PartBody\Sketch.1\Offset.6\Offset = (x/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(x/1000)-0.1260*(x/1000)-
0.3516*(x/1000)**2+0.2843*(x/1000)**3-0.1015*(x/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(x/1000))))
    
```

Código 1. Cotas en longitud de los puntos del extradós del perfil del encastre

```

if (20/1000*Cuerda) <= (Cuerda*(NACA2/10)) PartBody\Sketch.1\Offset.36\Offset =
(Semienvergadura+(NACA1/100)*(20/1000*Cuerda)*(2*(NACA2/10)-
20/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(20/1000)))) else
PartBody\Sketch.1\Offset.36\Offset = (Semienvergadura+(NACA1/100)*(Cuerda-20/1000*Cuerda)*(-
2*(NACA2/10)+20/1000+1)/(1-NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1/100)/((1-(NACA2/10)**2)*((NACA2/10)-(20/1000))))
    
```

Código 2. Cotas en altura de los puntos del extradós del perfil del encastre

```

if (20/1000*Cuerda) <= (Cuerda*(NACA2/10)) PartBody\Sketch.1\Offset.67\Offset = (20/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(20/1000)))) else
PartBody\Sketch.1\Offset.67\Offset = (20/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-0.1015*(20/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(20/1000))))

```

Código 3. Cotas en longitud de los puntos del intradós del perfil del encastre

```

if (20/1000*Cuerda) <= (Cuerda*(NACA2/10)) PartBody\Sketch.1\Offset.97\Offset =
(Semienvergadura+((NACA1/100)*(20/1000*Cuerda)*(2*(NACA2/10)-20/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(20/1000)))) else
PartBody\Sketch.1\Offset.97\Offset = (Semienvergadura +((NACA1/100)*(Cuerda-20/1000*Cuerda)*(-
2*(NACA2/10)+20/1000+1)/(1-NACA2/10)**2)-NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1/100)/((1-(NACA2/10))**2)*((NACA2/10)-(20/1000))))

```

Código 4. Cotas en altura de los puntos del intradós del perfil del encastre

Con todo esto ya se tienen las coordenadas x_U , y_U , x_L e y_L parametrizadas con el número del perfil NACA y con la cuerda. En las ecuaciones implementadas en los Código 2 y Código 4 se ha añadido la cantidad Semienvergadura (Cuerda si solo se dibuja el perfil y no el ala) para que todos los puntos estén elevados esa cantidad respecto al origen de coordenadas por los motivos comentados anteriormente.

Una vez realizado todo el proceso e introducidas todas las frases en el recuadro de programación de la *Rule* pulsamos *Apply* y *OK*. Posteriormente a esto, se dibuja un *Spline* sobre todos los puntos, obteniendo finalmente el perfil del encastre. En la Figura 18 y la Figura 19 se puede ver un ejemplo de perfil NACA, uno con incidencia nula y otro con un ángulo de incidencia no nulo.

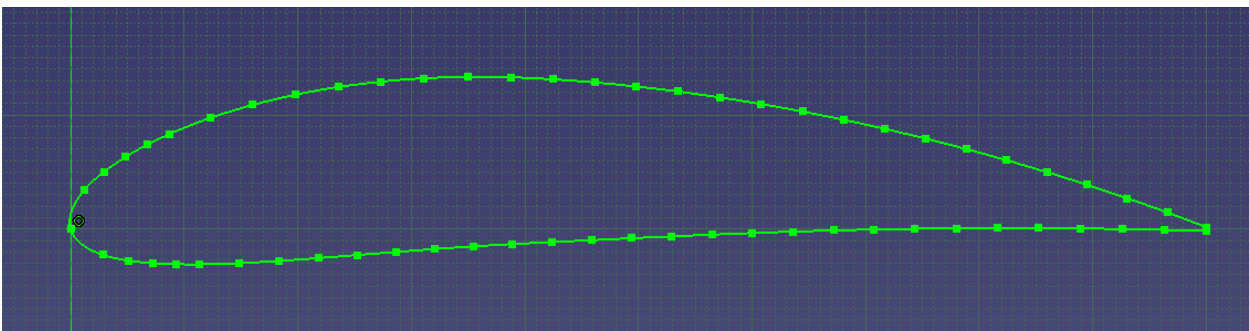


Figura 18. Perfil NACA con incidencia nula

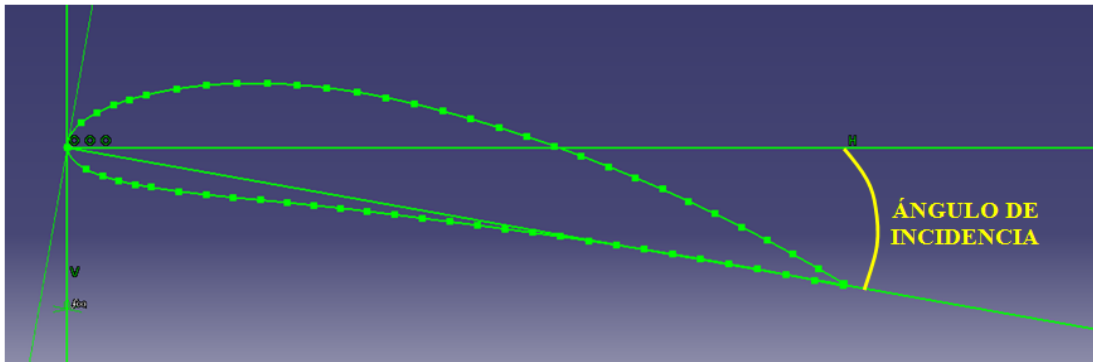


Figura 19. Perfil NACA con ángulo de incidencia

El siguiente paso después de dibujar el perfil del encastre es definir un plano paralelo al del *Sketch* donde se ha dibujado dicho perfil a una distancia igual a la semienvergadura. Para ello se debe editar una fórmula con ese parámetro para el *offset* del plano, de forma que la distancia entre planos se modifique automáticamente con tan solo cambiar el valor de Semienvergadura.

Con el plano ya creado se procede a crear un nuevo *Sketch* dentro de éste donde se dibujará el perfil de la punta del ala. Antes de comenzar a dibujar el perfil, se debe parametrizar algunas características muy importantes en las alas, que son: el diedro y la flecha. En la Figura 20 y la Figura 21 podemos visualizar que significa cada característica respectivamente.

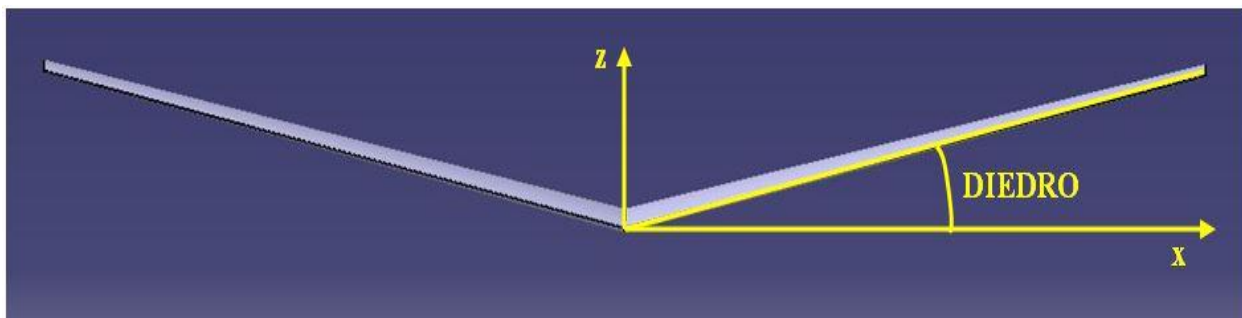


Figura 20. Vista frontal ala, ángulo de diedro

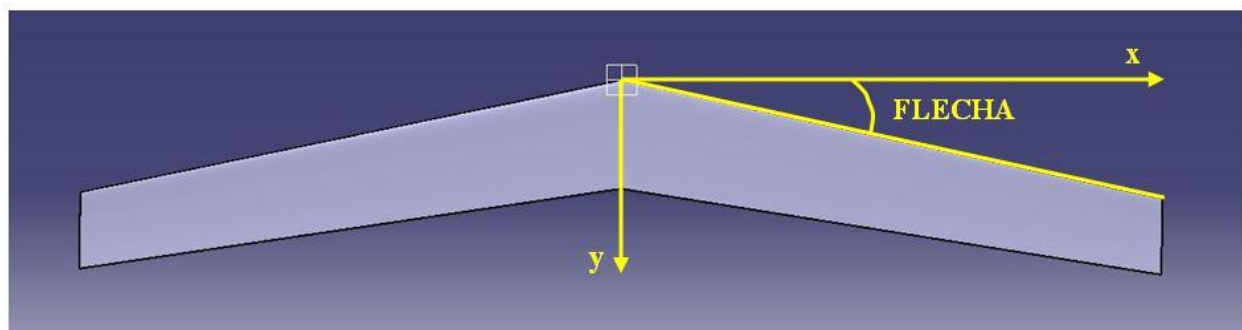


Figura 21. Vista en planta ala, ángulo de flecha

Para conseguir parametrizar el diedro y la flecha lo que se debe hacer es acotar el punto que define el borde de ataque del perfil de la punta del ala. Para ello, tal y como aparece la Figura 22, se le debe sumar a las distancias

Ahora que ya tenemos las líneas de referencia perfectamente acotadas, el siguiente paso es volver a crear 60 puntos por encima del origen de coordenadas y contenidas dentro de las líneas de referencia tal y como se hizo para el perfil del encastre del ala. Después de esto, se acotarán todos los puntos por su distancia en longitud y altura a las líneas de referencia siguiendo la misma lógica empleada en el perfil del encastre.

A continuación, se tiene que crear la *Rule* a través de la cual se conseguirá parametrizar todos los puntos del perfil. En este caso, a diferencia del otro perfil, se introducirá en la ecuación que define el perfil el parámetro estrechamiento a través del cual se consigue definir la cuerda del perfil de la punta de ala. Para ello se introducirá el trozo de línea $((100-\text{Estrechamiento})/100)*\text{Cuerda}$ dentro de la condición en el lugar donde ponía cuerda anteriormente.

En este caso, se introducirá las líneas de código del Código 5 para las cotas en longitud del extradós del perfil, el Código 6 para las cotas en altitud del extradós del perfil, el Código 7 para las cotas en longitud del intradós del perfil y el Código 8 para las cotas en altitud del intradós del perfil. Las cotas irán de la 134 a la 163, de la 164 a la 193, de la 194 a la 223 y de la 224 a la 253 respectivamente. Además, en dichos códigos, se puede observar que los parámetros que definen el número de perfil NACA utilizado, se llaman de distinta forma que los del encastre, pudiendo tener de esta forma perfiles diferentes en cada extremo del ala, permitiendo que la geometría del perfil vaya variando a lo largo de la envergadura si esto fuese requerido.

```
if (20/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.135\Offset = (20/1000*((100 - Estrechamiento)/100)*Cuerda -
NACA34T/(100*0.2)*((100 - Estrechamiento)/100)*Cuerda*(0.2969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA2T/10)**2)*((NACA2T/10)-(20/1000)))) else
PartBody\Sketch.2\Offset.135\Offset = (20/1000*((100-Estrechamiento)/100)*Cuerda -
NACA34T/(100*0.2)*((100-Estrechamiento)/100)*Cuerda*(0.2969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-0.1015*(20/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(20/1000))))))
```

Código 5. Cotas en altura de los puntos del extradós del perfil de la punta del ala

```
if (20/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.165\Offset = (Semienvergadura*tan(Diedro))+(Cuerda
+((NACA1T/100)*(20/1000*((100-Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
20/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA2T/10)**2)*((NACA2T/10)-(20/1000)))) else
PartBody\Sketch.2\Offset.165\Offset = (Semienvergadura*tan(Diedro))+(Cuerda
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-20/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+20/1000+1)/(1-NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-0.1015*(20/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(20/1000))))))
```

Código 6. Cotas en longitud de los puntos del extradós del perfil de la punta del ala

```

if (20/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.195\Offset = (20/1000*((100-Estrechamiento)/100)*Cuerda +
NACA34T/(100*0.2)*((100-Estrechamiento)/100)*Cuerda*(0.2969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA2T/10)**2)*((NACA2T/10)-(20/1000)))) else
PartBody\Sketch.2\Offset.195\Offset = (20/1000*((100-Estrechamiento)/100)*Cuerda +
NACA34T/(100*0.2)*((100-Estrechamiento)/100)*Cuerda*(0.2969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-0.1015*(20/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(20/1000))))))

```

Código 7. Cotas en longitud de los puntos del intradós del perfil de la punta del ala

```

if (20/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.225\Offset = (Semienvergadura*tan(Diedro))+
(Semienvergadura+((NACA1T/100)*(20/1000*((100-Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
20/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA2T/10)**2)*((NACA2T/10)-(20/1000)))) else
PartBody\Sketch.2\Offset.225\Offset = (Semienvergadura*tan(Diedro))+
(Semienvergadura+
((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-20/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+20/1000+1)/(1-NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-0.1015*(20/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(20/1000))))))

```

Código 8. Cotas en altura de los puntos del intradós del perfil de la punta del ala

Por último, para el perfil de la punta del ala, comentar que al igual que el perfil del encastre, la coordenada x debe ir de 20 a 1000 la cual esta adimensionalizada con la cuerda de forma que $x/1000 * \text{Cuerda}$ va de $20/1000 * \text{Cuerda}$ hasta $1000/1000 * \text{Cuerda}$. Se debe dividir los elementos por 1000 ya que las unidades de los parámetros tipo variable real están en el SI y las cotas en milímetros.

Una vez tengamos la *Rule* creada con todas las líneas de códigos para el perfil de la punta del ala, se dibujará un *Spline* con dichos puntos, obteniendo finalmente el dibujo del perfil.

A continuación se van a ilustrar una serie de figuras donde se representan las distintas características mencionadas anteriormente. En todas ellas se representará en blanco el perfil del encastre y en verde el de la punta del ala.

- En la Figura 24 se ha dibujado un perfil con un estrechamiento del 30% para el perfil de la punta del ala.

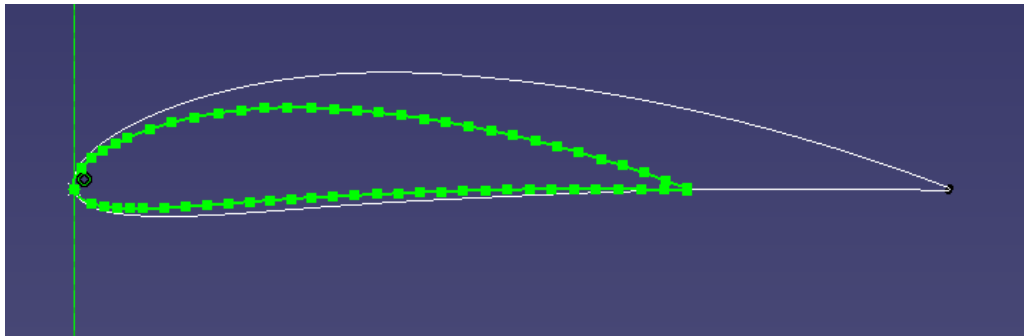


Figura 24. Estrechamiento

- En la Figura 25 se ha dibujado perfiles distintos para cada extremo del ala.

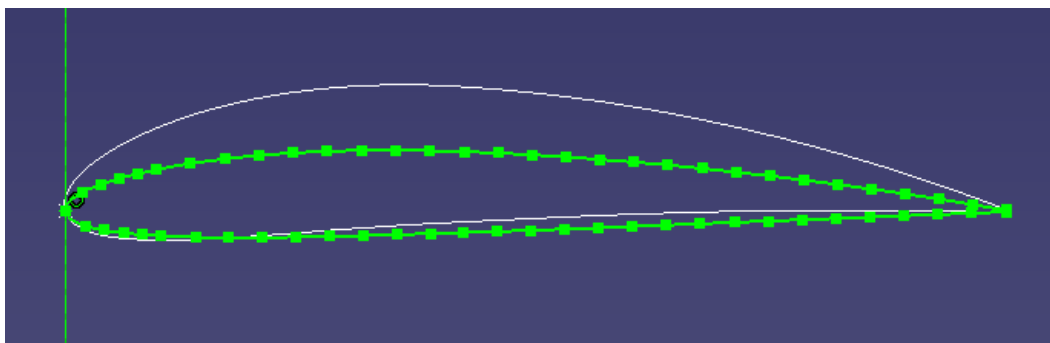


Figura 25. Perfiles distintos

- En la Figura 26 se ha dibujado el perfil del encastre con un ángulo de incidencia y el de la punta del ala con un ángulo de torsión. Aunque el ángulo de torsión en este ejemplo sea más grande que el de incidencia, este puede ser menor.

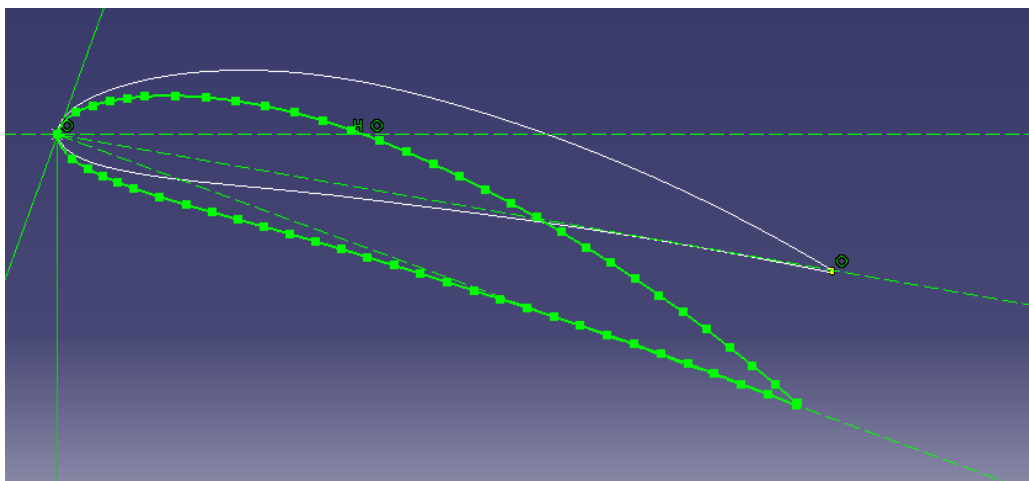


Figura 26. Incidencia y Torsión

- En la Figura 27 se puede visualizar como modifica la ubicación del perfil de la punta del ala respecto al del encastre al introducir flecha y diedro al ala que se esta diseñando.

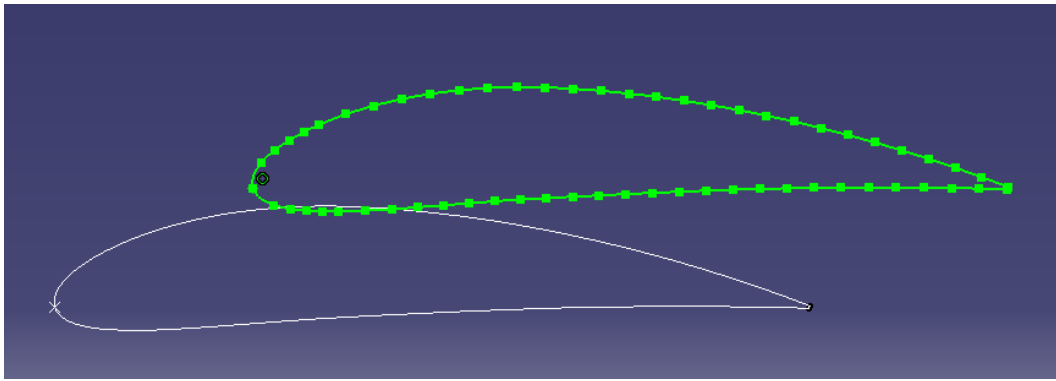


Figura 27. Diedro y Flecha

Llegados a este punto del diseño, ya se tiene ambos perfiles correctamente dibujados y parametrizados, por lo que ahora se debe dibujar la línea que unirá los puntos del borde de ataque de todos los perfiles a lo largo de la semienvergadura. Para ello, solo se tiene que crear una línea en el entorno 3D que una los puntos del borde de ataque del perfil del encastre y de la punta del ala, obtenido algo parecido a lo representado en la Figura 28.

Referencia [5]

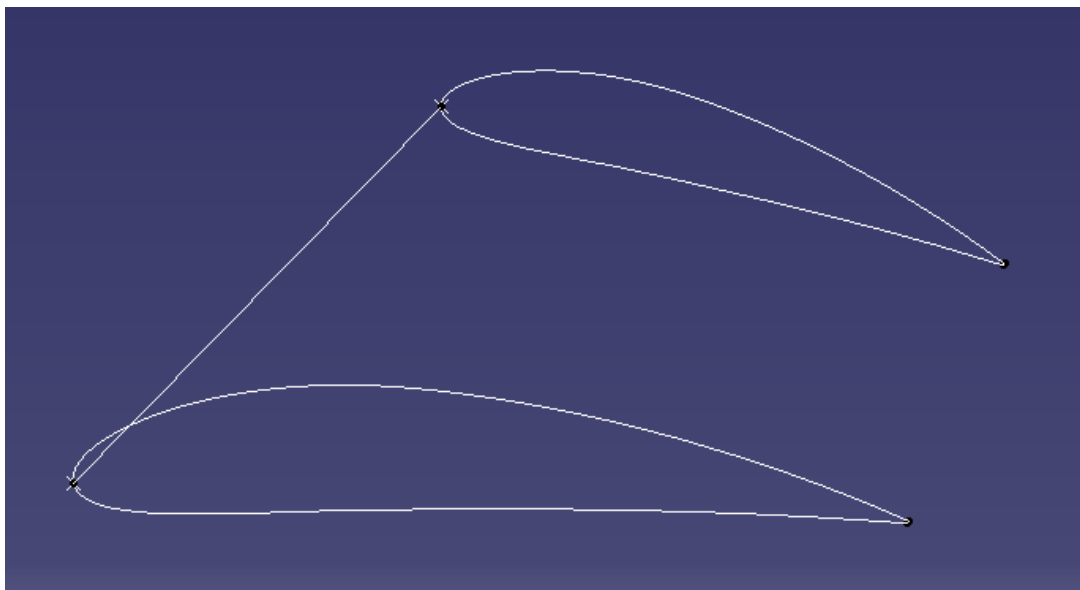


Figura 28. Perfiles y línea de borde de ataque

A partir de aquí, ya tan solo se tiene que extruir el ala, ya sea como superficie o como sólido, y aplicar las diferentes simetrías para llegar al resultado final que se requiera, el cual puede ser:

1. Semiala izquierda en superficie o sólida.
2. Semiala derecha en superficie o sólida.
3. Ala completa en superficie o sólida.

Se recomienda que en todos los casos, la extrusión del ala y demás operaciones se realicen en el módulo de superficies y que si al final se requiere un ala sólida, se utilice la herramienta *Close Surface* dentro del módulo

Part Design de CATIA para obtenerla. Por último, comentar que al trabajar con superficies se debe realizar cuando se acabe de trabajar un *Join* que contenga las diferentes operaciones realizadas dentro del módulo *Wireframe and Surface Design* para unir todos los elementos. En la Figura 29 se ilustra un ejemplo del resultado final de la realización del proceso de diseño paramétrico explicado en este apartado.

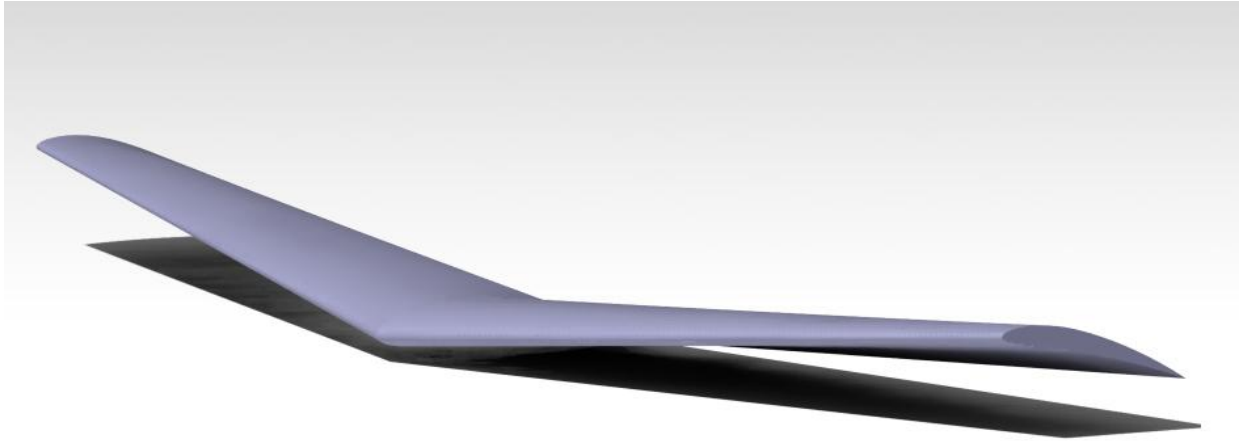


Figura 29. Ejemplo resultado final del diseño de un ala cualquiera

Como se puede ver después de leer este apartado, el proceso de diseño geométrico de un ala en CATIA no es algo trivial y menos aún si se realiza un diseño parametrizado. Debido a lo costoso y complicado que resulta llevar a cabo dicho proceso se decidió intentar crear una macro que realizase todo esto automáticamente.

2 EL ENTORNO DE PROGRAMACIÓN VBA EN CATIA V5

Puedes cerrar todas las bibliotecas si quieres, pero no hay barrera, cerradura, ni cerrojo que puedas imponer a la libertad de mi mente.

- Virginia Woolf -

En este capítulo se va a explicar a grandes rasgos el lenguaje de programación VBA (Visual Basic for Applications). Una vez introducido el entorno en el que se va a trabajar, se procederá a explicar el entorno general de trabajo VB6, así como nociones generales de programación de dicho lenguaje.

2.1 Visual Basic for Application

Como su nombre indica, el lenguaje de programación *Visual Basic* procede del denominado *BASIC* (*Beginner's All-purpose Symbolic Instruction Code*) que fue creado en 1964 en el *Dartmouth College*, como un medio para iniciarse en el mundo de la programación. Tras varias modificaciones, en 1978 se estableció el *BASIC standard*. Primero fue nombrado como *GW-BASIC*, luego quedó en *QuickBASIC* y actualmente se le conoce como *Visual Basic* tras adaptarse al entorno de ventanas “*Windows*” e incorporar herramientas de tipo visual como botones, listas de texto o cuadros de texto asociadas a eventos. La primera versión de *Visual Basic* fue presentada en 1991, siendo la última la versión 6, liberada en 1998.

Visual Basic for Applications (VBA) es el lenguaje de macros de *Visual Basic* v6. Éste fue incorporado en muchas aplicaciones de Microsoft y posteriormente en otras aplicaciones para ampliar la funcionalidad de las mismas. VBA incorpora las librerías y herramientas de *Visual Basic*, a las que añade librerías de objetos propias de cada software donde esta incluido. La debilidad de este lenguaje radica en que la compilación de la macro no puede realizarse si no se dispone del entorno en el que se ha desarrollado. Otra debilidad es su falta de versatilidad para trabajar en otros sistemas operativos.

CATIA en 1998 con la versión V5 incorporó VBA a su entorno, pudiendo realizar macros en VB y en lenguaje C++, siendo aún los lenguajes de macros que se han dispuesto para su versión V6. *Referencia [9]*

2.1.1 Entorno

En primer lugar se va a explicar que es una macro, ya que es importante saber que es lo que estamos realizando. Una macro consiste en una serie de funciones escritas en un lenguaje de programación que agrupa una serie de comandos, los cuales permiten realizar las operaciones requeridas automáticamente. Son usadas para ahorrar tiempo y reducir la posibilidad de errores humanos a la hora de realizar operaciones que se llevan a cabo de forma repetitiva.

El uso de macros para la automatización en el proceso de diseño es prácticamente ilimitado, siendo algunos ejemplos la importación de puntos desde Excel a un modelo CAD 3D, la generación de geometrías de manera automática, la creación de planos de modelos 3D, etc.

Para acceder al entorno de trabajo VBA no hace falta estar en ningún módulo de CATIA específico, se puede acceder a él desde cualquier módulo con tan solo pinchar dentro de la pestaña *Tools* en la opción *Macros*. Dentro de ésta se podrá o bien comenzar a grabar una macro, acceder a las macros ya realizadas y librerías o al editor de *Visual Basic*, ver Figura 30. *Referencia [10]*

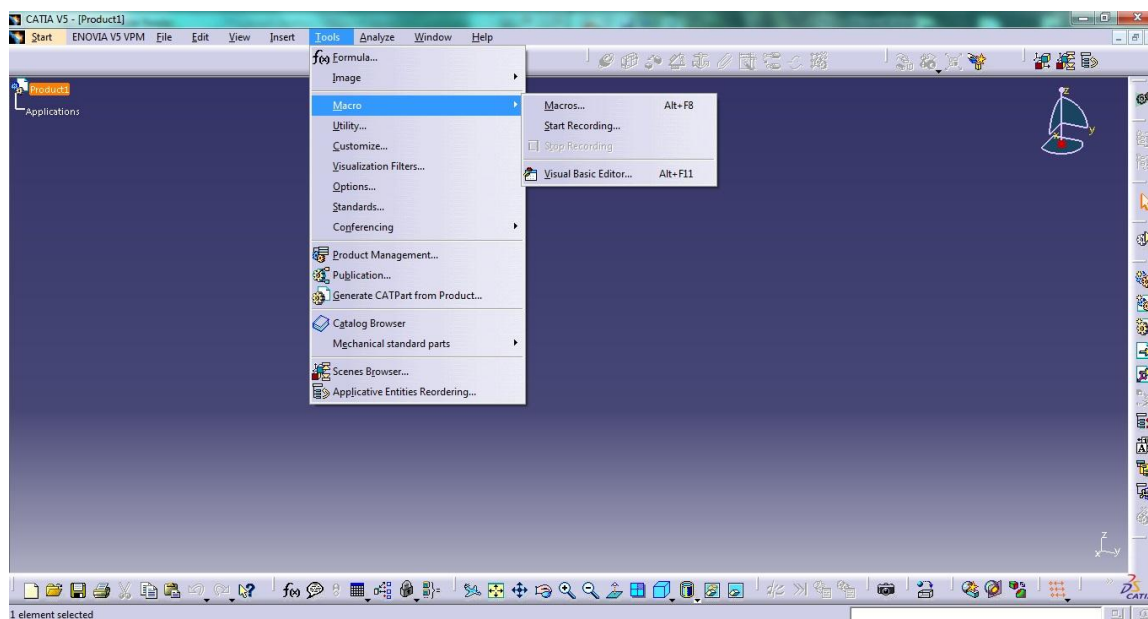


Figura 30. Acceso a macros

2.1.2 Librerías de las macros

Las macros de CATIA pueden ser almacenadas en las librerías de macros de tres formas posibles: *Folders* (vbscript y CATScript), *Project files* (catvba) o CATParts/CATProducts. Solo una de estas tres librerías de macros puede ser usada a la vez. Para crear una librería el procedimiento a usar es el siguiente.

1. Ir a *Tools* → *Macro* → *Macros*.
2. Abrir “*Macro libraries*”.
3. Asegurarse de que el tipo de librería esté cargada en “*Directories*” y luego hacer click en “*Add existing library*”.
4. Seleccionar la carpeta donde se van a guardar los CATScripts a lo largo del proyecto.
5. Cerrar la librería de macros. En dicha librería deberían aparecer posteriormente la lista de CATScripts que se realicen.

Referencia [10]

2.1.3 Macro Recording

Un método para crear macros es grabando las acciones que se realicen con el ratón. Para macros grabadas en un fichero o en un CATPart o CATProduct, los estamentos declarados se grabarán para CATScript pero no para MSVBScript. Para macros grabadas en una librería, “MS VBA” es la única opción. A la hora de grabar una macro mediante este procedimiento hay que tener en cuenta algunos aspectos:

- No seleccionar *Workbenches* (entornos de trabajo) mientras se está grabando una macro.
- No grabar más de lo que sea absolutamente necesario.
- No usar la opción “deshacer” mientras se está grabando.
- Ser consciente y darse cuenta de la configuración de CATIA cuando se está grabando.
- Salir de los *sketches* (dibujos) antes de parar de grabar.
- Verificar cada macro una vez se haya grabado.

Una vez se haya finalizado la grabación, se deshazá todo lo realizado y se reproducirá la macro, de manera que se podrá comprobar si es correcta la macro y si reproduce la operación que se quería realizar.

También hay que tener en cuenta que mediante dicho procedimiento aparecerán numerosas líneas de código que no son realmente necesarias, por lo que pueden eliminarse.

Por otro lado, tampoco aparecerán comentarios acerca de lo que se está haciendo o explicando los parámetros de entrada, por lo que se deberán añadir manualmente. *Referencia [10]*

Durante la realización de este proyecto se ha hecho uso de esta herramienta para aprender a usar operaciones de gran complejidad de las cuales no existía mucha información al respecto. Sin embargo, todos los códigos desarrollados en el presente proyecto se han realizado mediante la opción que se explica en el siguiente apartado.

2.2 Visual Basic Editor

El entorno de *Visual Basic Editor* es sobre el que se ha desarrollado este Trabajo Fin de Grado. Para acceder al mismo debemos pinchar en *Tools* → *Macro* → *Visual Basic Editor* apareciendo la ventana de la Figura 31. Todo lo que se programe y se realice dentro de esta ventana puede interactuar directamente con CATIA si empleamos los objetos del programa. Antes de comenzar conviene analizar detenidamente cada una de las partes o bloques que constituyen el entorno donde se realiza toda la programación.

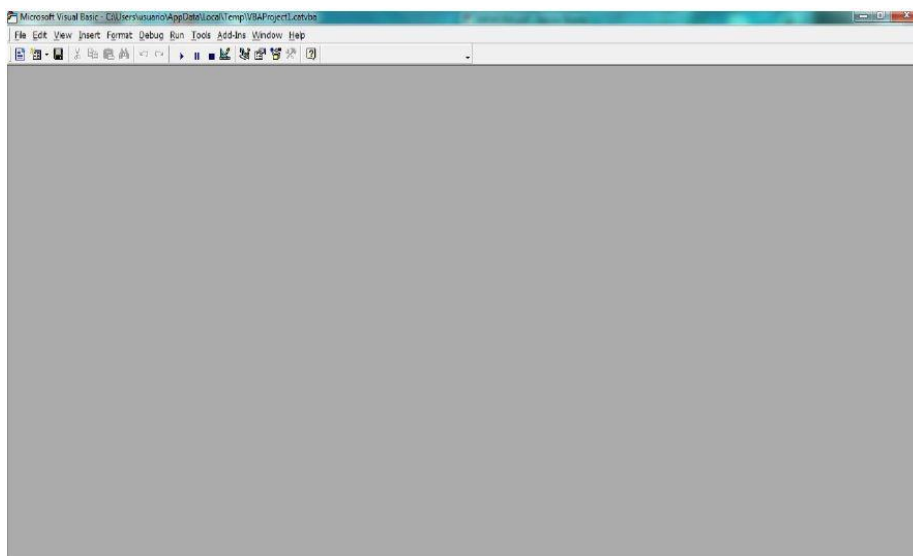


Figura 31. Entorno VBA

Antes de comenzar a programar conviene tener a disposición del programador tanto el menú denominado *Project Explorer* como el de *Properties Windows*, ya que son fundamentales y facilitan mucho el trabajo. Para acceder a estas ventanas es necesario ir a la pestaña *View* y adjuntar ambas aplicaciones a la pantalla principal, tal como se muestra en la Figura 32.

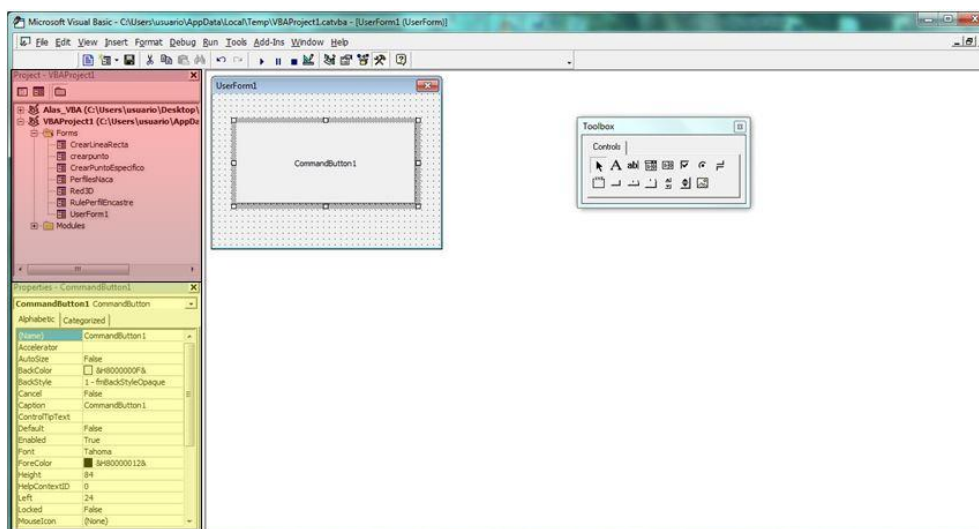


Figura 32. Ventana *Project Explorer* (Rojo) y *Properties Window* (Amarillo)

La ventana *Project Explorer* recoge todos los datos y archivos que conforman el *Project* o proyecto de programación sobre el que se está trabajando, suelen exportarse con extensión *.vbp, aunque CATIA lo almacena como *.catvba. Puede contener módulos (*Module*), formularios (*Userforms*) y clases (*Class module*).

- Módulos: son programas o rutinas independientes. Dado que el código contenido en un módulo estándar de *Visual Basic* es accesible desde distintos formularios del programa, será ventajoso colocar en este módulo todo lo que queramos disponer como “código compartido”, es decir, que puede ser utilizado en cualquier formulario del programa. Se guardan para su exportación en ficheros con extensión *.bas.
- Formularios: son rutinas asociadas a ventanas gráficas donde se incorporan objetos y eventos. Los formularios son el elemento básico que permite la interacción del programa con el usuario, demandando variables, opciones, etc. Se almacenan con extensión *.frm.
- Clases: son definiciones de nuevos objetos de tipo plantilla donde se definen las propiedades y eventos del mismo, son almacenados como *.cls.

En la ventana *Propiedades Windows* se refleja en una columna todas las propiedades de cada formulario o control que se tenga seleccionado en el momento en la ventana *Project Explorer*. Para acceder a dicha información solo tenemos que pinchar sobre el formulario o control deseado. Además, estas propiedades pueden ser cambiadas durante el diseño de dichos controles o formularios.

Por último, en este apartado se va a explicar otra ventana que ha sido de mucha utilidad en la ejecución de este Trabajo Fin de Grado, la ventana *Object Browser*, a la cual se accede haciendo click en el icono correspondiente señalado con un círculo rojo en la Figura 33, en dicha figura también podemos ver la apariencia de dicha ventana.

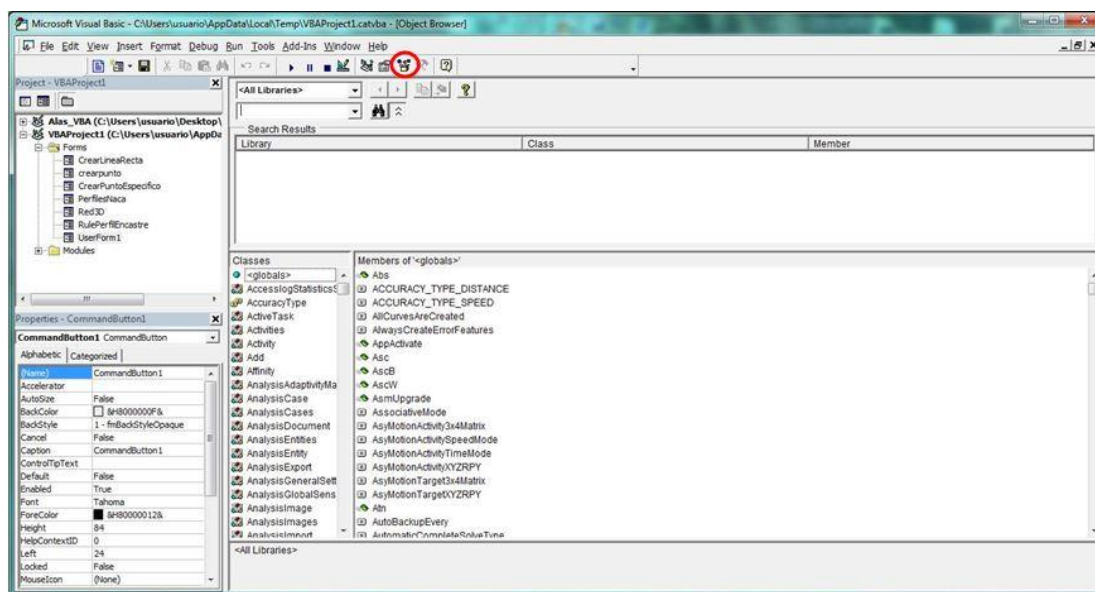


Figura 33. Ventana *Object Browser*

Dicha herramienta resulta de gran ayuda cuando se está en un punto de la programación en la que no se sabe bien qué hacer, cómo seguir o cómo funciona cualquier tipo de objeto. Si escribimos cualquier objeto en la barra de búsqueda, se muestran tres columnas que ayudan a controlar dicho objeto. La tercera de ellas, la columna *members*, muestra distintas formas para realizar la misma operación, eligiéndose la más conveniente. La segunda columna, muestra a qué tipo de objeto pertenece la operación que se quiere realizar, por lo que ya se sabrá que objeto hay que definir para poder acceder a la herramienta deseada. *Referencia [11]*

2.3 Iniciación a la programación

Visual Basic es un lenguaje de programación por eventos, es decir, la ejecución del programa se produce en diferentes secciones debido a respuestas a eventos. Estos se producen por la interacción con el usuario mediante una interfaz gráfica que solicita al usuario los parámetros necesarios para la ejecución de cualquier acción o también debido a la existencia de otras aplicaciones que desencadenan dichos eventos. Los eventos, ejecutados en forma de secuencia, determinan como se lee el código implementado.

Por esta razón, a la hora de programar, se tiene que prestar especial atención a cómo se está escribiendo el código para que responda de forma adecuada a los eventos para los que se diseñe la aplicación.

En la Tabla 2 se definen una serie de conceptos básicos totalmente necesarios de conocer ya que sobre ellos se basa la programación controlada por eventos.

Concepto	Definición
Tiempo de diseño	Instante en que se crea la aplicación
Tiempo de ejecución	Instante en el cual se ejecuta e si interactúa con la aplicación.
Formulario	Ventana sobre la que es posible personalizar la interfaz de la aplicación o cuadro de diálogo para obtener información del usuario.
Objetos	Formularios y controles.
Controles	Representación gráfica de objetos, con los que el usuario interactúa e aporta la información que se le pide.
Propiedades	Los valores de un objeto (<i>Properties Window</i> antes mencionada). Son características de un objeto, y definen el estado del mismo en un momento específico.
Métodos	Las acciones que un objeto puede realizar sobre sí mismo. Se suelen usar verbos para dar nombre a los métodos.

Eventos	Acciones que son reconocidas por un formulario o control. Los eventos ocurren a medida que el usuario interactúa con los objetos de la aplicación.
Colección	Grupo o lista de objetos similares que se ponen juntos por una razón específica. Las colecciones son objetos que agregan un conjunto de otros objetos.
Clases	Definen un tipo de objeto. Se suele usar la herencia para crear jerarquía entre clases y subclases.
Herencia	Todas aquellas clases que sean herencia de la misma clase tienen todas las propiedades y métodos en común de la herencia de la que provienen, pero también tienen sus propios métodos y propiedades que las diferencian entre ellas.

Tabla 2. Conceptos básicos de la programación

Referencia [9] y [10]

A continuación vamos a explicar otros conceptos que conviene resaltar antes de comenzar a programar.

2.3.1 Declaración de estamentos

En *Visual Basic* antes de poder trabajar con una determinada variable, constante o herramienta, es necesario nombrarla, obteniendo de esta forma una primera información acerca del estamento que posteriormente estableceremos.

Si no definimos el tipo de variable, VBA declarará la variable de tipo *Variant*, la cual puede aceptar cualquier tipo de variable. Cabe destacar que en muy raras ocasiones se tiene una buena razón para usar una variable tipo *Variant*. En la mayoría de ocasiones tendremos que usar un determinado tipo de variable, haciendo de esta manera que el código se ejecute más rápido y reduciendo errores, lo cual se debe a que:

1. CATIA ejecutará el tipo de variable que se especifique.
2. A la hora de revisar el código, siempre se sabrá de qué tipo es cada variable y la intención por la cual se creó.

El comando que se usa para definir el tipo de variable con el que se trabaja es *Dim*. En el Código 9 se muestra un ejemplo.

```
Dim documents1 As documents
```

Código 9. Declaración de estamento

[Referencia 10]

2.3.2 Estamentos

Son una instrucción completa que puede contener *Keywords* (*and, if, for, while, sub, function ...*), operadores (+, -, *, / ...), variables, constantes y expresiones. Un ejemplo de estamento sería el del Código 10.

```
Set partDocument1 = CATIA.ActiveDocument
```

Código 10. Estamento

Donde el comando *Set* se usa para asociar la variable definida con un objeto.

Referencia [9]

2.3.3 Estamentos ejecutables

Se trata de acciones iniciales como el del Código 11.

```
Select1.Search("name='Optimization.MinimumValue',all")
```

Código 11. Estamento ejecutable

Referencia [9]

2.3.4 Funciones y subfunciones

Son una secuencia de estados que conforman la operación deseada. Esta operación viene especificada en una función.

```
Sub mySubwithParameter (myParameter)
    MsgBox myParameter
End Sub
```

Código 12. Subfunción

La diferencia entre *function* y *sub* es que mientras la primera te devuelve un valor, la segunda no.

Referencia [9]

2.3.5 Estructuras condicionales e iterativas

2.3.5.1 Condicionales

Durante la programación, muchas veces se llega a situaciones en que es conveniente utilizar condicionales para comprobar, por ejemplo, que un valor se encuentra en un intervalo aceptable, es conveniente hacer uso de

la ejecución: *if... then...*

La estructura es la del Código 13.

```
If [condición] Then
    [Estamento]
Elseif [condición] Then
    [Estamento de Elseif]
Else
    [Estamento de Else]
End If
```

Código 13. Estructura condicional

Referencia [9]

2.3.5.2 Iterativas

Otra de las estructuras usadas con mucha frecuencia en la programación son las iterativas. Éstas son aquellas en las que se realiza una misma operación varias veces consecutivas con el fin de encontrar un valor justo o que haga funcionar un programa hasta que cierta condición se cumpla. Para hacer posible esto, existen varias configuraciones, las cuales se comentan a continuación.

1. *For ... Next*: se utiliza cuando queremos iterar un número de cosas dado.

```
Contador
For [contador] = [inicio] To [end] {paso a paso}
    [Estamentos]
Next
```

Código 14. Estructura *For ... Next*

2. *While ... Wend*: se utiliza cuando iteramos hasta que el contador cumple cierta condición.

```
Contador
While [{Contador} Condición]
    [Estamentos]
Wend
```

Código 15. Estructura *While ... Wend*

3. *Do ... Loop*: se utiliza cuando iteramos siempre que se cumpla cierta condición.

```
Do [{While/Until} condition]
```

```

[Estamentos]
[Exit Do]
[Estamentos]
Loop
Do [Estamentos]
[Exit Do]
[Estamentos]
Loop [{While|Until} condition]

```

Código 16. Estructura *Do ... Loop*

4. *For each ... Next*: se utiliza cuando se quiere iterar sobre los objetos en una determinada colección.

Referencia [9]

2.3.6 Objetos orientados a la programación

Los objetos son una parte de memoria en la que está contenida cierta información y metodología que permite operar con dicha información almacenada. Requiere de un cierto código especial para hacer que funcione, ya que dispone de comportamientos definidos por métodos específicos tales como:

1. Pasar información a través de parámetros.
2. Operaciones de cálculo que pueden:
 - a. Cambiar cierta parte de los datos iniciales.
 - b. Diseñar ciertas operaciones que no están por defecto.
 - c. Devolver valores de los datos iniciales.
 - d. Devolver resultados de los cálculos usando tanto la información introducida externamente como de la contenida en el objeto

Referencia [9]

2.3.7 Cómo definir un objeto

Para cada objeto en particular se define una clase (*classe*) que sirve como plantilla en la que se recoge cómo operan los objetos y los datos que contienen. Es conveniente resaltar que una clase puede ser utilizada para hacer funcionar uno o más objetos.

Clase	Objeto
Describe la estructura del objeto	Es el resultado de la clase
Es una plantilla	Tiene una copia única de toda variable no-

Especifica la representación de la información, el comportamiento, la interrelación (via variables, métodos y <i>parents</i> -estructura lógica)	estática pero no de las variables tipo clase (<i>static</i>).
--	---

Tabla 3. Diferencias entre clase y objeto

Referencia [9]

2.4 Interfaz con el usuario

En este apartado se explica cómo podemos crear nuestra propia interfaz para interactuar con el programa y así llegar a la programación por eventos de la que se habló anteriormente. Para ello se tiene que crear un formulario sobre el que se diseña la interfaz añadiendo los distintos controles necesarios para que el usuario interactúe con la aplicación.

Para insertar un formulario abrimos la pestaña *Insert* → *Userform*. El formulario tiene una forma como la que se puede apreciar en la Figura 34.

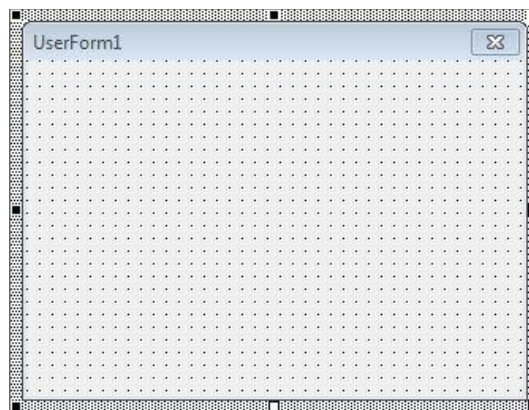


Figura 34. Formulario de trabajo

Ahora sólo falta añadir los elementos visuales através de los cuales se permite al usuario controlar el programa. Para ello es necesario activar la ventana de herramientas *Toolbox*, lo cual haremos desde la pestaña *view* → *Toolbox* Dicha ventana se muestre en la Figura 35.

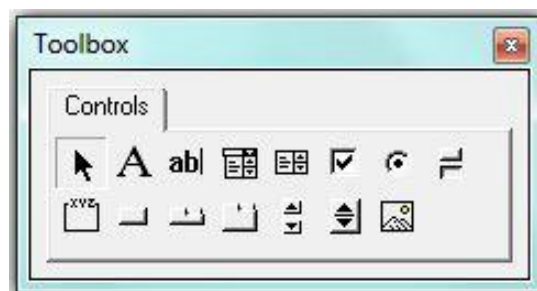


Figura 35. Herramientas de diseño

A continuación se en la Tabla 4 se analizan a grandes rasgos cada una de las herramientas existentes en este bloque de trabajo.

Símbolo	Etiqueta	Descripción
	<i>Label</i>	Permite escribir títulos o comentarios.
	<i>TextBox</i>	Permite al usuario introducir texto.
	<i>ListBox</i>	Control en el que se muestran varios registros, pudiendo seleccionar uno o más de uno.
	<i>ComboBox</i>	Control parecido al ListBox con una propiedad llamada <i>Style</i> , que permite 3 formas distintas de presentar una lista.
	<i>CheckBox</i>	Permite seleccionar una acción al usuario.
	<i>OptionButton</i>	Permite seleccionar una opción al usuario.
	<i>ToggleButton</i>	Botón para selección de opciones.
	<i>Frame</i>	Agrupar diferentes objetos referidos a un mismo tema.
	<i>CommandButton</i>	Permite ejecutar un evento.
	<i>TabStrip</i>	Separadores o etiquetas.
	<i>MultiPage</i>	Contenedor para una colección de objetos.
	<i>ScrollBar</i>	Permite tener una barra para desplazamientos.
	<i>SpinButton</i>	Permite aumentar o disminuir la cifra conforme se presionan las flechas del control.
	<i>Image</i>	Insertar una imagen en el formulario.

Tabla 4. Opciones de la barra de herramientas

La interfaz por si sola no realiza ninguna operación por lo que hay que introducir un código y programar cada uno de ellos. Para añadirlos sólo hay que hacer doble click sobre el mismo formulario y se abrirá una ventana como la de la Figura 36, la cual se corresponde a la de un *CommandButton*.

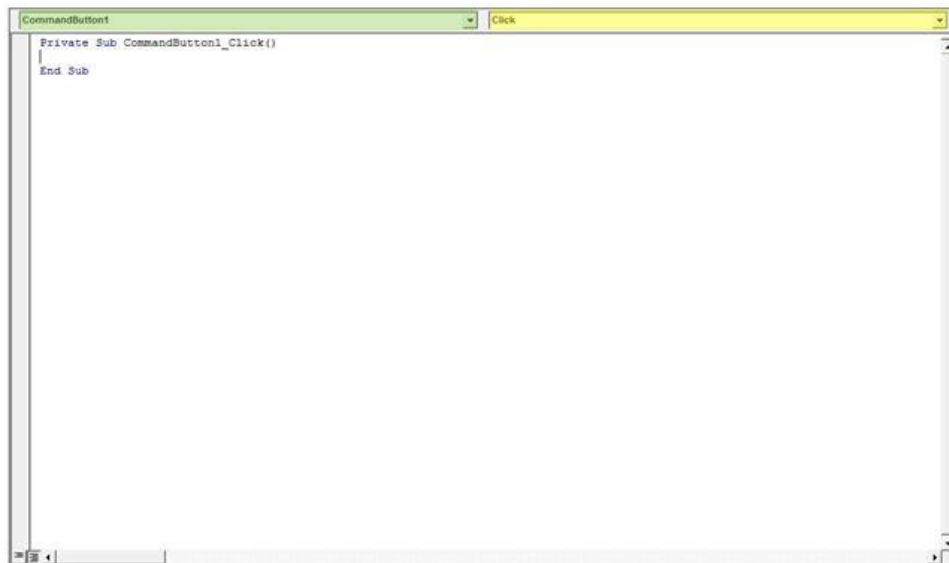


Figura 36. Editor

A continuación resulta conveniente describir la ventana de diálogo de la Figura 36. En ella se distinguen dos pestañas, una verde donde se recoge la lista de objetos disponibles para programar y una amarilla que indica para el objeto seleccionado, los eventos disponibles. Por último, en la esquina inferior izquierda, se tienen los botones para ver el procedimiento y el módulo.

En los siguientes apartados se explicará cómo funcionan algunos de los controles que se han mostrado en esta sección.

Finalmente, cuando tengamos la rutina diseñada, el siguiente paso será probarla. Para ello se dispone de una serie de opciones correspondientes con el menú *Run*, Figura 37, disponibles en el menú horizontal.



Figura 37. Comandos para hacer funcionar la aplicación

Cuando seleccionamos el botón *Play*, el programa se traslada al entorno de CATIA y aparece en pantalla la aplicación creada. *Referencia [9]*

3 HERRAMIENTAS DE PROGRAMACIÓN DE PERFILES Y ALAS

La vida es como montar en bicicleta. Si quieres mantener el equilibrio no puedes parar.

- Albert Einstein -

En esta sección se van a presentar todos los procedimientos llevados a cabo para realizar los perfiles y las alas que se describieron en el apartado 1.6.

Para la ejecución de la aplicación desarrollada, se ha trabajado con cuatro módulos:

1. El módulo *Sketcher* para realizar los diferentes *sketches* necesarios.
2. El módulo *Part Design* para crear sólidos en 3D a través de *Sketches*.
3. El módulo *Wireframe and Surface Design* para crear superficies en 3D a través de *Sketches*.
4. El módulo *Knowledgeware* para la creación de parámetros, fórmulas y reglas.

3.1 Arranque

El primer paso es definir y establecer el entorno de trabajo, sin necesidad de tener abierto el *Part*, lo cual se hará activando algunos parámetros que permitan al programador responder a las acciones que se le indican. De esta forma, cada vez que ejecutemos la aplicación se creará un nuevo *Part*. Para ello, se debe implementar el código escrito en Código 17.

```
Dim documents1 As Documents
Dim partDocument1 As PartDocument
Dim part1 As Part
Set documents1 = CATIA.Documents
```

```
Set partDocument1 = documents1.Add("Part")
Set part1 = partDocument1.Part
```

Código 17. Arranque de documentos

El *Documents* crea un nuevo *Part*, el *PartDocument* activa el módulo *Part Design*, que es donde se definen el resto de elementos que componen la aplicación desarrollada. Además, se define el archivo *Part* donde se creará la pieza, el cual se establece en el *PartDocument* definido anteriormente.

Una vez tengamos el *Part* de trabajo definido es preciso crear el árbol de trabajo donde se reflejarán todas las operaciones que se vayan realizando durante el proyecto, esto se consigue estableciendo el *body* de trabajo. Los *bodies* son los hijos del *Part*, por lo que en primer lugar habrá que definir el *bodies1* de trabajo y dentro de estos uno específico *body1*. Para ello se debe escribir un código como el del Código 18.

```
Dim bodies1 As Bodies
Set bodies1 = part1.Bodies
Dim bodies1 As Body
Set body1 = bodies1.Item("PartBody")
```

Código 18. Árbol de trabajo

Cuando estamos diseñando cualquier pieza, lo primero que se realiza es un *Sketch* donde poder dibujar. Dicho *Sketch* se genera dentro del *body* definido anteriormente. Por lo que el siguiente paso es establecer el sistema de referencia 3D con el que se va a trabajar y posteriormente definir en qué plano de este sistema se va a realizar el *Sketch* ya que es un dibujo en 2D. Finalmente, lo colocamos donde se va a realizar el dibujo dentro del plano de referencia escogido. En el Código 19 se puede ver un ejemplo en el cual se ha escogido el plano YZ al azar para realizar el *Sketch*, éste se guarda dentro de la referencia [*reference1*].

```
Dim sketches1 As Sketches
Set sketches1 = body1.Sketches
Dim originElements1 As OriginElements
Set originElements1 = part1.OriginElements
Dim reference1 As Reference
Set reference1 = originElements1.PlaneYZ
Dim sketch1 As Sketch
Set sketch1 = sketches1.Add(reference1)
```

Código 19. *Sketches*

Hasta este punto se puede observar una jerarquización de la estructura donde cada archivo está activado dentro de otro de manera que trabajan en un conjunto y en caso de faltar uno de ellos, el programa no encontrará donde realizar las acciones que se le piden originando el fallo global del mismo.

Con el *Sketch* y el plano de trabajo creados, lo siguiente es establecer una matriz que define las direcciones dentro de cada plano, de forma que el programa tenga claro en qué plano de trabajo se encuentra trabajando.

Esto funciona de la siguiente forma, CATIA coge las proyecciones de los vectores que definimos en la matriz de forma que estos sean perpendiculares entre sí de esta forma si el plano de trabajo no es ninguno de los planos principales de referencia resulta mas cómodo la elección de los vectores que conforman dicha matriz.

Para el plano YZ que es el que escogimos para el ejemplo anterior habría que escribir el Código 20.

```
Dim arrayOfVariantOfDouble1(8)
arrayOfVariantOfDouble1(0) = 0# 'Vector unidad x en el plano X
arrayOfVariantOfDouble1(1) = 0# 'Vector unidad y en el plano X
arrayOfVariantOfDouble1(2) = 0# 'Vector unidad z en el plano X
arrayOfVariantOfDouble1(3) = 0# 'Vector unidad x en el plano Y
arrayOfVariantOfDouble1(4) = 1# 'Vector unidad y en el plano Y
arrayOfVariantOfDouble1(5) = 0# 'Vector unidad z en el plano Y
arrayOfVariantOfDouble1(6) = 0# 'Vector unidad x en el plano Z
arrayOfVariantOfDouble1(7) = 0# 'Vector unidad y en el plano Z
arrayOfVariantOfDouble1(8) = 1# 'Vector unidad z en el plano Z
Set sketch1Variant = sketch1

sketch1Variant.SetAbsoluteAxisData arrayOfVariantOfDouble1
'Establece el sistema de ejes absoluto del sketch en 3D
```

Código 20. Vector de coordenadas

Finalmente se escribe la línea de código del Código 21 para informar que el *Sketch1* es el lugar de trabajo a partir de ese punto y todas las operaciones que se realicen serán sobre el mismo.

```
part1.InWorkObject = sketch1
```

Código 21. Establecimiento lugar de trabajo

Después de todo esto ya estamos dentro del *Sketch* y el siguiente paso es definir y establecer el conjunto de herramientas para realizar el dibujo 2D y todos los elementos geométricos necesarios.

```
' Establecimiento del conjunto de herramientas 2D y asignación al sketch de trabajo

Dim factory2D1 As Factory2D
Set factory2D1 = sketch1.OpenEdition()

' Se establecen los elementos geométricos

Dim geometricElements1 As GeometricElements
Set geometricElements1 = sketch1.GeometricElements

' Se define el sistema de ejes dentro del sketch

Dim axis2D1 As Axis2D
Set axis2D1 = geometricElements1.Item("AbsoluteAxis")

' Establecimiento de las direcciones horizontal y vertical

Dim line2D1 As Line2D
Set line2D1 = axis2D1.GetItem("HDirection")
```

```

line2D1.ReportName = 1
Dim line2D2 As Line2D
Set line2D2 = axis2D1.GetItem("VDirection")
line2D2.ReportName = 2

'Se cierra el sketch y se establece como objeto de trabajo

sketch1.CloseEdition
part1.InWorkObject = sketch1
part1.Update

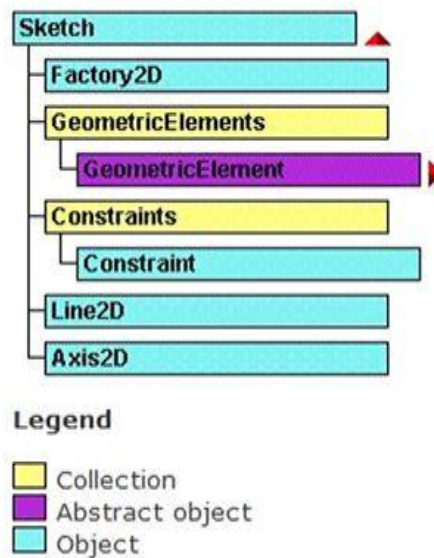
```

Código 22. Objeto *Factory 2D*

Las últimas líneas del Código 22, cierran el *Sketch* en cuestión, lo establece como objeto de trabajo, es decir, sobre el que se realizan a partir de ese punto las operaciones para el modelado de un sólido, y lo carga en el *Part*. Referencia [9]

3.2 SKETCHER

Dentro de este módulo existen una serie de objetos y colecciones como *Factory2D*, *Line2D* o *Axis2D*, tal como se puede apreciar en la que resultarán de gran utilidad en este proyecto.

Figura 38. Estructura del módulo *Sketcher*

Además, dentro de los objetos del módulo *Sketch* se encuentra una serie de métodos que se resumen en la Tabla 5.

Objeto	Comentario	Contenido
<i>Factory2D</i>	Incluye todos los métodos necesarios para poder trabajar en el módulo <i>Sketcher</i>	<i>CreateCircle</i>
		<i>CreateClosedCircle</i>
		<i>CreateClosedEllipse</i>
		<i>CreateControlPoint</i>
		<i>CreateEllipse</i>
		<i>CreateHyperbola</i>
		<i>CreateIntersection</i>
		<i>CreateIntersections</i>
		<i>CreateLine</i>
		<i>CreateLineFromVector</i>
		<i>CreateParabola</i>
		<i>CreatePoint</i>
		<i>CreateProjection</i>
		<i>CreateProjections</i>
<i>CreateSpline</i>		
<i>Line2D</i>	Engloba tres métodos que dan el uni-vector de la dirección de la recta, un punto sobre la línea y un último que permite modificar las características de la línea infinita.	<i>GetDirection</i>
		<i>GetOrigine</i>
		<i>SetData</i>
<i>Axis2D</i>	Vienen detalladas las propiedades del sistema de coordenadas	<i>HorizontalReference</i>
		<i>Origin</i>
		<i>VerticalReference</i>

Tabla 5. Objetos del módulo *Sketcher*

Una vez vista la estructura y los diferentes métodos que incluye cada objeto dentro del módulo *Sketcher*, se va a describir cómo usar los principales objetos y colecciones de dicho módulo para diseñar la geometría de los diferentes perfiles en cuestión. *Referencia [9]*

3.2.1 Crear un punto

Como se observa en la Tabla 5, el objeto *Factory2D* del módulo *Sketcher* posee herramientas para crear una amplia gama de geometrías. Se va a describir a continuación cómo crear un punto. Para ello, se usa las líneas de código del Código 23.

```
Dim point2D1 As Point2D
Set point2D1 = factory2D1.CreatePoint(coord_x, coord_y)
point2D1.ReportName = 3
```

```
point2D1.Construction = False
```

Código 23. Creación de un punto

Como se puede apreciar, solo es necesario introducir las coordenadas del punto que se desea crear sobre el *sketch* activado en dicho momento. A este punto se le puede asignar un nombre para que sea más fácil de localizar en el *Sketch*, para ello solo tendríamos que cambiar *point2D1* en las líneas de código del Código 23 por el nombre que queramos ponerle. A lo largo de todo el trabajo se ha seguido la lógica de llamarlo de la misma forma que el objeto incluyendo una numeración para identificar rápidamente con qué estamos tratando.

Por otro lado, también se puede imponer que dicho punto no sea de construcción, ya que CATIA lo crea de construcción por defecto, por este motivo se añade la última línea del Código 23, si se quisiese que el punto fuese de construcción tan solo tendríamos que cambiar el *False* por *True* o simplemente omitir esa línea. Referencia [9]

3.2.2 Crear una recta

Otra herramienta que se encuentra también en el objeto *Factory2D* es la que permite crear una recta a partir de dos puntos, el de origen y el final por lo que se trata de definir dos puntos y unirlos mediante el comando específico para crear una recta. A continuación, en el Código 24, se expone el código necesario para crear una recta.

```
Dim line2D1 As Line2D  
Set line2D1 = factory2D1.CreateLine(x1, y1, x2, y2)  
line2D1.ReportName = 3  
line2D1.StartPoint = point2D1  
line2D1.EndPoint = point2D2
```

Código 24. Creación de una recta

La creación del punto de origen y final (*Point2D1* y *Point2D2*) ha sido omitida en el Código 24, estos se crean tal y como se ha mostrado en el apartado anterior.

Como es posible apreciar, es preciso hacer uso del comando *Line2D* para crear la línea y posteriormente recurrir a *Factory2D* donde se encuentran todas las herramientas de trabajo en 2D, que en este caso se trata de *CreateLine*. Además, hay que señalar tanto el punto de inicio de la recta como el punto final, de forma que el programa pueda identificar que se trata de hacer una sola recta y no tiene que continuar con otra después.

Por último mencionar que CATIA por defecto crea una línea estándar, es decir, no es de construcción, y a lo largo de la macro desarrollada se hace uso de líneas de construcción tal y como vimos en el apartado 1.6. Para hacerlo, tan solo debemos añadir la línea del Código 25 justo antes de definir el punto de inicio y el punto final. Referencia [9]


```
line2D3.Construction = True
```

Código 25. Línea de construcción

3.2.3 Crear un Spline

Como bien sabemos con CATIA es posible crear curvas a partir de puntos de la misma, es decir, a través de una aproximación polinómica a partir de los distintos puntos que las conforman. Dichas curvas reciben el nombre de *Spline*.

Para generar un *Spline* en primer lugar se deben crear los puntos a través de los que se dibujara la curva, estos puntos no son unos puntos normales sino puntos de control. Para crear dichos puntos de control se hace uso de la herramienta *ControlPoint2D* del objeto *Factori2D*.

Para originar los puntos de control se procede de forma similar a la creación de un punto pero ahora se usa el comando *CreateControlPoint* tal y como se observa en el Código 26 donde se crean varios puntos de control a través de los que se dibujará el *Spline*.

```
Dim controlPoint2D1 As ControlPoint2D
Set controlPoint2D1 = factory2D1.CreateControlPoint(x1, y1)

controlPoint2D1.ReportName=3

Dim controlPoint2D2 As ControlPoint2D
Set controlPoint2D2 = factory2D1.CreateControlPoint(x2, y2)

controlPoint2D2.ReportName=4

Dim controlPoint2D3 As ControlPoint2D
Set controlPoint2D3 = factory2D1.CreateControlPoint(x3, y3)

controlPoint2D3.ReportName=5

Dim controlPoint2D4 As ControlPoint2D
Set controlPoint2D4 = factory2D1.CreateControlPoint(x4, y4)

controlPoint2D4.ReportName=6
```

Código 26. Creación de un punto de control

Una vez creado los puntos de control, se debe crear un vector que incluya todos los *ControlPoint* que conformarán la curva que queremos desarrollar. A la hora de crear dicho vector se debe tener especial cuidado a la hora de ir rellenando el vector ya que la curva recorrerá los puntos de control en el mismo orden que los introducimos. Por último, se debe hacer uso del comando *CreateSpline* del objeto *Spline* para generar la curva. Todo esto se puede ver en el Código 27.

```
Dim arrayOfObject1(3)
Set arrayOfObject1(0) = controlPoint2D1
Set arrayOfObject1(1) = controlPoint2D2
Set arrayOfObject1(2) = controlPoint2D3
Set arrayOfObject1(3) = controlPoint2D4

Dim spline2D1 As Spline2D
Set factory2D1temp=factory2D1
Set spline2D1 = factory2D1temp.CreateSpline(arrayOfObject1)
spline2D1.ReportName = 7
```

Código 27. Creación de un spline

Como se puede apreciar, el único argumento de entrada que precisa para el correcto funcionamiento de dicha herramienta es el vector objetos que en este caso son los *ControlPoint*. Referencia [10]

3.3 Restricciones

Para que el *Sketch* quede bien definido y fijo, es decir, que sea inalterable por algún error cuando se esté creando o manipulando, es necesario establecer las restricciones o *Constraints*.

Las *Constraints* trabajan con referencias, es decir, hay que establecer la referencia de cada uno de los objetos del *Sketch*. Una vez estén definidas, es necesario una instrucción que permita relacionarlas entre sí, para lo cual se usa el Código 28.

```
constraintsX.AddBiEltCs(CatCsTypeDistance, reference1, reference2)
```

Código 28 Creación constraints de forma general

En el comando del Código 28 se observan dos propiedades de gran interés:

- La instrucción *AddBiEltCs* que nos refleja que la restricción va a usar dos referencias, es decir, se van a relacionar dos objetos. Esta propiedad cambien en función de cuantos objetos se quieran relacionar, pueden ser uno, dos o tres objetos. Para ello solo se ha de cambiar el prefijo *Bi-* por el correspondiente, es decir, una restricción *AddMonoEltCs* solo necesitará una referencia y una *AddBiEltCs* tres referencias. En este proyecto solo se trabajará con restricciones de dos referencias.
- La instrucción *CatCsTypeDistance* que refleja el tipo de restricción que se esta imponiendo. Esta cambia en función del tipo de restricción que se quiera usar. En la Tabla 6 se observa todos los tipos de restricciones que existen.

Número de referencias	Tipo
<i>BiEtlcs</i>	<i>CatCsTypeAngle</i>
	<i>CatCsTypeAnnulContact</i>
	<i>CatCsTypeParallelims</i>
	<i>CatCsTypePerpendicularity</i>
	<i>CatCsTypeChamfer</i>
	<i>CatCsTypeConcentry</i>
	<i>CatCsTypeDistance</i>
	<i>CatCsTypeHorizontaly</i>
	<i>CatCsTypeLength</i>
	<i>CatCsTypeLineContact</i>
	<i>CatCsTypeMajor/MinorRadius</i>
	<i>CatCsTypeMidpoint</i>
	<i>CatCsTypeOn</i>
	<i>CatCsTypePlanarangle</i>
<i>CatCsTypeDistance</i>	
<i>CatCsTypeTangency</i>	
<i>MonoEtlC</i>	<i>CatCsTypeRadius</i>
<i>TriEtlcs</i>	<i>CatCsTypeSimetry</i>

Tabla 6. Restricciones

Para este tipo de órdenes, el programa sabe perfectamente que las referencias en cuestión se encuentran relacionadas según el tipo de restricción utilizada. Sin embargo, no conoce la cantidad exacta del parámetro, el cual dependerá del tipo de restricción en cuestión, que las relaciona (distancia, ángulo...). Por este motivo, hay que señalar que *Constraint* tiene dos modos de trabajo, estos son:

1. Modo *Constraint*: el valor asignado restringe la geometría en dicha posición.

```
Constraints1.Mode = CatCstModeDrivingDimension
```

Código 29. Modo *Constraint*

2. Modo *Measurement*: el valor solo refleja aquello que puede ser observado desde dicha posición.

```
Constraints1.Mode = CatCstModeDrivenDimension
```

Código 30. Modo *Measurement*

A continuación se va a explicar las restricciones empleadas en el desarrollo de este Trabajo Fin de Grado. *Referencia [9]*

3.3.1 CatCstTypeDistance

Esta restricción se usa para fijar la posición respecto al origen de coordenadas o la distancia entre dos objetos. También es posible utilizarlo para determinar la longitud de algunos objetos.

Además de los comandos introducidos al inicio de la subsección, es necesario declarar el valor de una longitud que será definida como la dimensión de la restricción. El procedimiento a seguir se detalla en el Código 31.

```

'Establecimiento de las constraints

Dim constraints1 As Constraints
Set constraints1 = sketch1.Constraints

'Constraint de la distancia horizontal de un punto al origen de coordenadas.

Dim reference7 As Reference
Set reference7 = part1.CreateReferenceFromObject(point2D3)

Dim reference8 As Reference
Set reference8 = part1.CreateReferenceFromObject(line2D2)

Dim constraint4 As Constraint
Set constraint4 = constraints1.AddBiEltCst(catCstTypeDistance, reference7, reference8)

constraint4.Mode = catCstModeDrivingDimension

Dim length2 As Length
Set length2 = constraint4.Dimension

length2.Value = x1

```

Código 31. *CatCstTypeDistance*

3.3.2 CatCstTypeOn

La restricción *CatCstTypeOn* es una de las restricciones más interesantes que existen. Dicha restricción permite imponer la coincidencia de dos puntos o curvas cualesquiera, por lo que permite introducir imposiciones geométricas muy importantes. En el Código 32, se crea la coincidencia de una recta con un punto de esta forma la recta siempre va a contener el punto. Como se puede apreciar, primero se hacen referencia al punto y a la línea por separado y luego se realiza la restricción de coincidencia.

```

Dim reference2 As Reference
Set reference2 = part1.CreateReferenceFromObject(line 2D7)

Dim reference3 As Reference
Set reference3 = part1.CreateReferenceFromObject(point2D128)

```

```
Dim constraint2 As Constraint
Set constraint2 = constraints1.AddBiEltCst(catCstTypeOn, reference2, reference3)
constraint2.Mode = catCstModeDrivingDimension
```

Código 32. *CatCstTypeOn*

3.3.3 CatCstTypePerpendicularity

Para fijar que una recta sea perpendicular a otra, se definen las dos rectas como referencias y se aplica la restricción *CatCstTypePerpendicularity* a las dos referencias creadas.

```
Dim reference20 As Reference
Set reference20 = part1.CreateReferenceFromObject(line2D7)

Dim reference21 As Reference
Set reference21 = part1.CreateReferenceFromObject(line2D5)

Dim constraint11 As Constraint
Set constraint11 = constraints1.AddBiEltCst(catCstTypePerpendicularity, reference20, reference21)
constraint11.Mode = catCstModeDrivingDimension
```

Código 33. *CatCstTypePerpendicularity*

3.3.4 CatCstTypeAngle

Cuando queremos que una recta forme un ángulo con otra se utiliza la herramienta *CatCstTypeAngle*. Para ello habrá que crear un ángulo y darle el valor a la restricción.

```
Dim reference19 As Reference
Set reference19 = part1.CreateReferenceFromObject(line2D6)

Dim reference20 As Reference
Set reference20 = part1.CreateReferenceFromObject(line2D7)

Dim constraint10 As Constraint
Set constraint10 = constraints1.AddBiEltCst(catCstTypeAngle, reference19, reference20)

constraint10.Mode = catCstModeDrivingDimension

Dim angle10 As Angle
Set angle10 = constraint10.Dimension
angle10.Value = torsionvalor
```

Código 34. *CatCstTypeAngle*

3.3.5 CatCstTypeSymetry

Ésta es una de las herramientas más potentes y complejas dentro de las restricciones. Para su uso es necesario definir tres referencias, puesto que trabaja con tres objetos, estos son:

1. Objeto al que se le aplica la simetría.
2. Objeto respecto al que se aplica la simetría.
3. Eje de simetría

Es importante recalcar que para poder aplicar dicha restricción se necesitan tres objetos, por lo que aparte del objeto de referencia y el eje de simetría, es necesario crear un objeto arbitrario (del mismo tipo que el primero por ejemplo) al que aplicar la simetría.

```
Dim constraintmirror1 As Constraint

Dim refs1 As Reference
Set refs1 = part1.CreateReferenceFromObject(point1)

Dim refs2 As Reference
Set refs2 = part1.CreateReferenceFromObject(point11)

Dim refs3 As Reference
Set refs3 = part1.CreateReferenceFromObject(linemirror)

Set constraintmirror1 = constraints1.AddTriEltCst(catCstTypeSymmetry, refs1, refs2, refs3)
```

Código 35. *CatCstTypeSymmetry*

3.4 CATPART

La sección de CATIA denominada CATPART engloba tres módulos:

1. *Wireframe and Surface Design*
2. *PartDesign*
3. *GenerativeShapeDesign*

De dichos módulos se usarán el módulo *PartDesign*, para generar modelos sólidos, y el *Wireframe And SurfaceDesign*, para generar superficies. A continuación se va a estudiar en profundidad dichos módulos.

3.4.1 PARTDESIGN

Este módulo se centra en una sección del software CATIA en el que se disponen de una gran variedad de objetos que permiten disponer de toda la información que es necesaria para generar un modelo sólido.

La estructura del *Part Design* se puede apreciar en la Figura 39, donde los colores significan lo mismo que en la Figura 38.

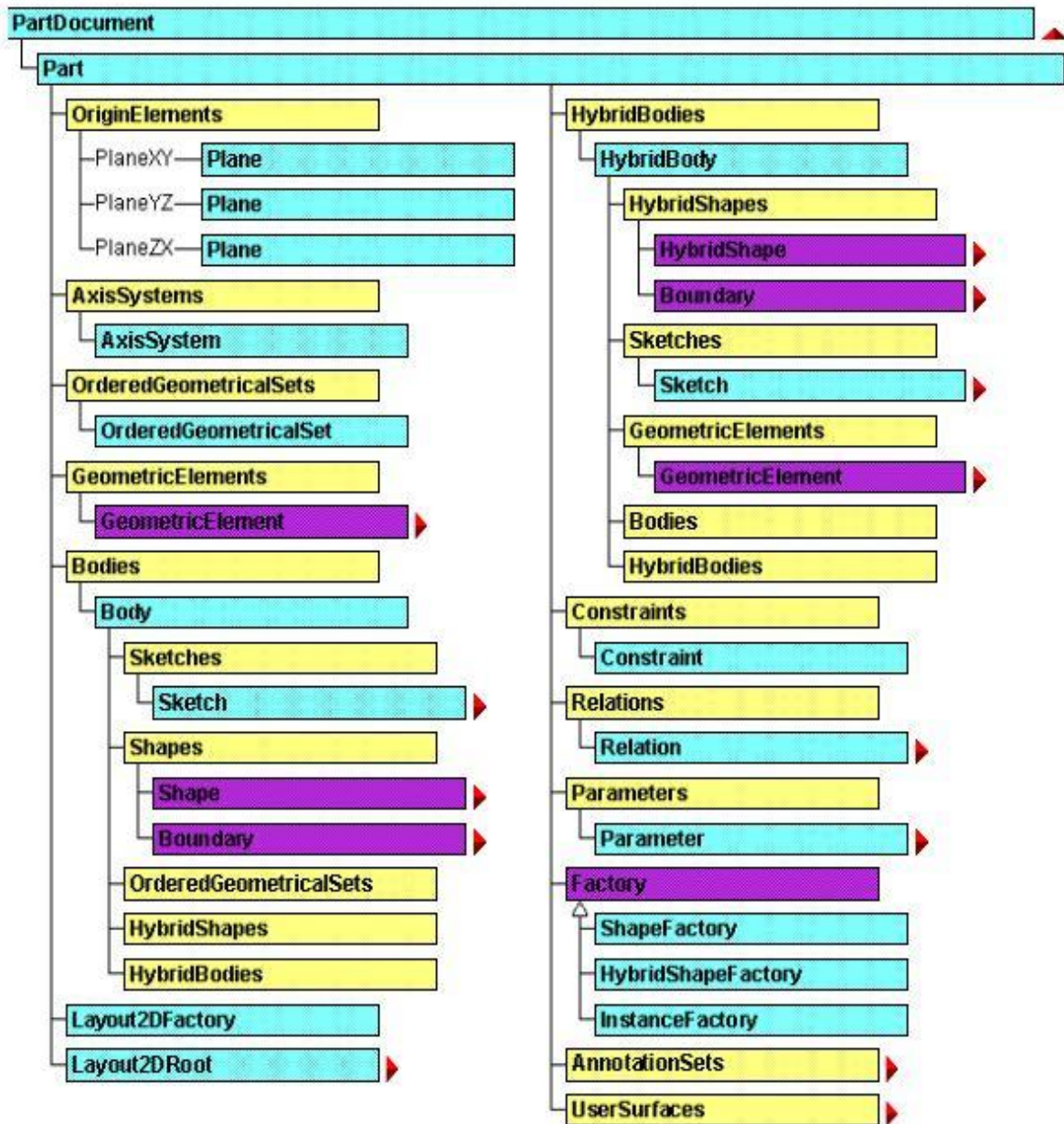
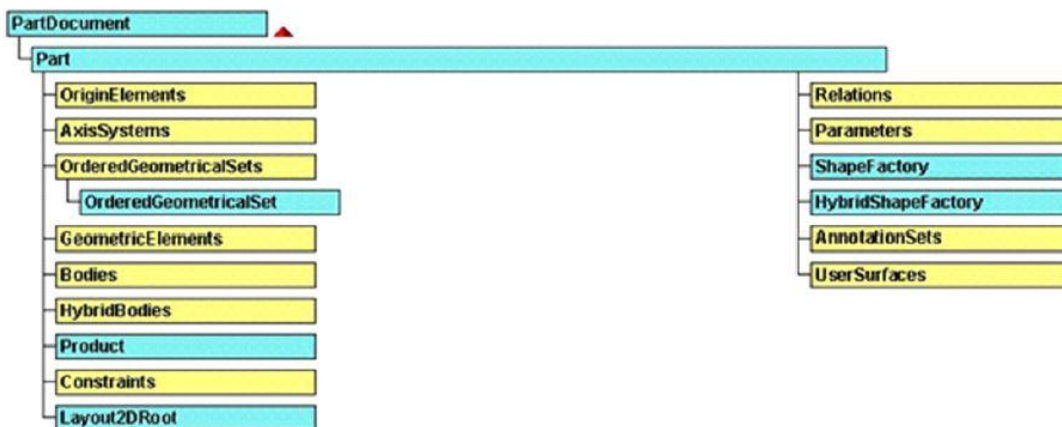


Figura 39. Estructura del módulo *PartDesign*

La estructura, como se puede apreciar, consta de obtejos abstractos (en color morado), objetos particulares (en amarillo) y más colecciones de objetos (en color azul cian). Se trata de una distribución por sectores y con una lógica determinada, en la que los objetos están agrupados según la función que realizan.

Como ya se ha mencionado anteriormente, la estructura de CATIA está muy jerarquizada, de manera que por ejemplo, para acceder a los planos de referencia, habrá que definir en primer lugar los objetos y propiedades referentes a *Plane* que contienen dichos planos (*PartDocument* → *Part* → *OriginElements*).

Figura 40. Estructura interna del *PartDocument*

En la Tabla 7 se recogen las propiedades de cada objeto que se utiliza en el módulo *PartDesign*, explicando brevemente que función tiene cada una de ellas.

Propiedad	Comando	Comentario
<i>OriginElements</i>	Dim origin1 As OriginElements Set origin1 = part1.OriginElements	Constituye el sistema de referencia absoluto del documento de planos XY, XZ e YZ.
<i>AxisSystems</i>	Dim refsist1 As AxisSystems Set refsist1 = part1.AxisSystems	Es la colección de sistemas de referencia que pueden existir en el documento <i>Part</i> .
<i>GeometricElements</i>	Dim geometric1 As GeometricElements Set geometric1 = Part1.GeometricElements	Colección de elementos geométricos 3D del <i>PartDocument</i> generadas directamente en 3D, es decir, sin la mediación del módulo <i>Sketcher</i>
<i>Bodies</i>	Dim Bodies1 As Bodies Set Bodies1 = Part1.Bodies	Colección de cuerpos sólidos dentro del <i>PartDocument</i>
<i>HybridBodies</i>	Dim HybBod1 As HybridBodies Set HybBod1 = Part1.HybridBodies	Colección de <i>OpenBodies</i> (elementos de referencia)
<i>Constraints</i>	Dim Constraints1 As Constraints Set Constraints1 = Part1.Constraints	Colección de restricciones geométricas y dimensionales del <i>PartDocument</i> .
<i>Relations</i>	DimRelations1 AsRelations SetRelations 1 = Part1.Relations	Colección de relaciones del <i>PartDocument</i> .
<i>Parameters</i>	DimParameters1 AsParameters SetParameters1 = Part1.Parameters	Colección completa de todos los parámetros del <i>PartPocument</i> .

Tabla 7. Objetos del *PartDesign*

La colección de objetos *ShapeFactory* contiene todas las operaciones que permiten:

- Pasar de un dibujo 2D a un sólido 3D.
- Realizar transformaciones sobre un sólido 3D
- Obtener un sólido a través de una superficie.

Este objeto presenta una amplia gama de operaciones que podemos encontrar en la ventana *Object Browser*. Las operaciones más significativas de este objeto las podemos ver en la Tabla 8. *Referencia [9]*

Objeto	Comentario	Contenido
<i>Shapefactory</i>	Incluye todos los métodos necesarios para poder trabajar en el módulo <i>Partdesign</i> .	<i>AddNewChamfer</i>
		<i>AddNewCircPattern</i>
		<i>AddNewCloseSurface</i>
		<i>AddNewDraft</i>
		<i>AddNewEdgeFilletWithConstantRadius</i>
		<i>AddNewEdgeFilletWithVaryingRadius</i>
		<i>AddNewFaceFillet</i>
		<i>AddNewHole</i>
		<i>AddNewLoft</i>
		<i>AddNewMirror</i>
		<i>AddNewSymmetry</i>
		<i>AddNewPad</i>
		<i>AddNewPocket</i>
		<i>AddNewRecPattern</i>
		<i>AddNewRib</i>
		<i>AddNewShaft</i>
<i>AddNewSlot</i>		
<i>AddNewStiffner</i>		
<i>AddNewthickness</i>		
<i>AddNewthicknesssurface</i>		

Tabla 8. Herramientas del *ShapeFactory*

A continuación vamos a explicar mas detalladamente las herramientas expuestas en la Tabla 8 que se han

utilizado para la ejecución del presente proyecto.

3.4.1.1 Close Surface

Esta herramienta es de muy gran utilidad ya que permite pasar del módulo *Wireframe and Surface Design* al *PartDesign*, transformando superficies sin espesor en sólidos macizos. Por este motivo, aunque el *Close Surface* pertenece al módulo del *PartDesign*, muchas veces se engloba dentro del *Wireframe and Surface Design*.

El *Close Surface* se usa para realizar cuerpos con acabados suaves, cerrando superficies y dándoles volumen. Normalmente esta herramienta complementa a otra llamada *Join* del módulo de superficies, la cual se explicará más adelante. En el Código 36 se puede ver un ejemplo de cómo utilizarla. *Referencia [9]*

```
'Cerrar lasuperficie
Dim refc As Reference
Set refc = part1.CreateReferenceFromObject(Join)
Dim myClose As CloseSurface
Set myClose = hpshapefactory.AddNewCloseSurface(refc)
part1.Update
```

Código 36. *Close Surface*

3.4.1.2 Mirror

Muchas veces a la hora de trabajar, nos encontramos que el objeto que pretendemos diseñar presenta una simetría. Para estos casos tenemos la herramienta *Mirror*, a través de la cual se puede obtener un objeto completo con tan solo diseñar la mitad de éste, resultando el proceso de diseño mucho más liviano ya que de esta forma se ahorra tener que hacer la mitad del trabajo prácticamente.

En el Código 37 se puede ver un ejemplo de realización de un *Mirror*. En él se observa que para la correcta realización de este, se necesita establecer una referencia, la del plano de simetría, que para el ejemplo expuesto es el plano YZ. Una vez creada la referencia tan solo debemos ejecutar la orden *AddNewMirror* de la colección de objetos *shapeFactory* para realizar la operación.

```
Dim hybridShapePlaneExplicit2 As HybridShapePlaneExplicit
Set hybridShapePlaneExplicit2 = originElements1.PlaneYZ

Dim reference281 As Reference
Set reference281 = part1.CreateReferenceFromObject(hybridShapePlaneExplicit2)

Dim mirror1 As Mirror
Set mirror1 = shapeFactory1.AddNewMirror(reference281)
```

Código 37. *Mirror*

3.4.1.3 Symmetry

La herramienta *Symmetry* funciona prácticamente igual que la herramienta *Mirror* anteriormente explicada. La única diferencia entre éstas, es que mientras que al hacer un *Mirror* te quedas con las dos caras, al hacer un *Symmetry* solo te quedas con la parte opuesta, es decir, con la mitad del objeto que “no” se ha diseñado. En el Código 38 se puede ver un ejemplo de cómo utilizar esta herramienta.

```
Dim hybridShapePlaneExplicit2 As HybridShapePlaneExplicit
Set hybridShapePlaneExplicit2 = originElements1.Plane.YZ

Dim reference281 As Reference
Set reference281 = part1.CreateReferenceFromObject(hybridShapePlaneExplicit2)

Dim symmetry1 As Symmetry
Set symmetry1 = shapeFactory1.AddNewSymmetry2(reference281)

Dim hybridShapeSymmetry1 As HybridShapeSymmetry
Set hybridShapeSymmetry1 = symmetry1.HybridShape
```

Código 38. *Symmetry*

3.4.2 WIREFRAME AND SURFACE DESIGN

En este subapartado se procede a dar una visión global del módulo *Wireframe and Surface Design*, que cómo es sabido, representa una de las herramientas más potentes de CATIA, proporcionando que dicho programa tenga la gran relevancia e importancia mundial que tiene.

Las superficies, son elementos que dentro del esquema *Part* se denominan *HybridBodies*, como se muestran en la Figura 39, de ahí la necesidad de activar dicho cuerpo antes de aplicar herramientas propiamente de superficies. Una vez activado los *HybridBodies*, se definirá y cargará el objeto *HybridShapeFactory*, el cual contiene todas las herramientas necesarias para trabajar con superficies.

De la misma forma que el objeto *ShapeFactory* del módulo *PartDesign* el objeto *HybridShapeFactory* también posee una amplia gama de operaciones que podemos encontrar en la ventana *Object Browser*. En la Tabla 9 podemos ver las herramientas más significativas.

Objeto	Comentario	Contenido
<i>HybridShapefactory</i>	Incluye todos los métodos necesarios para poder trabajar en el módulo <i>Partdesign</i> .	<i>AddNewFill</i>
		<i>AddNewLoft</i>
		<i>AddNewJoin</i>
		<i>AddNewLinePtPt</i>
		<i>AddNewLineTangency</i>
		<i>AddNewLineAngle</i>
		<i>AddNewBoundary</i>

		<i>AddNewCircle3Points</i>
		<i>AddNewSymmetry</i>
		<i>AddNewPlaneOffset</i>
		<i>AddNewPlaneNormal</i>
		<i>AddNewPlaneAngle</i>
		<i>AddNewPointCoord</i>

Tabla 9. Herramientas del *HybridShapefactory*

Ahora, una vez indicado que tipo de cuerpo se maneja en las superficies, en la Tabla 10 es posible ver los iconos y la descripción de las herramientas más importantes dentro de este módulo.






Nombre	Icono	Función
<i>Extrude</i>		Herramienta capaz de crear una superficie a través de la extrusión de un sketch en diferentes direcciones y dimensiones.
<i>Fill</i>		Crea superficies a partir de contornos creados previamente, útiles para rellenar superficies de manera suave.
<i>Split</i>		Capaz de eliminar partes de superficies intersecada con otras.
<i>Multi-sections Surface</i>		Herramienta similar al <i>Extrude</i> , en la cual se puede elegir varias secciones y guías para realizar la superficie.
<i>Join</i>		Une diferentes superficies dando como resultado una única superficie que engloba a las demás

Tabla 10. Herramientas de superficie

La principal diferencia entre el módulo *Wireframe and Surface Design* y el *PartDesign* es que en el primero se define y establece un *HybridBody* para poder acceder a las herramientas características de dicho módulo y en el segundo se crea un árbol de trabajo y un *body* para poder empezar a trabajar.



Figura 41. Icono *HybridBody*

El *HybridBody* es un elemento representado en el árbol de trabajo como *Geometrical Set*, con un icono como el de la Figura 41 que es una pequeña superficie. Este elemento es el que está dotado para el uso de las herramientas del módulo *Wireframe and Surface Design*, por este motivo para poder trabajar con las herramientas de superficie se debe escribir el Código 39 antes de empezar a realizar las operaciones.
Referencia [9]

```
Dim hybridBodies1 As HybridBodies
Set hybridBodies1 = part1.HybridBodies

Dim hybridBody1 As HybridBody
Set hybridBody1 = hybridBodies1.Add()

Dim hybridShapeFactory1 As HybridShapeFactory
Set hybridShapeFactory1 = part1.HybridShapeFactory
```

Código 39. Generar un cuerpo híbrido

En lo que sigue de este apartado, se van a explicar más detalladamente las herramientas utilizadas para el desarrollo de este Trabajo Fin de Grado.

3.4.2.1 Fill

Esta herramienta consiste en una de las más interesantes en el paquete de superficies de CATIA ya que nos permite crear una superficie suave utilizando como referencia contornos creados en un *sketch* o intersecciones.

En el Código 40 se indica los comandos principales para usar esta herramienta.

```
Dim ref1 As Reference
Set ref1 = part1.CreateReferenceFromObject(sketch1)

Dim Fill As HybridShapeFill
Set Fill = hybridShapeFactory1.AddNewFill
Fill.AddBound ref1

hybridBody1.AppendHybridShape Fill
part1.InWorkObject = Fill
```

Código 40. Herramienta *HybridShapeFill*

Para el correcto funcionamiento del *HybridShapeFill* es necesario definir la referencia de los bordes de la superficie mediante *AddBound*, en caso de usar otros *sketches* o más referencias como contorno se introducirán de la misma manera. Es decir, primero creando una referencia y posteriormente con *AddBound*. Finalmente se nombra el *hybridBody1* creado anteriormente como *Fill*. En el ejemplo expuesto en el Código

40 el contorno solo viene definido por un *sketch*. Referencia [9]

3.4.2.2 Multi-sections Surface

Con la herramienta *Multi-sections Surface* se puede realizar una superficie que vaya variando el perfil a lo largo de la misma, esto se consigue aportando a la herramienta diferentes perfiles en diferentes secciones y una o varias guías a través de la cual se realiza la extrusión. Los perfiles y las guías pueden ser líneas creadas en el mismo plano 3D o un *Sketch*.

```
Dim hybridShapeLoft1 As HybridShapeLoft
Set hybridShapeLoft1 = hybridShapeFactory1.AddNewLoft()

hybridShapeLoft1.SectionCoupling = 1
hybridShapeLoft1.Relimitation = 1
hybridShapeLoft1.CanonicalDetection = 2

Dim reference268 As Reference
Set reference268 = part1.CreateReferenceFromObject(hybridShapeLinePtPt1)
hybridShapeLoft1.AddGuide reference268

Dim reference269 As Reference
Set reference269 = part1.CreateReferenceFromBRepName
("WireREdge:(Wire:(Brp:(Sketch.1;142);None:(Limits1:());Limits2:());Cf1 1:());WithPermanentBody;Without
BuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", sketch1)
hybridShapeLoft1.AddSectionToLoft reference269, 1, Nothing

Dim reference270 As Reference
Set reference270 = part1.CreateReferenceFromBRepName
("WireREdge:(Wire:(Brp:(Sketch.2;1);None:(Limits1:());Limits2:());Cf1 1:());WithPermanentBody;WithoutBu
ildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", sketch2)
hybridShapeLoft1.AddSectionToLoft reference270, 1, Nothing

body1.InsertHybridShape hybridShapeLoft1
part1.InWorkObject = hybridShapeLoft1
part1.Update
```

Código 41. *Multi-sections Surface*

Como podemos ver en el Código 41, en este caso se activa y establece la herramienta en primer lugar para posteriormente crear y establecer las referencias que se usaran como guías y secciones las cuales se incluyen a la herramienta *Multi-sections Surface*, realizando en las último lugar la carga y actualización del cuerpo sobre el que se está trabajando la operación realizada.

3.4.2.3 Join

El comando *Join* resulta muy interesante, ya que permite transformar más de una superficie en una sola, lo cual es de gran utilidad para:

- Transformar la superficie resultante del *Join* en un sólido macizo ya sea usando *Thick Surface* o *Close Surface*.
- Añadir un material a toda la superficie.
- Exportar dicha superficie como una única a otros programas, ahorrándose establecer las *constraints* para unirlos.

En el Código 42 se puede observar que en la herramienta *Join* se necesitan como argumentos dos referencias, las cuales se corresponden con las superficies que se quieren unir.

```
'Unimos dos superficies cualquiera, en este caso un extrude y un fill
```

```
Dim refj1 As Reference  
Set refj1 = part1.CreateReferenceFromObject(Fill1)  
  
Dim refj2 As Reference  
Set refj2 = part1.CreateReferenceFromObject(Extrude1)  
  
Dim Join As HybridShapeAssemble  
Set Join = hybridShapefactory1.AddNewJoin(refj1, refj2)  
  
hybridBody1.AppendHybridShape Join  
part1.InWorkObject = Join  
part1.Update
```

Código 42. Unión de dos superficies

Si el *Join* que se desea realizar contiene más de dos superficies, se debe repetir el Código 43 para cada superficie que se desea agregar al *Join* antes de las últimas líneas del Código 42. *Referencia [9]*

```
Dim refj3 As Reference  
Set refj3 = part1.CreateReferenceFromObject(Fill2)  
  
hybridShapeAssemble1.AddElement refj3
```

Código 43. Añadir superficies al *Join*

3.4.2.4 Symmetry

Al igual que en el módulo *PartDesign*, en el módulo *Wireframe and Surface Design* también se dispone de una herramienta para realizar operaciones de simetría con la diferencia de que en este módulo con la misma herramienta podemos realizar las operaciones que se realizan con el *Mirror* y el *Symmetry* en el otro módulo. Además, para poder realizar esta operación, se necesitan dos referencias. Éstas son:

1. El plano de simetría.
2. El objeto al que se le va a aplicar dicha simetría.

Como podemos observar, a diferencia de las herramientas del *PartDesign*, aquí si puedes decidir a que objetos se le realiza la simetría. En el Código 44 se ilustra un ejemplo de cómo realizar esta operación.

```
Dim reference279 As Reference
Set reference279 = part1.CreateReferenceFromObject(hybridShapeAssemble1)

Dim hybridShapePlaneExplicit2 As HybridShapePlaneExplicit
Set hybridShapePlaneExplicit2 = originElements1.Plane.YZ

Dim reference280 As Reference
Set reference280 = part1.CreateReferenceFromObject(hybridShapePlaneExplicit2)

Dim hybridShapeSymmetry1 As HybridShapeSymmetry
Set hybridShapeSymmetry1 = hybridShapeFactory1.AddNewSymmetry(reference279, reference280)

hybridShapeSymmetry1.VolumeResult = False

body1.InsertHybridShape hybridShapeSymmetry1

part1.InWorkObject = hybridShapeSymmetry1
```

Código 44. *Symmetry*

3.5 KNOWLEDGEWARE

El módulo *Knowledgeware* es la herramienta de parametrización de CATIA la cual es indispensable para la ejecución del presente proyecto.

A lo largo de este apartado se va a explicar como se utilizan en la programación de macros las herramientas explicadas con anterioridad en el apartado 1.5. que pertenecen al submódulo *Knowledge Advisor* que esta dentro del módulo *Knowledgeware*.

3.5.1 Parámetros

Como ya sabemos existen infinidad de tipos de parámetros (integradores, variables, constantes de longitud, constantes de volumen...). CATIA los engloba todos en un mismo esquema, el de los *Parameters*, sea cual sea el tipo de parámetro del que se trate.

Para crear, chequear o borrar un parámetro es preciso utilizar los comandos del Código 45.

```
Dim part1 As document
Set part1 = CATIA.ActiveDocument

Dim parameters1 As Parameters
Set parameters1 = part1.Part.Parameters
```

Código 45. *Parameters*

Como se puede ver, para empezar a aplicar la herramienta en CATIA, se debe activar el documento que está abierto (a de ser *CatPart*) y posteriormente se entra en los parámetros refiriéndose a ellos con la orden *Parameters*. El esquema al que se ciñe CATIA es el mostrado en Figura 42.

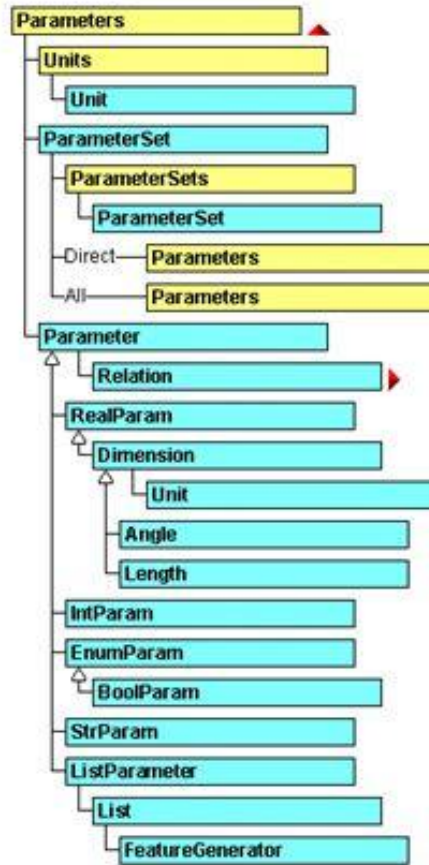


Figura 42. Esquema de parámetros

Los *Scripts* del módulo de parametrización sólo son utilizables en un documento CATPart, así que cualquier código deberá asegurar ese carácter del documento activo para así ahorrar errores posteriores.

Como se comentó anteriormente, existen multitud de tipos de parámetros, pero para la ejecución de este proyecto solo han sido necesarios dos tipos:

1. *Integer*: que no tiene unidades, simplemente contiene un valor.
2. *Dimension*: que a parte de contener un valor dicho valor va a tener unas unidades específicas, las que nosotros queramos. De este tipo solo se utilizarán los de longitud, en *mm*, y ángulo, en *deg*.

El comando a utilizar para la creación del parámetro dependerá del tipo de parámetro que queramos crear, para crear los que se utilizan en la macro desarrollada se debe seguir los comandos que aparecen en el Código 46.

```
Dim NACA34T As Parameter
Set NACA34T = parameters.CreateInteger("NACA34T", 0)

Dim Cuerda As Parameter
Set Cuerda = parameters.CreateDimension("Cuerda", "LENGTH", 0)

Dim Incidencia As Parameter
Set Incidencia = parameters.CreateDimension("Incidencia", "ANGLE", 0)
```

Código 46. Creación de parámetros

En el Código 46 se crea un parámetro del tipo *Integer* y dos del tipo *Dimension*, uno de longitud y otro de ángulo. Como se puede observar después del comando utilizado para la creación del parámetro se debe incluir entre paréntesis y comillas el nombre que tendrá en CATIA, la dimensión (si es del tipo *Dimension*) y el valor inicial que tendrá este.

En la macro desarrollada el valor del parámetro que se crea será un dato de entrada, de hecho, todos los datos de entrada serán parámetros, así que a continuación se va a explicar como se consigue asignar un valor de entrada a un parámetro.

En el Código 47 se puede ver que el dato de entrada se introduce a través de un *TextBox* previamente colocado en la interfaz con el usuario. El valor escrito en el *TextBox* se asigna a una variable para posteriormente introducir el valor de dicha variable al parámetro a través de la orden *.Value*.

```
NACA1 Valor = TextBox1
NACA1.Value = NACA1 Valor
```

Código 47. Asignación de un valor a un parámetro

Algunas veces, un parámetro no puede tener cualquier valor, si no que debe estar dentro de un margen. Por ejemplo, en nuestro caso, el parámetro NACA1 es el primer dígito del perfil NACA por lo que solo puede ser un número del 0 al 9. Observando esto, resultaría interesante implementar un trozo de código que compruebe que dicho parámetro esta dentro del margen al principio del código de forma que si el valor introducido esta fuera del margen para el que ha sido creado se pare el programa y no se produzca ningún error en su ejecución. Para ello se introduce el Código 48 que simplemente consiste en un sencillo condicional que comprueba el valor y al cual se le ha añadido un mensaje de advertencia el cual le saldrá al usuario si el valor introducido es erróneo consiguiendo de esta forma que el usuario pueda identificar rápidamente en que valor introducido se ha equivocado ya que en la macro desarrollada existen varios. *Referencia [9]*

```
If NACA1 Valor > -1 And NACA1 Valor < 10 Then
Else
MsgBox "El valor introducido en NACA 1 en perfil encastre debe estar comprendido entre 0 y 9."
End
' Este End se pone para parar el programa si se encuentra el error
End If
```

Código 48. Comprobación del valor de los parámetros

3.5.2 Fórmulas

Las fórmulas tal y como se vió en el apartado 1.5.3 son un enclave fundamental en la parametrización. A continuación en la Figura 43 se muestra un esquema de las herramientas referentes a *Relations*, fórmulas, en el que se puede ver de donde cuelga cada uno de los comandos que se usan en este módulo. Además, en la Figura 42 se observa que las fórmulas cuelgan de parámetros, es decir, que antes de introducir cual fórmula se deberá asociar a un parámetro. Posteriormente CATIA mostrará la relación en el árbol tal como se vió previamente y trabajará con el resultado.

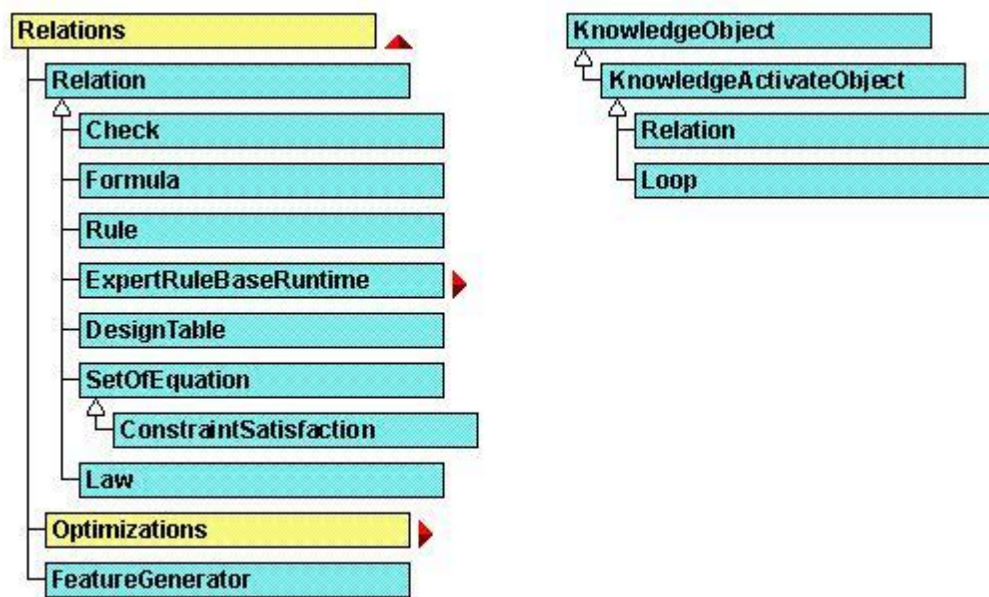


Figura 43. Esquema *Relations*

Para activar las fórmulas se deberá escribir previamente el Código 49 para poder trabajar con ellas.

```

Dim parameters As parameters
Set parameters = part1.parameters

Dim relations As relations
Set relations = part1.relations
    
```

Código 49. *Relations*

Como bien se sabe se puede relacionar parámetros y restricciones con fórmulas. Para la realización de este trabajo, lo que se requiere es de relacionar *constraints* con fórmulas que contengan los parámetros, por este motivo solo vamos a explicar este último. Para ello se debe seguir los siguientes pasos:

1. Crear los parámetros necesarios para la fórmula, los cuales se crean siguiendo el procedimiento ya explicado en el apartado 3.5.1.
2. Crear la constraint a la cual queremos vincular la fórmula siguiendo el procedimiento del apartado 3.3.

3. Crear una variable que contenga el valor de dicha *constraint*.
4. Crear la fórmula y vincularla a la variable anterior.

Se debe tener especial cuidado en que el parámetro y la constraint creada tengan las mismas unidades ya que si estas no coinciden la macro no se ejecutará correctamente.

En el Código 50 se ilustra un ejemplo de los pasos a llevar a cabo para la creación de dos fórmulas con distintas unidades, la primera en milímetros y la segunda en grados. Por último, en dicho código se ha omitido la parte donde se crea los parámetros y se establece las referencias.

```
Dim constraint2 As Constraint
Set constraint2 = constraints1.AddBiEltCst(catCstTypeDistance, reference3, reference4)

constraint2.Mode = catCstModeDrivingDimension

Dim length2 As Length
Set length2 = constraint2.Dimension

Dim formula2 As Formula
Set formula2 = relations.CreateFormula("Punto borde de ataque", "Borde de ataque", length2, "Cuerda")

Dim constraint3 As Constraint
Set constraint3 = constraints1.AddBiEltCst(catCstTypeAngle, reference5, reference2)

constraint3.Mode = catCstModeDrivingDimension

constraint3.AngleSector = catCstAngleSector0

Dim angle3 As Angle
Set angle3 = constraint3.Dimension

Dim formula3 As Formula
Set formula3 = relations.CreateFormula("Linea referencia vertical Incidencia", "Incidencia", angle3,
"Incidencia")
```

Código 50. Crear fórmula

La orden utilizada para crear una fórmula es *CreateFormula* seguida de unos paréntesis donde se incluye toda la información sobre la relación. Dentro de los paréntesis la lógica a seguir es: ("Nombre en el árbol", "Comentarios", constraint/parámetro relacionado, "fórmula en sí"). Los comentarios escritos se mostrarán cuando seleccionemos con el puntero la fórmula sobre el árbol, estos pueden ser textos explicativos o bien indicativos para próximos pasos. *Referencia [9]*

3.5.3 Reglas

La herramienta *Rule* al igual que la herramienta *Formula* están contenidas en *Relations* que a su vez está contenida en *Parameters* tal y como se puede ver en la Figura 42 y la Figura 43. Es decir, que antes de introducir una regla también se deberá asociar un parámetro, esto se consigue implementando las líneas de código que aparecen en el Código 49. Posteriormente CATIA mostrará las reglas en el árbol dentro de la pestaña *Relations* junto a las fórmulas como se muestra en la Figura 44.



Figura 44. Vista del módulo *Relations* en el árbol

Para crear una *Rule* se procede de forma similar a una fórmula. En el Código 51 se puede ver que tenemos que escribir para poder hacerlo.

```
Dim rule1 As Rule
Set rule1 = relations.CreateProgram("rule1", "", "")
```

Código 51. Crear una *Rule*

Como se puede apreciar, en este caso se hace uso de la orden *.CreateProgram* seguida de unos paréntesis que contienen en primer lugar el nombre de la *Rule*, en segundo lugar los comentarios y en tercer lugar la rule en sí.

En el caso de este proyecto la regla creada es tan larga que se opta por dejar el hueco de la rule en blanco y escribir la regla en sí en otra línea de código. Además, es importante destacar que el lenguaje de programación que se está usando no permite escribir la cantidad de argumentos que se quiera dentro de la regla, si no que este número está limitado por la cantidad de bits del propio lenguaje.

```
rule1.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _
"if (20/1000*Cuerda) <= (Cuerda*(NACA2/10)) PartBody\Sketch.1\Offset.7\Offset = (20/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(20/1000))))    else
PartBody\Sketch.1\Offset.7\Offset = (20/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-0.1015*(20/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(20/1000))))" & vbCrLf & _
"if (20/1000*Cuerda) <= (Cuerda*(NACA2/10)) PartBody\Sketch.1\Offset.37\Offset =
(Cuerda+((NACA1/100)*(20/1000*Cuerda)*(2*(NACA2/10)-
20/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(20/1000))))    else
PartBody\Sketch.1\Offset.37\Offset = (Cuerda+((NACA1/100)*(Cuerda-20/1000*Cuerda)*(-
2*(NACA2/10)+20/1000+1)/(1-NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1/100)/((1-(NACA2/10)**2)*((NACA2/10)-(20/1000))))"
```

Código 52. Orden *Modify*

En el Código 52 se enseña como se añade una regla a la *Rule*. Además, se observa un par de cosas que es importante que se comprendan para el correcto funcionamiento de esta herramienta:

1. Todo lo que vaya escrito entre comillas es lo que irá dentro de la *Rule*.
2. Al final de cada línea (sin contar la última) se escribe `& vbCrLf & _` para poder continuar escribiendo en la línea de abajo y que dicha línea aparezca también en la línea de abajo dentro de la *Rule*.

Con todo esto, el aspecto que presentará la *Rule* creada es el que aparece en la Figura 45.

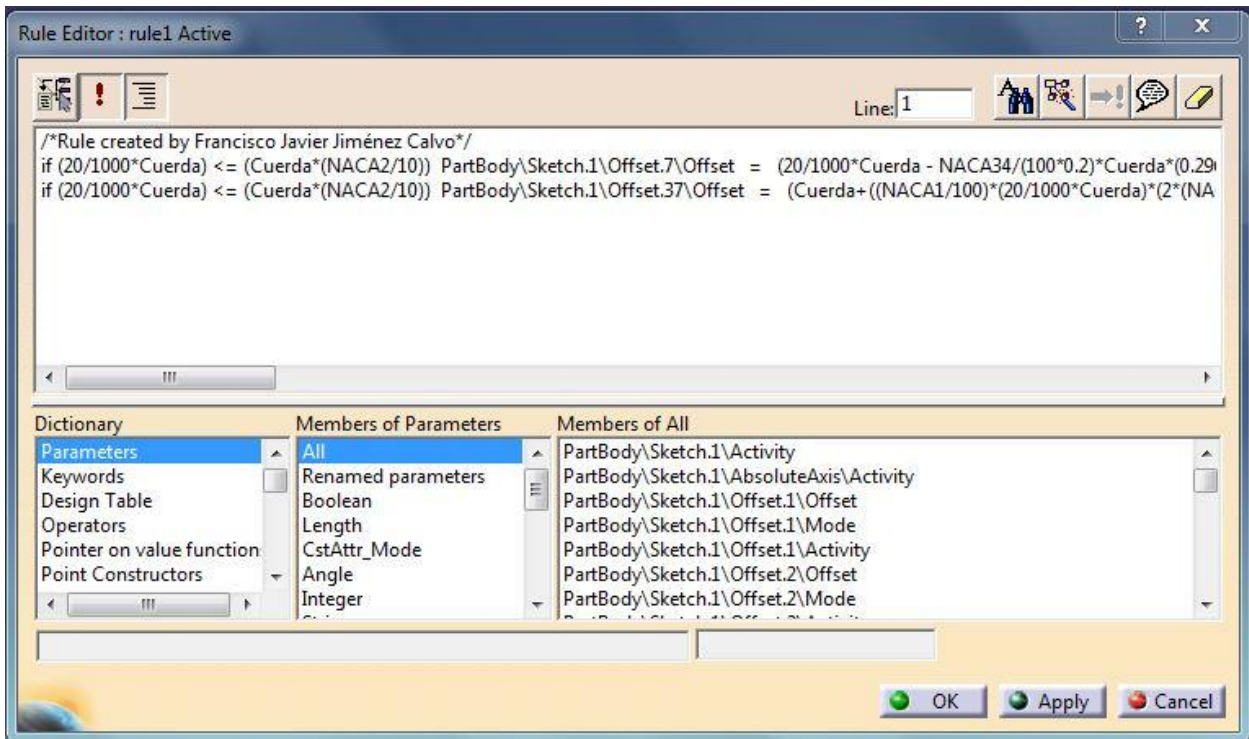


Figura 45. *Rule Editor*

4 APLICACIÓN: ENTORNO DE PROGRAMACIÓN

*Quando hay una tormenta los pajaritos se esconden,
pero las águilas vuelan más alto.*

- Mahatma Gandhi -

En este capítulo se van a desarrollar los procedimientos que se han llevado a cabo para programar el diseño de perfiles y alas parametrizadas con nomenclatura NACA de 4 dígitos.

En primer lugar se crea el formulario principal del programa el cual permite elegir entre las opciones de crear un perfil, crear un ala o salir del programa. En la Figura 46 se puede ver una imagen de este formulario.

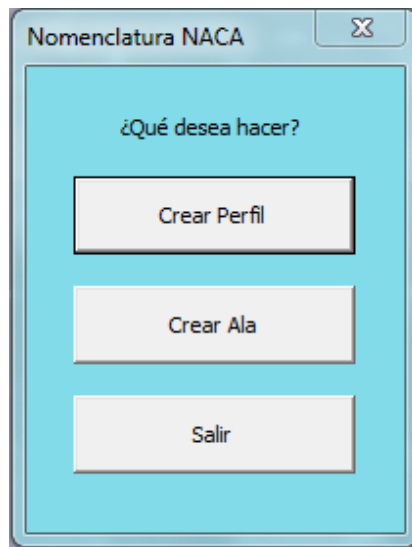


Figura 46. Interfaz principal del programa

En la interfaz principal aparecen tres *CommandButton*: “Crear Perfil”, “Crear Ala” y “Salir”. Los *CommandButton* “Crear Perfil” y “Crear Ala” están programados para que cuando se haga click sobre ellos, el programa abra automáticamente el formulario correspondiente a la opción seleccionada. Para ello se utiliza el Código 53. Por el contrario, el *CommandButton* “Salir” esta programado para parar y cerrar la ejecución del

programa, esto se consigue implementando el Código 54.

```
Private Sub CommandButton1_Click()

Load NombreUserform
Unload Me
NombreUserform.Show

End Sub
```

Código 53. Llamada a los *UserForms* con un *CommandButton*

```
Private Sub CommandButton1_Click()

Unload Me

End Sub
```

Código 54. Salir del programa con un *CommandButton*

En la Figura 47 se puede ver el *UserForm* asociado al *CommandButton* “Crear Perfil” y el asociado al *CommandButton* “Crear Ala”. Como se puede ver en ambos *UserForm* aparece un *CommandButton* llamado “Atrás” el cual esta programado para que al hacer click sobre él se vuelva a abrir automáticamente el formulario principal del programa desarrollado. Apareciendo de esta forma un flujo de llamadas entre los distintos formularios. Este flujo también se puede ver en la Figura 47.

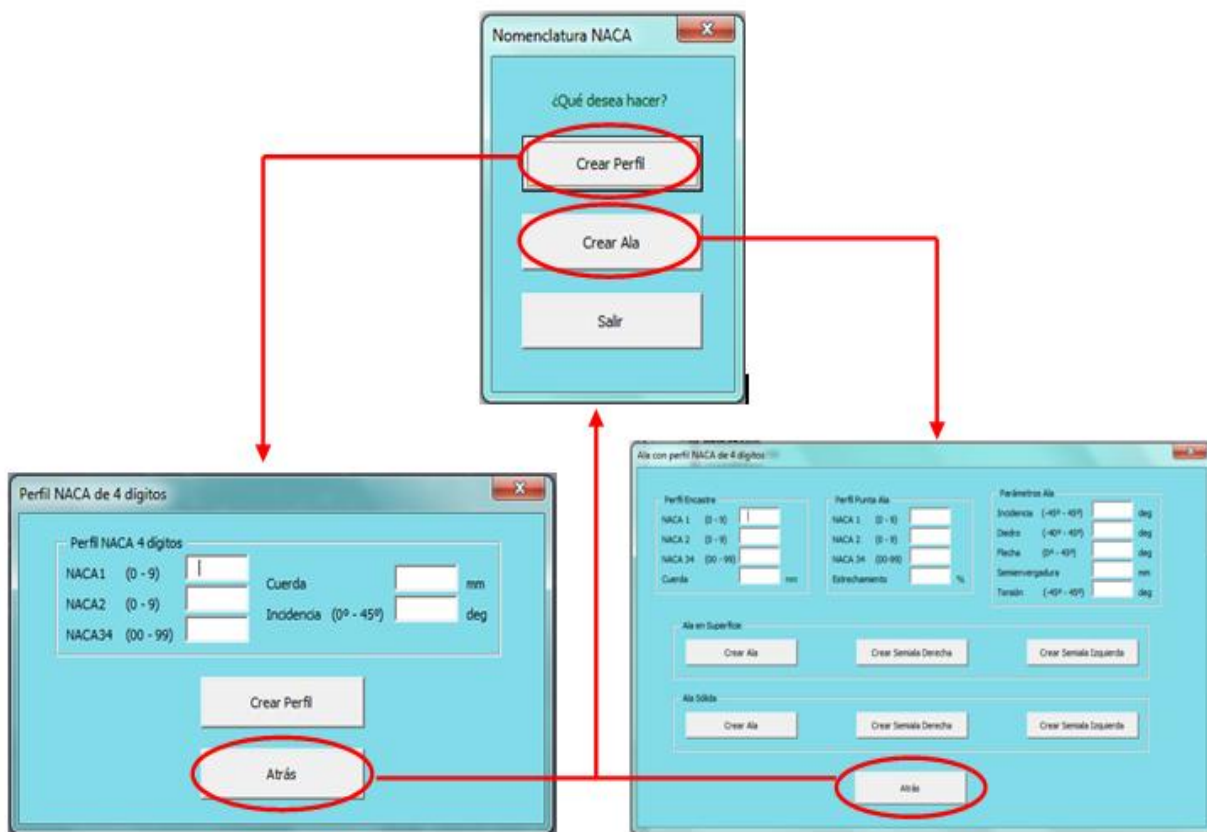


Figura 47. Flujo *UserForms*

4.1 Parámetros de entrada en los UserForms

Dentro de los correspondientes *UserForms* para la creación de un perfil o ala parametrizada con nomenclatura NACA de 4 dígitos, lo primero que se debe hacer es introducir los valores de los parámetros necesarios de forma manual por lo que se incluye un *TextBox* dentro del *UserForm* para cada parámetro que se necesita. Estos *TextBoxs* se agrupan en diferentes en diferentes *Frames* para distinguir a que afecta el parámetro que estamos introduciendo. Además, junto al *TextBox* se introduce a la izquierda del mismo el nombre del parámetro y el rango de aplicación, si lo tiene, con un *Label* y a la derecha las unidades, en el caso de que las tuviese, con otro *Label*. En la Figura 48 y la Figura 49 se puede ver más claramente de lo que se está hablando.

Figura 48. Parámetros formulario Crear Perfil

Figura 49. Parámetros formulario Crear Ala

Aún teniendo el rango de aplicación en la interfaz del usuario, es posible cometer el error de introducir algún dato fuera de rango. Por este motivo, se ha programado un aviso de error de forma que el programa sea capaz de avisar de que se ha cometido un error introduciendo algún parámetro con un mensaje en una ventana emergente donde comunica que parámetro se ha introducido mal y te vuelve a recordar el rango de operación del mismo. Cómo introducir este tipo de mensajes se explicó en el apartado 3.5.1. En la Figura 50 se muestra la ventana emergente de uno de los avisos programados donde al pulsar el botón aceptar, el programa se parará y cerrará por lo que si se quiere volver a realizar alguna operación se deberá volver a ejecutar el programa desde el principio.

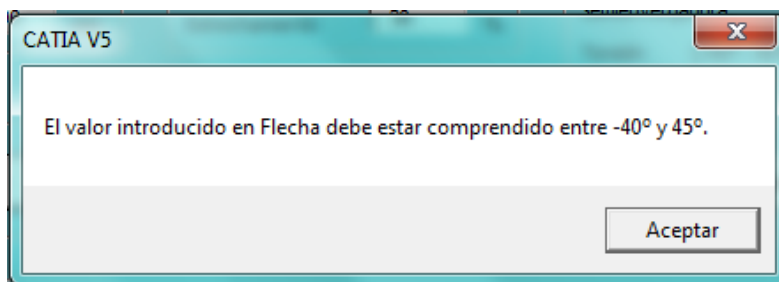


Figura 50. Mensaje de error

4.2 Estructura de la Aplicación

Para la creación de los perfiles y alas con nomenclatura NACA de 4 dígitos se incluye dentro de cada *CommandButton* de los formularios correspondientes las líneas de código necesarias para realizar con éxito la operación requerida.

En los Anexos del 2 al 9 se encuentran todas las líneas de códigos necesarias para la correcta ejecución del programa estructuradas por bloques. La decisión de estructurar el código por bloques radica en que las operaciones que se realizan al comienzo de la mayoría de las opciones posibles son las mismas, por lo que en vez de adjuntar el código del programa completo, el cual ocuparía muchísimo espacio, se hizo por bloques. En la Tabla 11 se puede ver una estructura jerarquizada del programa desarrollado junto con los anexos que incluyen las diferentes opciones finales.

<i>CommandButton</i>	<i>Frame</i>	<i>CommandButton</i>	Código
Crear perfil	-	Crear Perfil	Anexo 2
Crear Ala	Ala en Superficie	Crear Semiala Izquierda	Anexo 3 Anexo 4
		Crear Semiala Derecha	Anexo 3 Anexo 5
		Crear Ala	Anexo 3 Anexo 6
	Ala Sólida	Crear Semiala Izquierda	Anexo 3 Anexo 7
		Crear Semiala Derecha	Anexo 3 Anexo 8
		Crear Ala	Anexo 3 Anexo 9

Tabla 11. Estructura jerarquizada del programa

4.3 Estructura del Código

Como bien se comenta en el apartado anterior, el código del programa se adjunta por bloques en el ANEXO del presente trabajo pero en la aplicación no, las líneas de código que se repiten en los distintos *CommandButton* deben estar repetidas dentro de cada uno de ellos. Sin embargo, CATIA permite crear módulos que almacenan una operación para que no se tenga que repetir las líneas de código de dicha operación si ésta se repite constantemente a lo largo del código de forma que solo se tenga que hacer una llamada al *Module* en la parte del código donde se quiere que se realice la operación. Esto solo es posible cuando se trabaja con un tipo de variable fija pero como en la aplicación desarrollada se utilizan variables de tipo variable como los parámetros, no es posible agrupar todo el ANEXO 3, que es el que se repite, en un *Module*, teniendo que escribir todas las líneas de código dentro de cada *CommandButton*.

Dentro de cada ANEXO perteneciente al código de la aplicación, contiene líneas de código que realizan diferentes operaciones. A continuación se va a comentar que operaciones incluye cada uno.

- ANEXO 2 → Contiene todas las líneas de código necesarias para la creación de un perfil NACA de 4 dígitos 2D. Para ello se ha programado el procedimiento que se desarrolló en el apartado 1.6, salvo algunas diferencias que se comentan al final de este apartado, haciendo uso de las herramientas explicadas en la sección 3. En la Figura 51 se puede ver un ejemplo del resultado de crear un perfil.



Figura 51. Ejemplo del resultado final de crear un perfil

- ANEXO 3 → Contiene las líneas de código de parte de las operaciones necesarias para crear un ala. Dentro de él se tiene el código de las operaciones donde se crea tanto el perfil del encastre como el de la punta del ala así como la línea que une todo los bordes de ataque del ala. Este ANEXO representa la estructura principal para crear un ala ya que una vez realizada estas operaciones ya solo quedará extruir los perfiles utilizando la línea de borde de ataque como guía. El resultado de ejecutar solamente estas líneas de código y no el programa completo es el representado en la Figura 52.

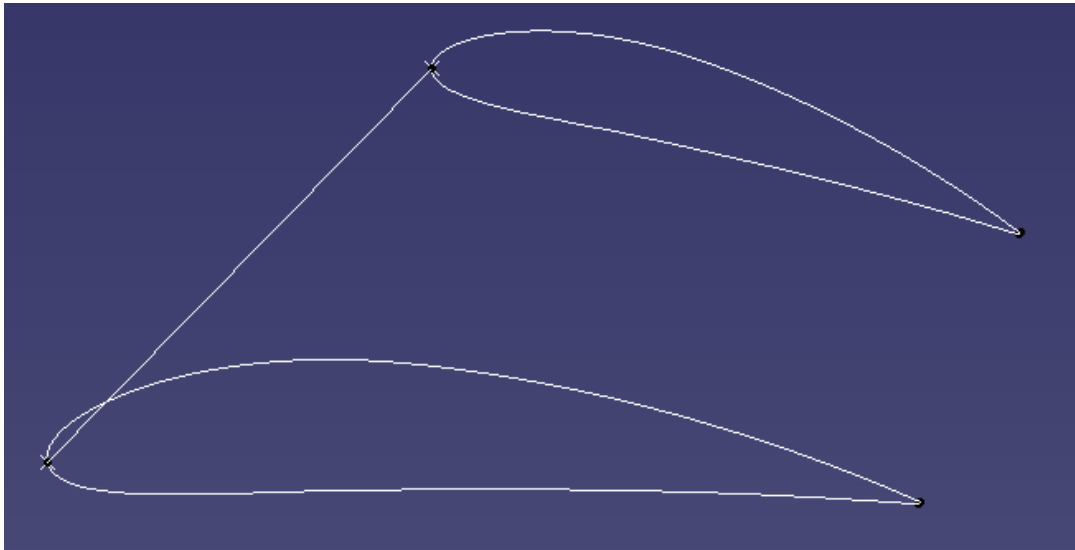


Figura 52. Resultado de ejecutar el ANEXO 3

- ANEXO 4 y 7 → Contiene las líneas de código necesarias para crear la semiala izquierda en superficie y sólida respectivamente. En ambos ANEXOS se tiene prácticamente el mismo código, la mayor diferencia es que en el ANEXO 7 se incluye la operación *Close Surface* para convertir la semiala, que al principio estará en superficie, en sólida. Esto es así ya que como se explicó en el apartado 1.6 resulta conveniente realizar todas las operaciones posibles dentro del módulo *Wireframe and Surface Design* realizando en último lugar las operaciones correspondientes al módulo *Part Design* que no se hayan podido realizar en el anterior módulo. En este caso solo se utiliza la herramienta *Close Surface* de dicho módulo. En la Figura 53 se puede ver un ejemplo del resultado final de crear una semiala izquierda. Dicho resultado es aparentemente el mismo para semialas en superficie y sólidas, la única diferencia entre ambas es que una esta hueca y la otra no.

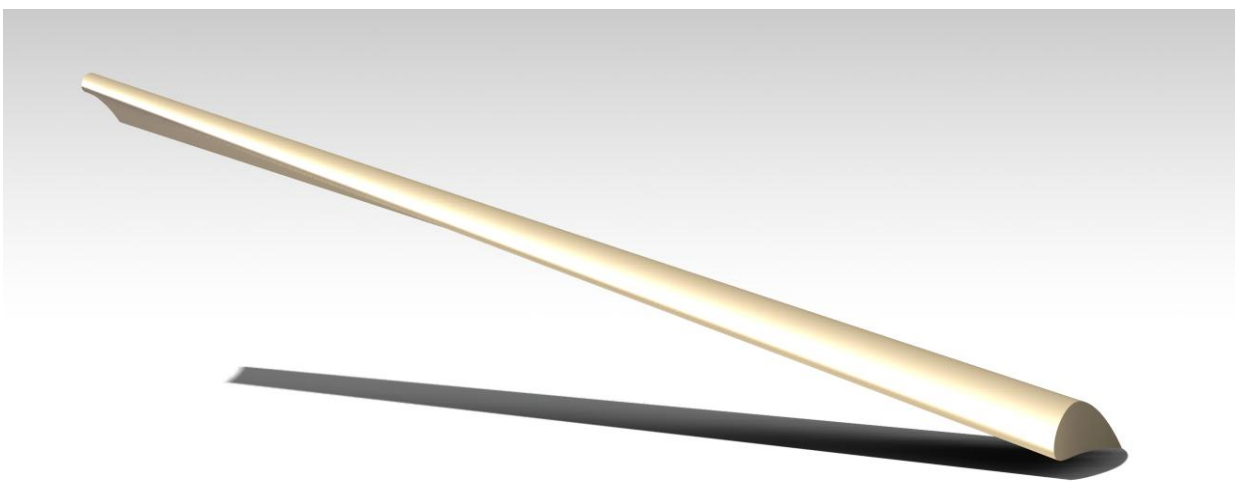


Figura 53. Ejemplo del resultado final de crear una semiala izquierda

- ANEXO 5 y 8 → Contiene las líneas de código necesarias para crear la semiala derecha en superficie y sólida respectivamente. Ambos ANEXOS son prácticamente iguales, en ambos se utiliza la programación de la creación de la semiala izquierda para obtener la semiala derecha. Esto es así ya que durante la realización de la aplicación se obtuvo en primer lugar la semiala izquierda. A continuación se van a comentar las mayores diferencias que existen entre ambos ANEXOS.
 1. En el ANEXO 8, al igual que en el ANEXO 7 anteriormente y por el mismo motivo, se incluye la operación *Close Surface* para convertir la semiala, que al principio estará en superficie, en sólida.
 2. Las operaciones de simetría son distintas en cada caso ya que cada módulo, *Wireframe and Surface Design* y *Part Design* respectivamente, contiene una herramienta de simetría distinta como se vio en el apartado 3.4. Además, se tiene que tener cuidado al utilizar la simetría dentro del módulo *Part Design* ya que a diferencia del otro módulo, en este existe diferentes herramientas en función del resultado final que se quiera obtener. Para este caso se elige la herramienta que te permite quedarte solamente con la semiala simétrica a la que se ha diseñado, ocultando esta última. En el caso de la simetría en superficie tan solo basta indicar que resultado final se quiere tener, si ambas partes, la diseñada y la simétrica, o solo la simétrica.

En la Figura 54 se puede ver un ejemplo del resultado final de crear una semiala derecha. Al igual que antes, dicho resultado es aparentemente el mismo para semialas en superficie y sólidas, la única diferencia entre ambas es que una esta hueca y la otra no.

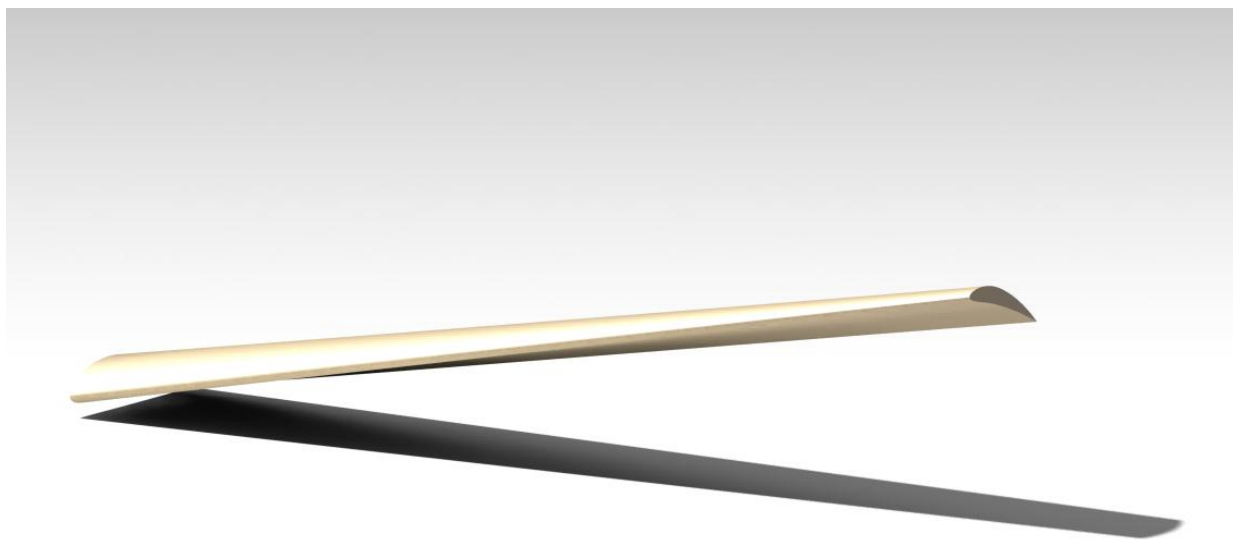


Figura 54. Ejemplo del resultado final de crear una semiala derecha

- ANEXO 6 y 9 → Contiene las líneas de código necesarias para crear la ala completa en superficie y sólida respectivamente. Estos códigos son prácticamente iguales a los de los ANEXOS 5 y 8 respectivamente, la única diferencia entre ellos es a la hora de realizar la simetría.

- En la ala en superficie la operación de simetría es la misma que en el ANEXO 5 pero en este caso no se debe ocultar la semiala izquierda y después de realizar esta operación se debe hacer un *Join* que incluya ambas semialas para formar una sola ala.
- En la ala sólida la operación de simetría es distinta que en el ANEXO 8 ya que en este caso se debe utilizar la operación que nos permite quedarnos con ambas semialas y no la simétrica a la que se realiza la operación.

En la Figura 55 se puede ver un ejemplo del resultado final de crear un ala. Como en los casos anteriores, dicho resultado es aparentemente el mismo para alas en superficie y sólidas, la única diferencia entre ambas es que una esta hueca y la otra no.



Figura 55. Ejemplo del resultado final de crear un ala

Ahora se va a proceder a mencionar que cosas se realizan de forma diferente al apartado 1.6 donde se explicó los pasos a seguir para crear un perfil y un ala parametrizada con nomenclatura NACA de 4 dígitos.

En primer lugar, para la creación únicamente de un perfil, la semienvargadura del ala no es un dato de entrada, por lo que el punto del borde de ataque se coloca a una longitud y altura igual a la cuerda del perfil en vez de a la semienvargadura. Esto no conllevará posibles problemas ya que en este caso, al no tratarse de un ala, no hay que parametrizar ni el diedro ni la flecha, razón por la cual se decidió poner el borde de ataque a tal distancia.

En segundo lugar, a la hora de programar, como bien vimos en el apartado 3.2.3, antes de crear un *Spline* se debe crear los *ControlPoints* a través de los cuales se definirá la curva, por lo que en el programa desarrollado lo que se hace es crear y acotar los *ControlPoints* en vez de crear y acotar los puntos con los que posteriormente se creaba el *Spline*. De esta forma se ahorra gran parte del trabajo para dibujar el perfil ya que

si no tendríamos que definir en vez de 60 puntos de control, 120 de los cuales la mitad serían puntos y la otra mitad puntos de control. Sin embargo, el punto que define el borde de ataque si se ha creado como punto y como punto de control, ya que el primero nos hace falta para poder definir las líneas de referencia y la línea que une todos los bordes de ataque del ala.

Tal y como se comento en el apartado 3.5.3, en el lenguaje de programación que se esta usando no se permite escribir la cantidad de argumentos que se quiera dentro de una regla, si no que el número de argumentos está limitado por la cantidad de bits del propio lenguaje. Por este motivo no se puede crear una *Rule* que contenga todas las líneas de código necesarias para dibujar el perfil, de hecho, el programa soporta poco mas de una regla para definir las *Constraints* de un punto. Por esta razón en el código del programa se crea una *Rule* por cada *ControlPoint* haciendo un total de 60 reglas para crear un perfil y 120 para un ala.

Para finalizar esta sección, se ha diseñado un diagrama de flujo de la aplicación el cual podemos ver en la Figura 56 que se encuentra en la pagina 78. En él se puede visualizar a grandes rasgos todo lo explicado en esta sección de como funciona la aplicación desarrollada en este Trabajo Fin de Grado.

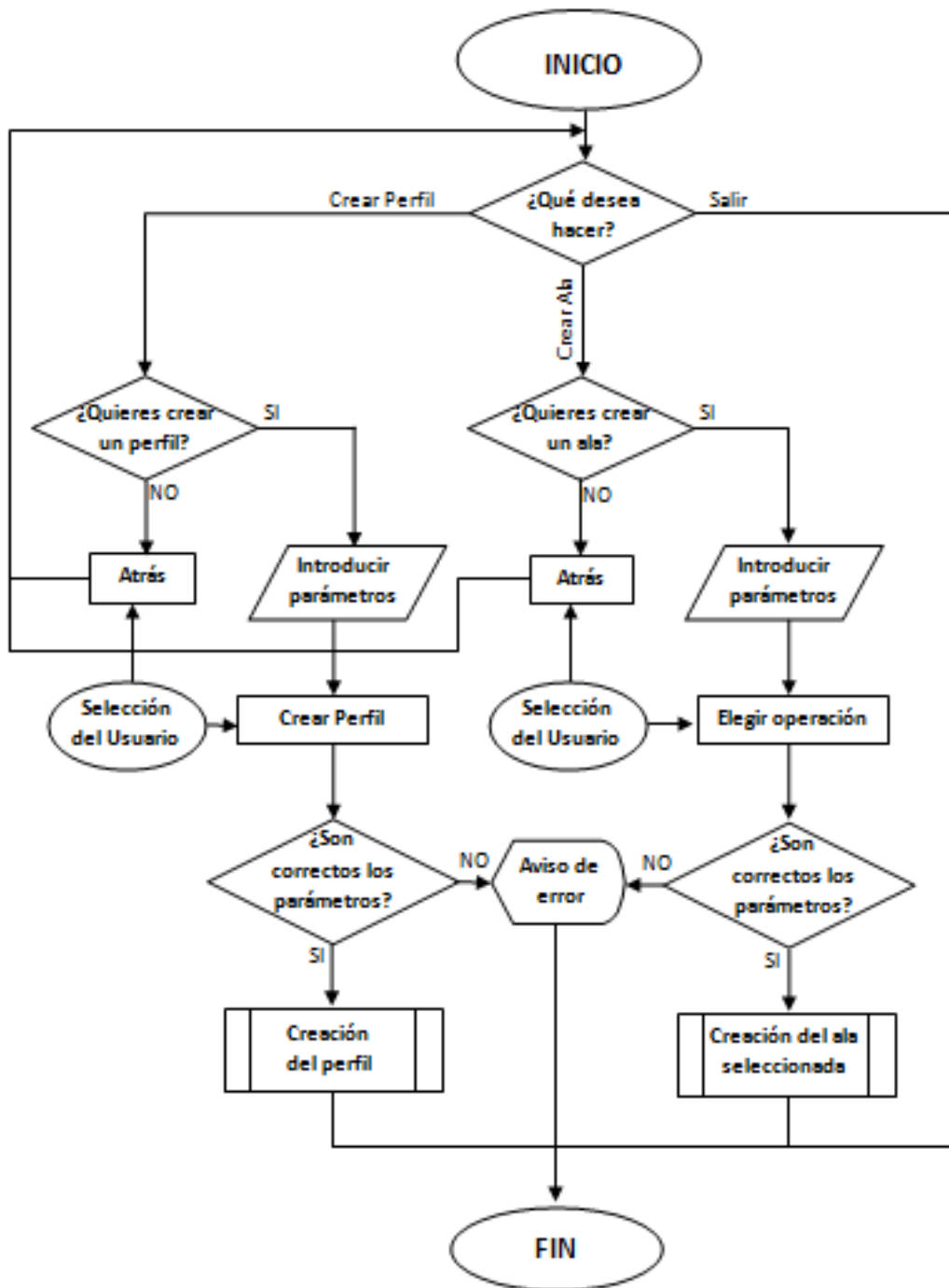


Figura 56. Diagrama de flujo de la aplicación

5 APLICACIÓN PERFILES Y ALAS CON NOMENCLATURA NACA DE 4 DÍGITOS: MANUAL DE USUARIO

Sabemos lo que somos pero no lo que podemos llegar a ser.

- William Shakespeare -

Una vez explicado el software desarrollado para realizar perfiles y alas con nomenclatura NACA de 4 dígitos. A continuación se procede a explicar los pasos que debe seguir el lector para el uso del mismo.

Antes de empezar a trabajar con la aplicación, el usuario debe asegurarse que tiene activado la visualización de parámetros y relaciones en CATIA. Si no se tiene activada, se debe ir al ANEXO 1 donde se desarrolla el procedimiento a seguir para su activación.

Para comenzar a usar la aplicación, el usuario debe arrancar CATIA si aún no lo ha hecho. Cuando CATIA se haya cargado, se debe ir a la pestaña *Tools* de la barra de herramientas y acceder a la opción *Macros* → *Visual Basic Editor*.

Una vez abierta la interfaz de VBA, el siguiente paso es arrancar la macro, para eso se debe hacer click sobre el formulario “VentanaPrincipal” y reproducirla dándole al botón *Play* de la barra de herramientas abriéndose la ventana de la Figura 57.

En la ventana principal se da opción a crear un perfil, crear un ala o salir del programa. Para ello hay tres botones: “Crear Perfil”, “Crear Ala” y “Salir” respectivamente. Si se hace click sobre uno de los dos primeros, se abrirá otra ventana correspondiente a dicha opción y si se pulsa “Salir” la aplicación se parará y cerrará.

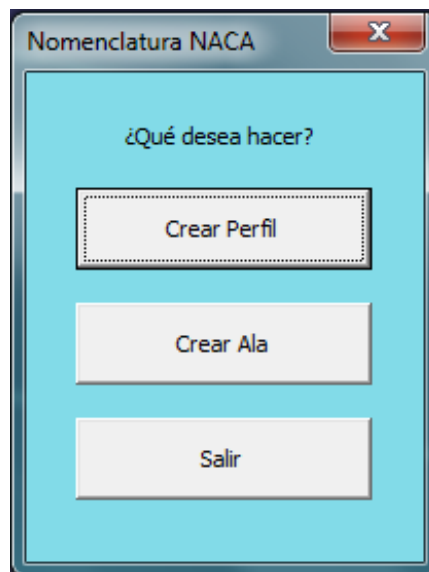


Figura 57. Interfaz principal de la aplicación

A continuación se va a hablar de las opciones “Crear Perfil” y “Crear Ala”. Si se hace click sobre alguna de estas, aparecerá la ventana de la Figura 58 en el caso de haber elegido la opción de crear un perfil o la ventana de la Figura 59 si se ha elegido la de crear una ala. Si por casualidad el usuario se ha equivocado de operación en la ventana principal, tan solo tiene que pulsar el botón “Atrás” que se incluye en ambos formularios para volver a la ventana principal del programa.

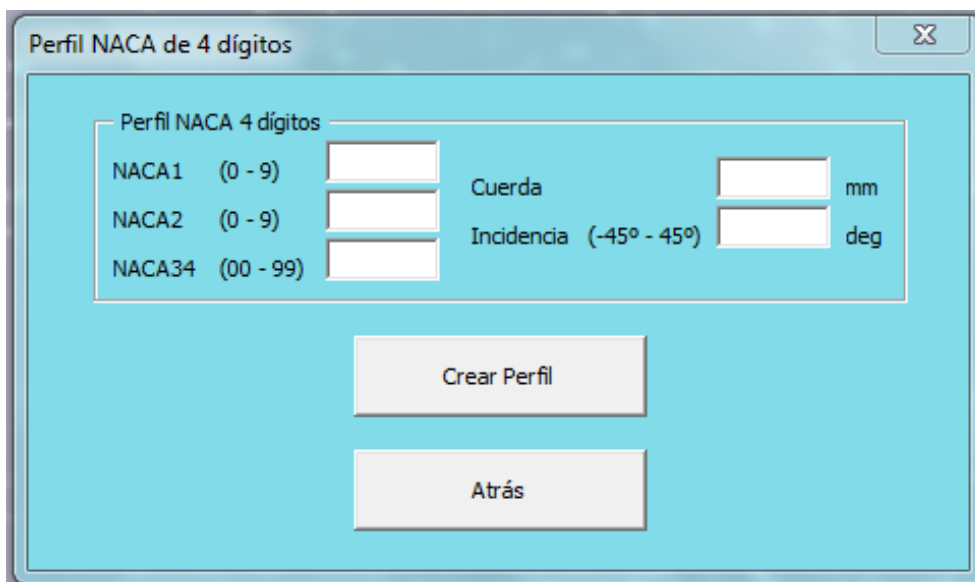


Figura 58. Interfaz PerfilesNACA4Digitos

Ala con perfil NACA de 4 dígitos

Perfil Encastre

NACA 1 (0 - 9)

NACA 2 (0 - 9)

NACA 34 (00 - 99)

Cuerda mm

Perfil Punta Ala

NACA 1 (0 - 9)

NACA 2 (0 - 9)

NACA 34 (00-99)

Estrechamiento %

Parámetros Ala

Incidencia (-45° - 45°) deg

Diedro (-40° - 45°) deg

Flecha (-40° - 45°) deg

Semievergadura mm

Torsión (-45° - 45°) deg

Ala en Superficie

Ala Sólida

Figura 59. Interfaz AlaNACA4Digitos

Ya dentro de la interfaz para crear un perfil o una ala se debe rellenar todos los parámetros que se piden. Para facilitar la entrada de dichos parámetros en la misma interfaz se puede ver el rango de aplicación que tienen los mismos y las unidades, si las tiene. El dato introducido en el parámetro debe estar comprendido en el rango de aplicación del mismo para que el programa pueda funcionar. A lo largo del apartado 1.6 de este trabajo se detalla y explica que significa cada parámetro y como afecta a la geometría del perfil o el ala que se pretende diseñar.

Una vez se tenga todos los valores de los parámetros introducidos, el siguiente paso es hacer click sobre la operación que se quiera ejecutar. Una vez pulsado el botón si nos hemos equivocado introduciendo algún parámetro y éste se encuentra fuera del rango permitido, el programa lanza un mensaje de advertencia y para el programa tal y como explicó en el apartado 4.1. En caso de que se hayan introducido correctamente todos los parámetros el programa al terminar de ejecutarse el programa, éste se cerrará y tendremos en CATIA un archivo .CATPART que contiene el perfil o ala creada.

5.1 Uso posterior a la aplicación

Durante todo el Trabajo Fin de Grado se habla de la parametrización, esto se debe a la gran importancia que tiene sobre este proyecto ya que uno de los fines para los que se ha diseñado esta aplicación es para que, una vez ejecutada la misma, se pueda realizar cambios sobre el perfil o ala diseñada con tan solo modificar el valor de los parámetros creados con la aplicación los cuales aparecerán en el árbol de trabajo una vez ejecutado el programa.

A partir del momento en el que se trabaje directamente sobre el entorno de CATIA, se debe tener cuidado con las modificaciones que se realiza sobre la geometría, ya que ahora no existe ningún ninguna imposición sobre estos y se excede demasiado el rango de aplicación empezarán a salir mensajes de error al actualizar CATIA. El botón de actualizar de CATIA es el que se muestra en la Figura 60.



Figura 60. Botón actualizar de CATIA

Respecto a los rangos de aplicación de la macro, cualquier persona con conocimientos básicos de la aeronáutica se daría cuenta de que estos son muy amplios, abarcando prácticamente todos los casos, por no decir todos, que se pueden encontrar en el mercado actualmente. Aunque si el usuario final requiriese aumentar dicho rango por cualquier motivo, existe una posibilidad. Lo que se debe hacer es introducir en primer lugar unos valores en los parámetros dentro de rango para ejecutar la aplicación y una vez tengamos el archivo donde se encuentra la operación realizada, modificar los parámetros, los cuales se encontrarán en el árbol de trabajo del entorno de CATIA, y actualizar la geometría. Esto no funciona para todos los parámetros ya que se ha comprobado que el ángulo de la flecha y del diedro nunca puede tener un ángulo menor de -40° .

El hecho de que se pueda aumentar los rangos de aplicación es debido a que para la creación de los puntos que se usan para definir el perfil se introducen unas cotas arbitrarias y estas cotas deben estar contenidas en las líneas de referencia y del primer cuadrante del *Sketch* en cuestión, tal y como se explicó en el apartado 1.6, ya que si quedan fuera o cambia de cuadrante, da error. Pero una vez ya este acotado y dibujado correctamente se permite aumentar el rango de aplicación en algunos de los parámetros. De todas formas no se aconseja sobrepasar los rangos para los que se ha establecido ya que no se sabe cuales serán los nuevos valores límites.

Conclusiones

CATIA es un programa informático de alto nivel que viene siendo utilizado en grandes y relevantes áreas de la ingeniería como son la aeronáutica y la automoción, ya que permite realizar tanto el diseño y el análisis de los modelos, como la fabricación final de los mismos.

Durante la realización del proyecto, en primer lugar, se decidió aprender a usar el módulo *Knowledgeware* de CATIA el cual permite el uso de parámetros, fórmulas y reglas ya que resulta vital saber usarlo para la realización de la aplicación.

El segundo paso fue generar perfiles y alas parametrizadas de forma manual para asimilar el procedimiento que se debía seguir, que no resultó ser trivial. Para ello se empezó parametrizando perfiles donde se cometió el primer gran error, no tener ningún cuidado al acotar los puntos con los que se dibujaría el perfil, ya que posteriormente a cada cota va asociada un conjunto de ecuaciones y si éstas no han sido acotadas siguiendo una lógica resulta ser un quebradero de cabeza ya que la regla que define el perfil requiere del número de cota y por cada perfil había alrededor de 130 *constraints* distintas. El segundo gran problema que se encontró fue que CATIA no entiende de distancias negativas en las cotas y había puntos con coordenadas negativas. Dicho problema se solucionó sumándole un *offset* a las coordenadas en altitud de forma que todas éstas resultasen positivas. Pero no bastaba cualquier *offset*, éste debía ser bastante grande si se quería poder poner un diedro negativo al ala.

Al final del proyecto se tuvieron problemas para parametrizar la flecha del ala, ya que de la forma en que se estaba realizando la parametrización no permitía establecer una flecha negativa, esto se solucionó sumando también un *offset* a las cotas en longitud. Descubriendo que los límites de operación de la aplicación se alcanza cuando algún punto cambia de cuadrante debido a que alguna de las coordenada del punto se hace negativa. Por este motivo se decidió sumarle un *offset* a las cotas en longitud y altura para alejar dichas cotas de los extremos del cuadrante y además se decidió que dicho *offset* debía tener el valor mas alto posible el cual normalmente sería la cuerda cuando solo se diseñan perfiles y la semienvergadura cuando se diseñan alas.

En lo referente a la programación en VBA de la aplicación, se pasó por diferentes etapas. Al inicio del proyecto no se tenía conocimiento alguno del lenguaje de programación ni del entorno. Dicho conocimiento se

adquirió durante la ejecución del proyecto para la posible realización de la aplicación y resultó no ser un proceso fácil ya que existe muy poca bibliografía sobre programación en CATIA. Por este motivo, la mayor parte de la programación del trabajo está basada en un libro aún en edición de la tutora del proyecto, pudiendo solventar de esta forma dicha carencia.

En el proceso de aprendizaje del lenguaje de programación se empezaron a desarrollar en primer lugar aplicaciones sencillas, incrementando el nivel de dificultad poco a poco hasta que se creyó que se tenían los conocimientos básicos necesarios para empezar a programar una aplicación propia.

Una vez se empezó a programar la aplicación del proyecto, se fue programando por pasos el procedimiento desarrollado para parametrizar perfiles y alas encontrando mas ausencia de información sobre operaciones avanzadas como era la utilización de *Rules*.

Es importante mencionar que llegados a cierto punto del proyecto se tuvieron dos líneas paralelas de trabajo que iban cogidas de la mano, la parametrización manual del perfil y de la ala y la programación de la aplicación debido a que un cambio en la parametrización incurria a un cambio en el código del programa y se tuvieron que ir implementado mejoras ya que se encontró algún que otro fallo en la parametrización que se estaba haciendo.

Finalmente, no solo se ha obtenido una macro para CATIA programada en VBA para generar perfiles y alas con nomenclatura NACA de 4 dígitos, sino que, además, dichos perfiles y alas se encuentran parametrizadas, por lo que una vez ejecutada la aplicación se puede implementar modificaciones en la geometría con tan solo cambiar el valor de los parámetros que definen la misma. De esta forma se consigue facilitar enormemente la etapa de diseño donde continuamente se están realizando cambios hasta llegar a la geometría final.

En contraposición, decir que la aplicación desarrollada tiene un defecto. El tiempo de ejecución de la macro es un poco elevado frente a otras, en torno a medio minuto. Durante su ejecución puede parecer que CATIA se queda colgado si no se posee un ordenador potente. Esto es debido no solo a la cantidad de líneas de código que tiene que ejecutarse para poder realizar la operación programada si no a la complejidad de las operaciones programadas. Aún así resulta factible su utilización frente al diseño manual, el cual puede llevar hasta días para su finalización si no se tiene conocimientos avanzados de CATIA.

Bibliografía

- [1] José Manuel Gordillo Arias de Saavedra, Guillaume Riboux Acher. *Introducción a la aerodinámica Potencial*.
- [2] Ciencia y poder Aéreo. Web Site. 27/08/2015
- [3] Geometry for Aerodynamicists. Web site. 27/08/2015
- [4] Temario Asignatura Cálculo de aeronaves del 4ºCurso de Ingeniería Aeroespacial. Tema 5.1 Introducción a los perfiles NACA. Universidad de Sevilla.
- [5] Damián Patón Terreno. *Diseño paramétrico de un ala con perfil NACA de 4 dígitos, flecha, diedro torsión y estrechamiento en CATIA V5 R19*. Web site. 17/04/2014
- [6] Eduardo Torrecillas Insagurbe. *El gran libro de CATIA*.
- [7] Dassault Systèmes. Web Site. 27/08/2015
- [8] *Fundamentos del KBE (Knowledge Based Engineering). Aplicación al diseño de engranajes de ejes paralelos con Catia v5*. Proyecto Fin de Carrera.
- [9] Cristina Torrecillas Lozano. *Introducción a la programación Visual Basic en CATIA V5 Y V6* (no publicado)
- [10] Emmett Ross. *VB Scripting for CATIA V5*. Segunda edición.
- [11] Dieter Ziethen. *CATIA V5 Macro Programming with Visual Basic Script*.

Glosario

RAE: Real Academia Española

RAE: Royal Aircraft Establishment

NACA: National Advisory Committee for Aeronautics

NASA: National Aeronautics and Space Administration

CATIA: Computer-Aided Three Dimensional Interactive Application

BASIC: Beginner's All-purpose Symbolic Instruction Code

VBA: Visual Basic for applications

ANEXOS

ANEXO 1. ACTIVACIÓN DE PARÁMETROS Y RELACIONES EN CATIA

Para poder trabajar con fórmulas y parámetros correctamente en CATIA, este se tiene que configurar para poder visualizarlos en el árbol de trabajo y de esta forma poder ir modificándolos. La operación de activación que se explica a continuación solo se debe realizar una vez.

Dentro de CATIA en la pestaña *Tools* y se debe acceder a la opción *Options*. En el menú de la izquierda despliega *Infraestructure* y selecciona *Product Structure*. Ahora ve a la pestaña *Tree Customization* y activa los parámetros y las relaciones. Esto se hace para poder mostrar parámetros tanto en un *Product* como en un *Part* ya que a veces se parametriza elementos que están fuera de un *Part* como por ejemplo una *Constraint* de un *Product*. En este proyecto no será necesario pero puede serlo a la hora de diseñar un avión completo. En la Figura 61 y la Figura 62 se puede ver el procedimiento a seguir.

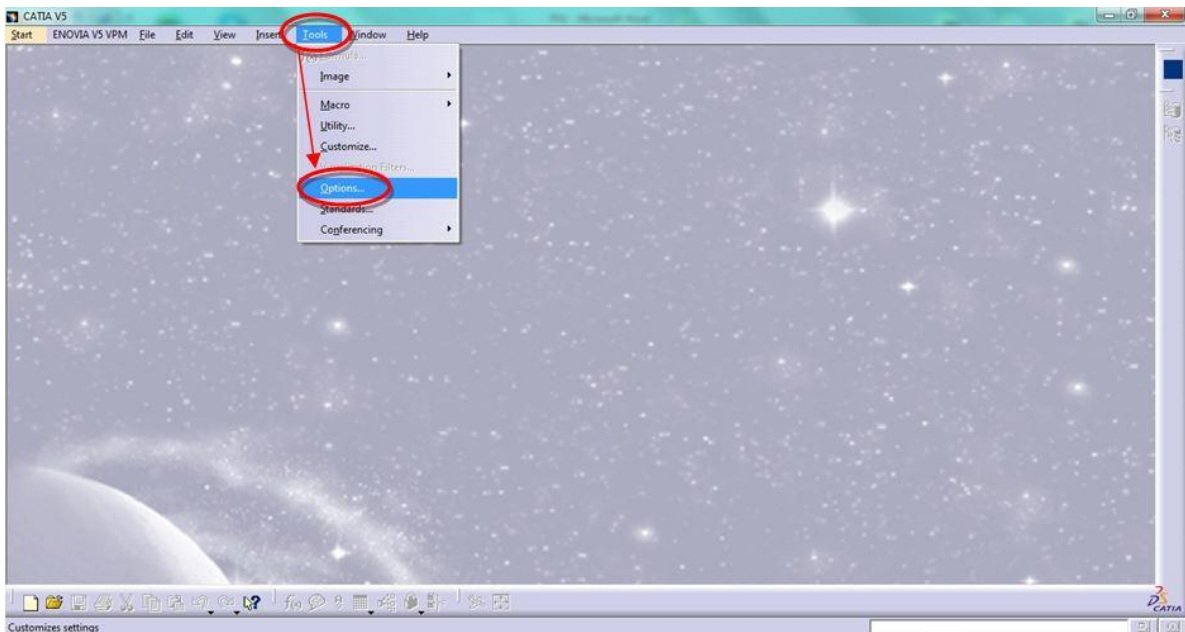


Figura 61. Localización *Options*

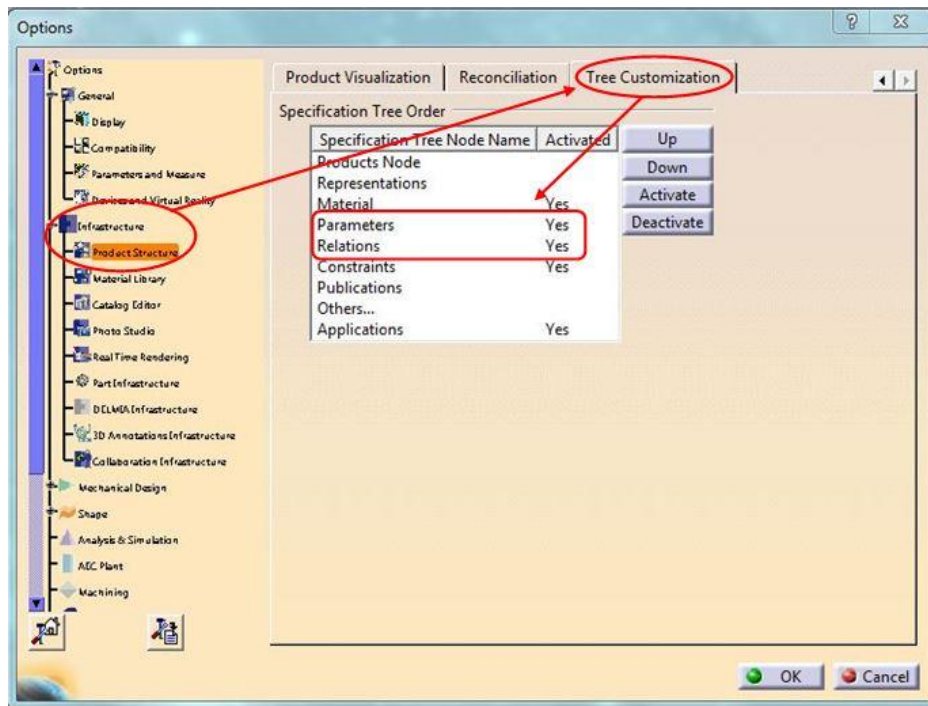


Figura 62. Configuración *Tree Customization*

A continuación selecciona *Part Infrastructure* en la misma ventana de *Options*. En la pestaña *General* se debe asegurar que esté marcada la opción *Show newly external references*. Esto sirve para poder usar parámetros de un *Part* en otro *Part* distinto, los parámetros de este tipo aparecerán en el árbol como *External Parameters*. En este proyecto tampoco es necesario usar esta funcionalidad, pero al igual que se mencionó antes, para diseños mas avanzados resulta de gran utilidad. En la Figura 63 se puede observar la configuración adecuada.

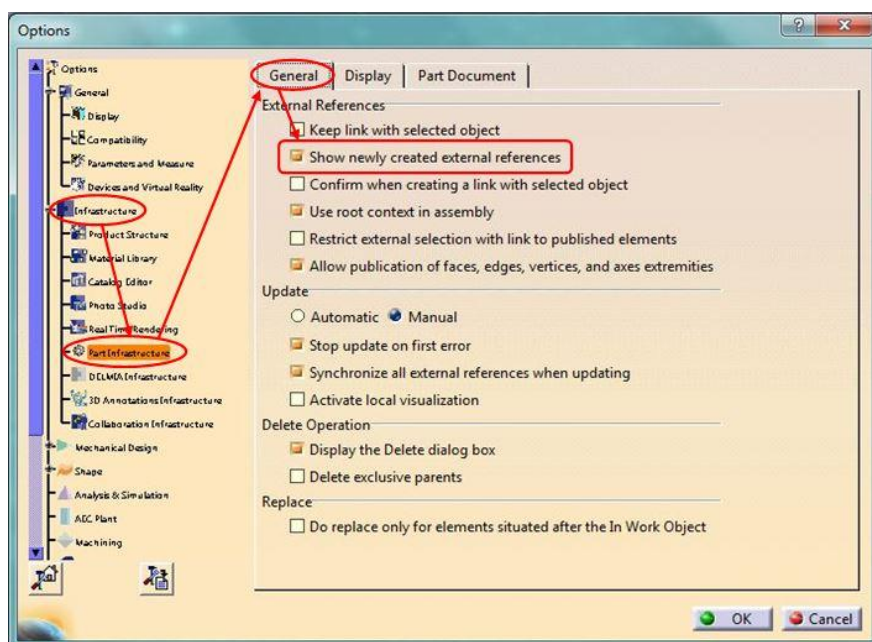


Figura 63. Configuración *Part Infrastructure, General*

Ahora dentro de la pestaña *Display* se tiene que asegurar que estén marcadas las opciones *External References*,

Parameters y *Relations*. Esto se hace para poder ver los parámetros en la pantalla y así poder modificarlos, pero si queremos ver el valor actual y la fórmula que los definen se tiene que desplegar el menú *General* en la ventana *Options*, seleccionar *Parameters and Measure* y en la pestaña *Knowledge* asegurar que estén marcadas las opciones *With value* y *With formula*. En la Figura 64 y Figura 65 se observa la configuración que debemos tener. Referencia [5]

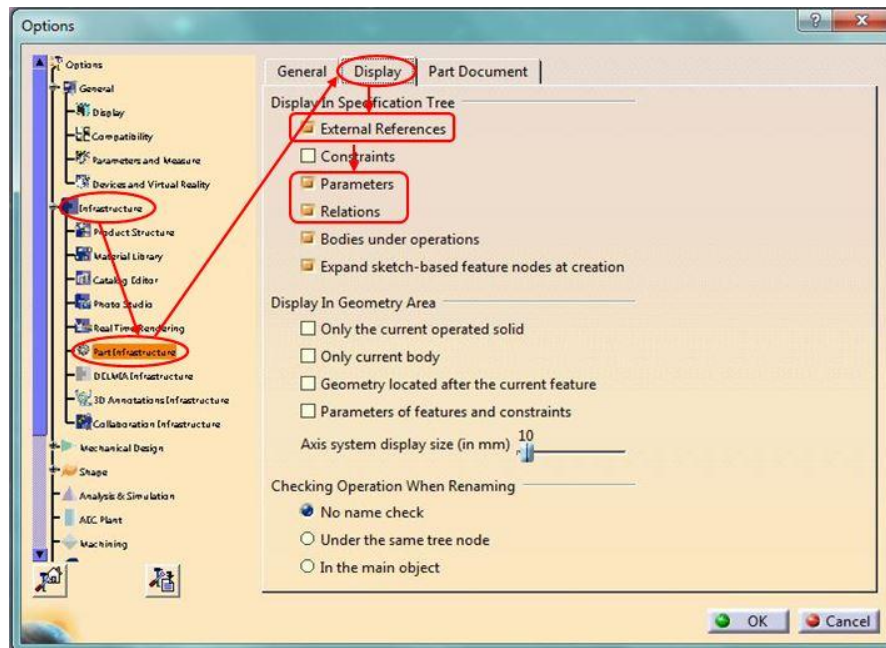


Figura 64. Configuración *Part Infrastructure*, *Display*

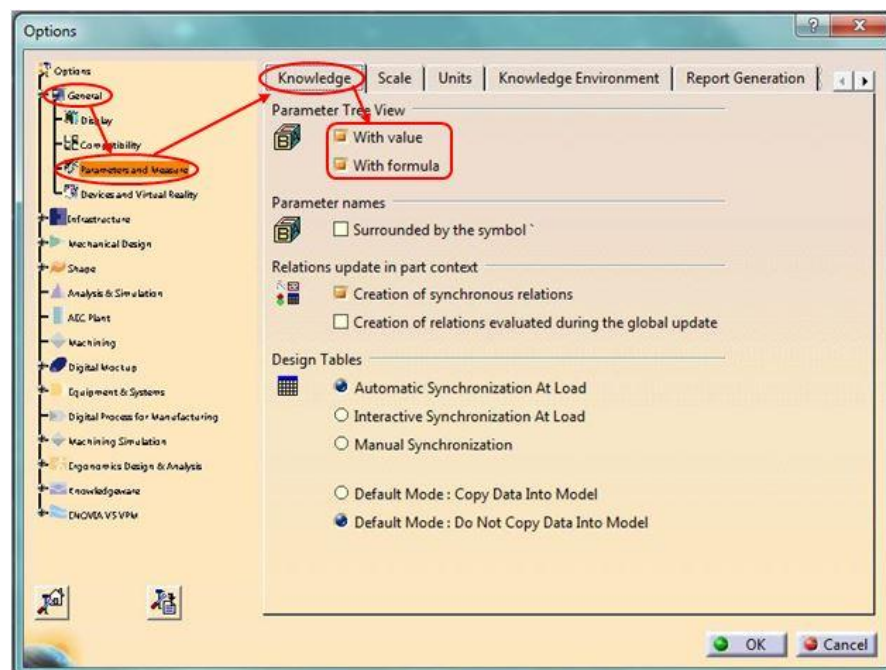


Figura 65. Configuración *Parameters and Measure*

Una vez realizado todos los pasos explicados ya tenemos CATIA correctamente configurado.

ANEXO 2. CÓDIGO PARA DIBUJAR UN PERFIL NACA DE 4 DÍGITOS

```

Private Sub CommandButton1_Click()
' Para que el programa funcione bien debemos estar trabajando
en un documento .CATPart y que no existan parámetros

' que se llamen iguales a los que creamos con nuestro programa
ya que entrarían en conflicto. Para evitar todos

' estos problemas vamos a hacer que este programa siempre se
ejecute en un nuevo documento .CATPart.

' De esta forma el programa pueda arrancar sin necesidad de
tener un part abierto, de forma que al ejecutarse se

' abra un nuevo part de forma automática.

' Al crearse un nuevo part cada vez que se ejecuta el programa,
no es necesario comprobar que estamos trabajando con

' archivo .CATPart ni que en dicho archivo existan ya
parámetros que se llamen igual a los que vamos a crear

' ya que al ser un nuevo archivo estará totalmente vacío.

' Normalmente se suele nombrar al objeto con el mismo
nombre de la orden seguida de un número para identificar

' rápidamente que estamos haciendo, esta es la lógica que
seguiremos en nuestro programa.

' Creamos y abrimos el nuevo Part

Dim documents1 As Documents

Dim partDocument1 As PartDocument

Dim part1 As Part

Set documents1 = CATIA.Documents

Set partDocument1 = documents1.Add("Part") ' Se carga
un documento activo part para trabajar

Set part1 = partDocument1.Part ' Se establece el
part1 como entorno para trabajar

' Activamos las relaciones y parámetros

Dim relations As relations

Dim parameters As parameters

Set relations = part1.relations

Set parameters = part1.parameters

' Asignamos los valores introducidos a cada parámetro

NACA1Valor = TextBox1

NACA2Valor = TextBox2

NACA34Valor = TextBox3

CuerdaValor = TextBox4

IncidenciaValor = TextBox5

' Se comprueba que los valores introducidos son correctos, si
estos no lo están se lanza un mensaje

' de advertencia y se para el programa para que los
modifiquemos y volvamos a ejecutar el programa.

' NACA1 debe ser un número entero positivo de una cifra
If NACA1Valor > -1 And NACA1Valor < 10 Then

Else

MsgBox "El valor introducido en NACA1 debe estar
comprendido entre 0 y 9."

End ' Este End se pone para parar el programa
si se encuentra el error

End If

' NACA2 debe ser un número entero positivo de una cifra
If NACA2Valor > -1 And NACA2Valor < 10 Then

Else

MsgBox "El valor introducido en NACA2 debe estar
comprendido entre 0 y 9."

End

End If

' NACA34 debe ser un número entero positivo este puede ir
desde el 0 al 99.
If NACA34Valor > -1 And NACA34Valor < 100 Then

Else

MsgBox "El valor introducido en NACA34 debe estar
comprendido entre 0 y 99."

End

End If

' Cuerda debe ser un número entero positivo mayor que 0
If CuerdaValor > 0 Then

Else

MsgBox "El valor introducido en Cuerda debe ser mayor que
0."

End

End If

If IncidenciaValor > -0.001 And IncidenciaValor < 46 Then

Else

MsgBox "El valor introducido en Incidencia debe estar
comprendido entre -45° y 45°."

End

End If

' Creamos los parámetros

Dim NACA1 As Parameter

Set NACA1 = parameters.CreateInteger("NACA1", 0)

Dim NACA2 As Parameter

Set NACA2 = parameters.CreateInteger("NACA2", 0)

Dim NACA34 As Parameter

Set NACA34 = parameters.CreateInteger("NACA34", 0)

Dim Cuerda As Parameter

Set Cuerda = parameters.CreateDimension("Cuerda",

```

```

"LENGTH", 0)
Dim Incidencia As Parameter
Set Incidencia = parameters.CreateDimension("Incidencia",
"ANGLE", 0)
' Asignamos el valor correspondiente a cada parámetro
NACA1.Value = NACA1Valor
NACA2.Value = NACA2Valor
NACA34.Value = NACA34Valor
Cuerda.Value = CuerdaValor
Incidencia.Value = IncidenciaValor
' Ahora debemos crear 60 puntos, 30 para extradós y 30 para
intradós
' Definimos el entorno de trabajo
Dim bodies1 As Bodies
Dim body1 As Body
Dim sketches1 As Sketches
Dim originElements1 As OriginElements
Dim reference1 As Reference
Dim sketch1 As Sketch
' Establecimiento del entorno de trabajo

Set bodies1 = part1.Bodies
Set body1 = bodies1.Item("PartBody") ' Se activa el body1
como body de trabajo
Set sketches1 = body1.Sketches ' Se establece el sketch
Set originElements1 = part1.OriginElements ' Sistema de
referencia 3D
Set reference1 = originElements1.PlaneYZ ' Se establece el
plano YZ como plano de trabajo
Set sketch1 = sketches1.Add(reference1) ' Toma la
referencia1 como referencia para trabajar
' Trabajamos en el plano yz
Dim arrayOfVariantOfDouble1(8)
arrayOfVariantOfDouble1(0) = 0# ' Establece el vector
unidad x en el plano X
arrayOfVariantOfDouble1(1) = 0# ' Establece el vector
unidad y en el plano X
arrayOfVariantOfDouble1(2) = 0# ' Establece el vector
unidad z en el plano X
arrayOfVariantOfDouble1(3) = 0# ' Establece el vector
unidad x en el plano Y
arrayOfVariantOfDouble1(4) = 1# ' Establece el vector
unidad y en el plano Y
arrayOfVariantOfDouble1(5) = 0# ' Establece el vector
unidad z en el plano Y
arrayOfVariantOfDouble1(6) = 0# ' Establece el vector
unidad x en el plano Z
arrayOfVariantOfDouble1(7) = 0# ' Establece el vector
unidad y en el plano Z
arrayOfVariantOfDouble1(8) = 1# ' Establece el vector
unidad z en el plano Z
Set sketch1Variant = sketch1
sketch1Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble1 ' Sistema absoluto 3D para el sketch
part1.InWorkObject = sketch1
' Establecimiento del conjunto de herramientas 2D y asignarlo
al sketch de trabajo
Dim factory2D1 As Factory2D
Set factory2D1 = sketch1.OpenEdition()
' Se establecen los elementos geométricos
Dim geometricElements1 As GeometricElements
Set geometricElements1 = sketch1.GeometricElements
' Se define el sistema de ejes dentro del sketch
Dim axis2D1 As Axis2D
Set axis2D1 = geometricElements1.Item("AbsoluteAxis")
' Establecimiento de las direcciones horizontal y vertical
Dim line2D1 As Line2D
Set line2D1 = axis2D1.GetItem("HDirection")
line2D1.ReportName = 1
Dim line2D2 As Line2D
Set line2D2 = axis2D1.GetItem("VDirection")
line2D2.ReportName = 2
' Definimos los puntos con unos valores distribuidos
uniformemente. Estos valores
' no son los definitivos ya que estos serán modificados con la
rule.
Dim point2D1 As ControlPoint2D
Set point2D1 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D1.ReportName = 2 + 1
Dim point2D2 As ControlPoint2D
Set point2D2 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D2.ReportName = 2 + 2
Dim point2D3 As ControlPoint2D
Set point2D3 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D3.ReportName = 2 + 3
Dim point2D4 As ControlPoint2D
Set point2D4 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D4.ReportName = 2 + 4
Dim point2D5 As ControlPoint2D
Set point2D5 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)

```

point2D5.ReportName = 2 + 5
 Dim point2D6 As ControlPoint2D
 Set point2D6 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D6.ReportName = 2 + 6
 Dim point2D7 As ControlPoint2D
 Set point2D7 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D7.ReportName = 2 + 7
 Dim point2D8 As ControlPoint2D
 Set point2D8 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D8.ReportName = 2 + 8
 Dim point2D9 As ControlPoint2D
 Set point2D9 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D9.ReportName = 2 + 9
 Dim point2D10 As ControlPoint2D
 Set point2D10 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D10.ReportName = 2 + 10
 Dim point2D11 As ControlPoint2D
 Set point2D11 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D11.ReportName = 2 + 11
 Dim point2D12 As ControlPoint2D
 Set point2D12 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D12.ReportName = 2 + 12
 Dim point2D13 As ControlPoint2D
 Set point2D13 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D13.ReportName = 2 + 13
 Dim point2D14 As ControlPoint2D
 Set point2D14 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D14.ReportName = 2 + 14
 Dim point2D15 As ControlPoint2D
 Set point2D15 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D15.ReportName = 2 + 15
 Dim point2D16 As ControlPoint2D
 Set point2D16 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D16.ReportName = 2 + 16
 Dim point2D17 As ControlPoint2D
 Set point2D17 = factory2D1.CreateControlPoint(CuerdaValor,

CuerdaValor)
 point2D17.ReportName = 2 + 17
 Dim point2D18 As ControlPoint2D
 Set point2D18 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D18.ReportName = 2 + 18
 Dim point2D19 As ControlPoint2D
 Set point2D19 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D19.ReportName = 2 + 19
 Dim point2D20 As ControlPoint2D
 Set point2D20 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D20.ReportName = 2 + 20
 Dim point2D21 As ControlPoint2D
 Set point2D21 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D21.ReportName = 2 + 21
 Dim point2D22 As ControlPoint2D
 Set point2D22 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D22.ReportName = 2 + 22
 Dim point2D23 As ControlPoint2D
 Set point2D23 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D23.ReportName = 2 + 23
 Dim point2D24 As ControlPoint2D
 Set point2D24 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D24.ReportName = 2 + 24
 Dim point2D25 As ControlPoint2D
 Set point2D25 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D25.ReportName = 2 + 25
 Dim point2D26 As ControlPoint2D
 Set point2D26 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D26.ReportName = 2 + 26
 Dim point2D27 As ControlPoint2D
 Set point2D27 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D27.ReportName = 2 + 27
 Dim point2D28 As ControlPoint2D
 Set point2D28 = factory2D1.CreateControlPoint(CuerdaValor, CuerdaValor)
 point2D28.ReportName = 2 + 28


```

Dim point2D29 As ControlPoint2D
Set point2D29 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D29.ReportName = 2 + 29
Dim point2D30 As ControlPoint2D
Set point2D30 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D30.ReportName = 2 + 30
Dim point2D31 As ControlPoint2D
Set point2D31 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D31.ReportName = 2 + 31
Dim point2D32 As ControlPoint2D
Set point2D32 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D32.ReportName = 2 + 32
Dim point2D33 As ControlPoint2D
Set point2D33 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D33.ReportName = 2 + 33
Dim point2D34 As ControlPoint2D
Set point2D34 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D34.ReportName = 2 + 34
Dim point2D35 As ControlPoint2D
Set point2D35 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D35.ReportName = 2 + 35
Dim point2D36 As ControlPoint2D
Set point2D36 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D36.ReportName = 2 + 36
Dim point2D37 As ControlPoint2D
Set point2D37 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D37.ReportName = 2 + 37
Dim point2D38 As ControlPoint2D
Set point2D38 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D38.ReportName = 2 + 38
Dim point2D39 As ControlPoint2D
Set point2D39 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D39.ReportName = 2 + 39
Dim point2D40 As ControlPoint2D
Set point2D40 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D40.ReportName = 2 + 40
Dim point2D41 As ControlPoint2D
Set point2D41 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D41.ReportName = 2 + 41
Dim point2D42 As ControlPoint2D
Set point2D42 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D42.ReportName = 2 + 42
Dim point2D43 As ControlPoint2D
Set point2D43 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D43.ReportName = 2 + 43
Dim point2D44 As ControlPoint2D
Set point2D44 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D44.ReportName = 2 + 44
Dim point2D45 As ControlPoint2D
Set point2D45 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D45.ReportName = 2 + 45
Dim point2D46 As ControlPoint2D
Set point2D46 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D46.ReportName = 2 + 46
Dim point2D47 As ControlPoint2D
Set point2D47 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D47.ReportName = 2 + 47
Dim point2D48 As ControlPoint2D
Set point2D48 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D48.ReportName = 2 + 48
Dim point2D49 As ControlPoint2D
Set point2D49 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D49.ReportName = 2 + 49
Dim point2D50 As ControlPoint2D
Set point2D50 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D50.ReportName = 2 + 50
Dim point2D51 As ControlPoint2D
Set point2D51 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D51.ReportName = 2 + 51
Dim point2D52 As ControlPoint2D
Set point2D52 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D52.ReportName = 2 + 52

```

```

Dim point2D53 As ControlPoint2D
Set point2D53 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D53.ReportName = 2 + 53

Dim point2D54 As ControlPoint2D
Set point2D54 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D54.ReportName = 2 + 54

Dim point2D55 As ControlPoint2D
Set point2D55 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D55.ReportName = 2 + 55

Dim point2D56 As ControlPoint2D
Set point2D56 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D56.ReportName = 2 + 56

Dim point2D57 As ControlPoint2D
Set point2D57 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D57.ReportName = 2 + 57

Dim point2D58 As ControlPoint2D
Set point2D58 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D58.ReportName = 2 + 58

Dim point2D59 As ControlPoint2D
Set point2D59 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D59.ReportName = 2 + 59

Dim point2D60 As ControlPoint2D
Set point2D60 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D60.ReportName = 2 + 60
.....
.....

' Creamos el punto del borde de ataque tanto como control
point
' como punto (pero de construcción) ya que este ultimo
' lo necesitamos para parametrizar las lineas de referencia para
poner la incidencia a nuestro perfil.
.....
.....

Dim point2D61 As ControlPoint2D
Set point2D61 = factory2D1.CreateControlPoint(CuerdaValor,
CuerdaValor)
point2D61.ReportName = 63

Dim point2D62 As Point2D
Set point2D62 = factory2D1.CreatePoint(CuerdaValor,
CuerdaValor)
point2D62.ReportName = 64
point2D62.Construction = True

' Definimos las constraints de todos los puntos acotando estos
en distancias a ambas direcciones del sketch.

' Debemos seguir una lógica para acotar los puntos ya que esta
afectará a nuestra Rule.
.....
.....
'''          INCIDENCIA          '''
.....
.....

' Los perfiles suelen tener una incidencia, ángulo, respecto a la
horizontal, por lo que crearemos unas nuevas

' lineas de referencia que formen un ángulo igual a la
incedencia y posteriormente acotaremos sobre estas lineas.

' Creamos las lineas que utilizaremos como referencia.
Comenzamos por la "vertical".

' En primer lugar acotamos y parametrizamos el punto del
borde de ataque.

Dim constraints1 As constraints
Set constraints1 = sketch1.constraints

Dim reference2 As Reference
Set reference2 = part1.CreateReferenceFromObject(line2D2)
'Referencia al eje vertical

Dim reference3 As Reference
Set reference3 = part1.CreateReferenceFromObject(line2D1)
'Referencia al eje horizontal

Dim reference4 As Reference
Set reference4 = part1.CreateReferenceFromObject(point2D62)
'Punto de
construcción

Dim constraint1 As Constraint
Set constraint1 = constraints1.AddBiEltCst(catCstTypeDistance,
reference2,
reference4)

constraint1.Mode = catCstModeDrivingDimension

Dim length1 As Length
Set length1 = constraint1.Dimension

Dim formula1 As Formula
'Set formula1 = relations.CreateFormula("Punto borde de
ataque", "Borde de ataque", length1,
"Cuerda*tan(Incidencia)")

Set formula1 = relations.CreateFormula("Punto borde de
ataque", "Borde de ataque", length1, "Cuerda")

Dim constraint2 As Constraint
Set constraint2 = constraints1.AddBiEltCst(catCstTypeDistance,
reference3,
reference4)

```

```

constraint2.Mode = catCstModeDrivingDimension
Dim length2 As Length
Set length2 = constraint2.Dimension
Dim formula2 As Formula
Set formula2 = relations.CreateFormula("Punto borde de
ataque", "Borde de ataque", length2, "Cuerda")
' Se crean los puntos de interes para crear los ejes de
coordenadas rotados.
Dim point2D63 As Point2D
Set point2D63 = factory2D1.CreatePoint(-7000#, -7000#)
point2D63.ReportName = 65
Dim point2D64 As Point2D
Set point2D64 = factory2D1.CreatePoint(7000#, 7000#)
point2D64.ReportName = 66
' Se crea la linea
Dim line2D3 As Line2D
Set line2D3 = factory2D1.CreateLine(-7000#, -7000#, 7000#,
7000#)
line2D3.ReportName = 67
line2D3.Construction = True
line2D3.StartPoint = point2D63
line2D3.EndPoint = point2D64
' Acotamos la linea. En primer lugar que forme un ángulo con
la dirección vertical y en segundo
' lugar que sea coincidente con el punto del borde de ataque.
Dim reference5 As Reference
Set reference5 = part1.CreateReferenceFromObject(line2D3)
Dim constraint3 As Constraint
Set constraint3 = constraints1.AddBiEltCst(catCstTypeAngle,
reference5, reference2)
constraint3.Mode = catCstModeDrivingDimension
constraint3.AngleSector = catCstAngleSector0
Dim angle3 As Angle
Set angle3 = constraint3.Dimension
Dim formula3 As Formula
Set formula3 = relations.CreateFormula("Linea referencia
vertical Incidencia", "Incidencia", angle3, "Incidencia")
Dim reference6 As Reference
Set reference6 = part1.CreateReferenceFromObject(line2D3)
'vertical cambiada
Dim reference7 As Reference
Set reference7 = part1.CreateReferenceFromObject(point2D62)
'Punto
construcción borde ataque
Dim constraint4 As Constraint
Set constraint4 = constraints1.AddBiEltCst(catCstTypeOn,
reference6, reference7) 'Constraint de coincidencia

```

```

constraint4.Mode = catCstModeDrivingDimension
' Linea "horizontal de referencia".
Dim point2D65 As Point2D
Set point2D65 = factory2D1.CreatePoint(-7000, 7000)
point2D65.ReportName = 68
Dim point2D66 As Point2D
Set point2D66 = factory2D1.CreatePoint(7000, -7000)
point2D66.ReportName = 69
Dim line2D4 As Line2D
Set line2D4 = factory2D1.CreateLine(-7000, 7000, 7000, -
7000)
line2D4.ReportName = 70
line2D4.Construction = True
line2D4.StartPoint = point2D65
line2D4.EndPoint = point2D66
Dim reference8 As Reference
Set reference8 = part1.CreateReferenceFromObject(line2D4)
Dim constraint5 As Constraint
Set constraint5 = constraints1.AddBiEltCst(catCstTypePerpendicularity,
reference8, reference6)
constraint5.Mode = catCstModeDrivingDimension
Dim constraint6 As Constraint
Set constraint6 = constraints1.AddBiEltCst(catCstTypeDistance,
reference7, reference8)
constraint6.Mode = catCstModeDrivingDimension
Dim length6 As Length
Set length6 = constraint6.Dimension
Dim formula4 As Formula
Set formula4 = relations.CreateFormula("Linea referencia
horizontal Incidencia", "Referencia horizontal Incidencia",
length6, "Cuerda")
' Definimos las constraints de todos los puntos acotando estos
en distancias a ambas direcciones del sketch.
' Debemos seguir una lógica para acotar los puntos ya que esta
afectará a nuestra Rule.
' En primer lugar vamos a acotar los puntos del Extradós por
su distancia al eje vertical
Dim reference9 As Reference
Set reference9 = part1.CreateReferenceFromObject(point2D1)
Dim constraint7 As Constraint
Set constraint7 =

```

```

constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference9)
constraint7.Mode = catCstModeDrivingDimension
Dim reference10 As Reference
Set reference10 =
part1.CreateReferenceFromObject(point2D2)
Dim constraint8 As Constraint
Set constraint8 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference10)
constraint8.Mode = catCstModeDrivingDimension
Dim reference11 As Reference
Set reference11 =
part1.CreateReferenceFromObject(point2D3)
Dim constraint9 As Constraint
Set constraint9 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference11)
constraint9.Mode = catCstModeDrivingDimension
Dim reference12 As Reference
Set reference12 =
part1.CreateReferenceFromObject(point2D4)
Dim constraint10 As Constraint
Set constraint10 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference12)
constraint10.Mode = catCstModeDrivingDimension
Dim reference13 As Reference
Set reference13 =
part1.CreateReferenceFromObject(point2D5)
Dim constraint11 As Constraint
Set constraint11 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference13)
constraint11.Mode = catCstModeDrivingDimension
Dim reference14 As Reference
Set reference14 =
part1.CreateReferenceFromObject(point2D6)
Dim constraint12 As Constraint
Set constraint12 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference14)
constraint12.Mode = catCstModeDrivingDimension
Dim reference15 As Reference
Set reference15 =
part1.CreateReferenceFromObject(point2D7)
Dim constraint13 As Constraint
Set constraint13 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference15)
constraint13.Mode = catCstModeDrivingDimension
Dim reference16 As Reference
Set reference16 =
part1.CreateReferenceFromObject(point2D8)
Dim constraint14 As Constraint
Set constraint14 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference16)
constraint14.Mode = catCstModeDrivingDimension
Dim reference17 As Reference
Set reference17 =
part1.CreateReferenceFromObject(point2D9)
Dim constraint15 As Constraint
Set constraint15 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference17)
constraint15.Mode = catCstModeDrivingDimension
Dim reference18 As Reference
Set reference18 =
part1.CreateReferenceFromObject(point2D10)
Dim constraint16 As Constraint
Set constraint16 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference18)
constraint16.Mode = catCstModeDrivingDimension
Dim reference19 As Reference
Set reference19 =
part1.CreateReferenceFromObject(point2D11)
Dim constraint17 As Constraint
Set constraint17 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference19)
constraint17.Mode = catCstModeDrivingDimension
Dim reference20 As Reference
Set reference20 =
part1.CreateReferenceFromObject(point2D12)
Dim constraint18 As Constraint
Set constraint18 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference20)
constraint18.Mode = catCstModeDrivingDimension
Dim reference21 As Reference
Set reference21 =
part1.CreateReferenceFromObject(point2D13)
Dim constraint19 As Constraint
Set constraint19 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference21)
constraint19.Mode = catCstModeDrivingDimension
Dim reference22 As Reference

```

```

Set reference22 = Dim constraint26 As Constraint
part1.CreateReferenceFromObject(point2D14)

Dim constraint20 As Constraint

Set constraint20 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference22)
constraint20.Mode = catCstModeDrivingDimension

Dim reference23 As Reference

Set reference23 = part1.CreateReferenceFromObject(point2D15)

Dim constraint21 As Constraint

Set constraint21 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference23)
constraint21.Mode = catCstModeDrivingDimension

Dim reference24 As Reference

Set reference24 = part1.CreateReferenceFromObject(point2D16)

Dim constraint22 As Constraint

Set constraint22 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference24)
constraint22.Mode = catCstModeDrivingDimension

Dim reference25 As Reference

Set reference25 = part1.CreateReferenceFromObject(point2D17)

Dim constraint23 As Constraint

Set constraint23 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference25)
constraint23.Mode = catCstModeDrivingDimension

Dim reference26 As Reference

Set reference26 = part1.CreateReferenceFromObject(point2D18)

Dim constraint24 As Constraint

Set constraint24 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference26)
constraint24.Mode = catCstModeDrivingDimension

Dim reference27 As Reference

Set reference27 = part1.CreateReferenceFromObject(point2D19)

Dim constraint25 As Constraint

Set constraint25 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference27)
constraint25.Mode = catCstModeDrivingDimension

Dim reference28 As Reference

Set reference28 = part1.CreateReferenceFromObject(point2D20)

Set constraint26 = Dim constraint26 As Constraint
constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference28)
constraint26.Mode = catCstModeDrivingDimension

Dim reference29 As Reference

Set reference29 = part1.CreateReferenceFromObject(point2D21)

Dim constraint27 As Constraint

Set constraint27 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference29)
constraint27.Mode = catCstModeDrivingDimension

Dim reference30 As Reference

Set reference30 = part1.CreateReferenceFromObject(point2D22)

Dim constraint28 As Constraint

Set constraint28 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference30)
constraint28.Mode = catCstModeDrivingDimension

Dim reference31 As Reference

Set reference31 = part1.CreateReferenceFromObject(point2D23)

Dim constraint29 As Constraint

Set constraint29 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference31)
constraint29.Mode = catCstModeDrivingDimension

Dim reference32 As Reference

Set reference32 = part1.CreateReferenceFromObject(point2D24)

Dim constraint30 As Constraint

Set constraint30 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference32)
constraint30.Mode = catCstModeDrivingDimension

Dim reference33 As Reference

Set reference33 = part1.CreateReferenceFromObject(point2D25)

Dim constraint31 As Constraint

Set constraint31 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference33)
constraint31.Mode = catCstModeDrivingDimension

Dim reference34 As Reference

Set reference34 = part1.CreateReferenceFromObject(point2D26)

Dim constraint32 As Constraint

Set constraint32 =

```

```

constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference34)
constraint32.Mode = catCstModeDrivingDimension
Dim reference35 As Reference
Set reference35 =
part1.CreateReferenceFromObject(point2D27)
Dim constraint33 As Constraint
Set constraint33 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference35)
constraint33.Mode = catCstModeDrivingDimension
Dim reference36 As Reference
Set reference36 =
part1.CreateReferenceFromObject(point2D28)
Dim constraint34 As Constraint
Set constraint34 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference36)
constraint34.Mode = catCstModeDrivingDimension
Dim reference37 As Reference
Set reference37 =
part1.CreateReferenceFromObject(point2D29)
Dim constraint35 As Constraint
Set constraint35 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference37)
constraint35.Mode = catCstModeDrivingDimension
Dim reference38 As Reference
Set reference38 =
part1.CreateReferenceFromObject(point2D30)
Dim constraint36 As Constraint
Set constraint36 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference38)
constraint36.Mode = catCstModeDrivingDimension
.....
'En segundo lugar vamos a acotar los puntos del Extradós por
su distancia al eje horizontal
.....
Dim reference39 As Reference
Set reference39 =
part1.CreateReferenceFromObject(point2D1)
Dim constraint37 As Constraint
Set constraint37 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference39)
constraint37.Mode = catCstModeDrivingDimension
Dim reference40 As Reference
Set reference40 =
part1.CreateReferenceFromObject(point2D2)

```

```

Dim constraint38 As Constraint
Set constraint38 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference40)
constraint38.Mode = catCstModeDrivingDimension
Dim reference41 As Reference
Set reference41 =
part1.CreateReferenceFromObject(point2D3)
Dim constraint39 As Constraint
Set constraint39 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference41)
constraint39.Mode = catCstModeDrivingDimension
Dim reference42 As Reference
Set reference42 =
part1.CreateReferenceFromObject(point2D4)
Dim constraint40 As Constraint
Set constraint40 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference42)
constraint40.Mode = catCstModeDrivingDimension
Dim reference43 As Reference
Set reference43 =
part1.CreateReferenceFromObject(point2D5)
Dim constraint41 As Constraint
Set constraint41 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference43)
constraint41.Mode = catCstModeDrivingDimension
Dim reference44 As Reference
Set reference44 =
part1.CreateReferenceFromObject(point2D6)
Dim constraint42 As Constraint
Set constraint42 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference44)
constraint42.Mode = catCstModeDrivingDimension
Dim reference45 As Reference
Set reference45 =
part1.CreateReferenceFromObject(point2D7)
Dim constraint43 As Constraint
Set constraint43 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference45)
constraint43.Mode = catCstModeDrivingDimension
Dim reference46 As Reference
Set reference46 =
part1.CreateReferenceFromObject(point2D8)
Dim constraint44 As Constraint
Set constraint44 =

```

```

constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference46)
constraint44.Mode = catCstModeDrivingDimension
Dim reference47 As Reference
Set reference47 = part1.CreateReferenceFromObject(point2D9)
Dim constraint45 As Constraint
Set constraint45 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference47)
constraint45.Mode = catCstModeDrivingDimension
Dim reference48 As Reference
Set reference48 = part1.CreateReferenceFromObject(point2D10)
Dim constraint46 As Constraint
Set constraint46 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference48)
constraint46.Mode = catCstModeDrivingDimension
Dim reference49 As Reference
Set reference49 = part1.CreateReferenceFromObject(point2D11)
Dim constraint47 As Constraint
Set constraint47 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference49)
constraint47.Mode = catCstModeDrivingDimension
Dim reference50 As Reference
Set reference50 = part1.CreateReferenceFromObject(point2D12)
Dim constraint48 As Constraint
Set constraint48 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference50)
constraint48.Mode = catCstModeDrivingDimension
Dim reference51 As Reference
Set reference51 = part1.CreateReferenceFromObject(point2D13)
Dim constraint49 As Constraint
Set constraint49 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference51)
constraint49.Mode = catCstModeDrivingDimension
Dim reference52 As Reference
Set reference52 = part1.CreateReferenceFromObject(point2D14)
Dim constraint50 As Constraint
Set constraint50 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference52)
constraint50.Mode = catCstModeDrivingDimension
Dim reference53 As Reference
Set reference53 = part1.CreateReferenceFromObject(point2D15)
Dim constraint51 As Constraint
Set constraint51 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference53)
constraint51.Mode = catCstModeDrivingDimension
Dim reference54 As Reference
Set reference54 = part1.CreateReferenceFromObject(point2D16)
Dim constraint52 As Constraint
Set constraint52 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference54)
constraint52.Mode = catCstModeDrivingDimension
Dim reference55 As Reference
Set reference55 = part1.CreateReferenceFromObject(point2D17)
Dim constraint53 As Constraint
Set constraint53 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference55)
constraint53.Mode = catCstModeDrivingDimension
Dim reference56 As Reference
Set reference56 = part1.CreateReferenceFromObject(point2D18)
Dim constraint54 As Constraint
Set constraint54 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference56)
constraint54.Mode = catCstModeDrivingDimension
Dim reference57 As Reference
Set reference57 = part1.CreateReferenceFromObject(point2D19)
Dim constraint55 As Constraint
Set constraint55 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference57)
constraint55.Mode = catCstModeDrivingDimension
Dim reference58 As Reference
Set reference58 = part1.CreateReferenceFromObject(point2D20)
Dim constraint56 As Constraint
Set constraint56 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference58)
constraint56.Mode = catCstModeDrivingDimension
Dim reference59 As Reference

```

```

Set          reference59          =
part1.CreateReferenceFromObject(point2D21)

Dim constraint57 As Constraint

Set          constraint57          =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference59)

constraint57.Mode = catCstModeDrivingDimension

Dim reference60 As Reference

Set          reference60          =
part1.CreateReferenceFromObject(point2D22)

Dim constraint58 As Constraint

Set          constraint58          =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference60)

constraint58.Mode = catCstModeDrivingDimension

Dim reference61 As Reference

Set          reference61          =
part1.CreateReferenceFromObject(point2D23)

Dim constraint59 As Constraint

Set          constraint59          =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference61)

constraint59.Mode = catCstModeDrivingDimension

Dim reference62 As Reference

Set          reference62          =
part1.CreateReferenceFromObject(point2D24)

Dim constraint60 As Constraint

Set          constraint60          =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference62)

constraint60.Mode = catCstModeDrivingDimension

Dim reference63 As Reference

Set          reference63          =
part1.CreateReferenceFromObject(point2D25)

Dim constraint61 As Constraint

Set          constraint61          =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference63)

constraint61.Mode = catCstModeDrivingDimension

Dim reference64 As Reference

Set          reference64          =
part1.CreateReferenceFromObject(point2D26)

Dim constraint62 As Constraint

Set          constraint62          =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference64)

constraint62.Mode = catCstModeDrivingDimension

Dim reference65 As Reference

Set          reference65          =
part1.CreateReferenceFromObject(point2D27)

Dim constraint63 As Constraint

Set          constraint63          =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference65)

constraint63.Mode = catCstModeDrivingDimension

Dim reference66 As Reference

Set          reference66          =
part1.CreateReferenceFromObject(point2D28)

Dim constraint64 As Constraint

Set          constraint64          =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference66)

constraint64.Mode = catCstModeDrivingDimension

Dim reference67 As Reference

Set          reference67          =
part1.CreateReferenceFromObject(point2D29)

Dim constraint65 As Constraint

Set          constraint65          =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference67)

constraint65.Mode = catCstModeDrivingDimension

Dim reference68 As Reference

Set          reference68          =
part1.CreateReferenceFromObject(point2D30)

Dim constraint66 As Constraint

Set          constraint66          =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference68)

constraint66.Mode = catCstModeDrivingDimension
.....
'En tercer lugar vamos a acotar los puntos del Intradós por su
distancia al eje vertical
.....

Dim reference69 As Reference

Set          reference69          =
part1.CreateReferenceFromObject(point2D31)

Dim constraint67 As Constraint

Set          constraint67          =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference69)

constraint67.Mode = catCstModeDrivingDimension

Dim reference70 As Reference

Set          reference70          =
part1.CreateReferenceFromObject(point2D32)

Dim constraint68 As Constraint

Set          constraint68          =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference70)

constraint68.Mode = catCstModeDrivingDimension

```



```

Dim reference71 As Reference
Set reference71 = part1.CreateReferenceFromObject(point2D33)
Dim constraint69 As Constraint
Set constraint69 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference71)
constraint69.Mode = catCstModeDrivingDimension
Dim reference72 As Reference
Set reference72 = part1.CreateReferenceFromObject(point2D34)
Dim constraint70 As Constraint
Set constraint70 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference72)
constraint70.Mode = catCstModeDrivingDimension
Dim reference73 As Reference
Set reference73 = part1.CreateReferenceFromObject(point2D35)
Dim constraint71 As Constraint
Set constraint71 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference73)
constraint71.Mode = catCstModeDrivingDimension
Dim reference74 As Reference
Set reference74 = part1.CreateReferenceFromObject(point2D36)
Dim constraint72 As Constraint
Set constraint72 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference74)
constraint72.Mode = catCstModeDrivingDimension
Dim reference75 As Reference
Set reference75 = part1.CreateReferenceFromObject(point2D37)
Dim constraint73 As Constraint
Set constraint73 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference75)
constraint73.Mode = catCstModeDrivingDimension
Dim reference76 As Reference
Set reference76 = part1.CreateReferenceFromObject(point2D38)
Dim constraint74 As Constraint
Set constraint74 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference76)
constraint74.Mode = catCstModeDrivingDimension
Dim reference77 As Reference
Set reference77 = part1.CreateReferenceFromObject(point2D39)
Dim constraint75 As Constraint
Set constraint75 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference77)
constraint75.Mode = catCstModeDrivingDimension
Dim reference78 As Reference
Set reference78 = part1.CreateReferenceFromObject(point2D40)
Dim constraint76 As Constraint
Set constraint76 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference78)
constraint76.Mode = catCstModeDrivingDimension
Dim reference79 As Reference
Set reference79 = part1.CreateReferenceFromObject(point2D41)
Dim constraint77 As Constraint
Set constraint77 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference79)
constraint77.Mode = catCstModeDrivingDimension
Dim reference80 As Reference
Set reference80 = part1.CreateReferenceFromObject(point2D42)
Dim constraint78 As Constraint
Set constraint78 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference80)
constraint78.Mode = catCstModeDrivingDimension
Dim reference81 As Reference
Set reference81 = part1.CreateReferenceFromObject(point2D43)
Dim constraint79 As Constraint
Set constraint79 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference81)
constraint79.Mode = catCstModeDrivingDimension
Dim reference82 As Reference
Set reference82 = part1.CreateReferenceFromObject(point2D44)
Dim constraint80 As Constraint
Set constraint80 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference82)
constraint80.Mode = catCstModeDrivingDimension
Dim reference83 As Reference
Set reference83 = part1.CreateReferenceFromObject(point2D45)
Dim constraint81 As Constraint

```

Set	constraint81	=	reference89)
constraints 1.AddBiEltCst(catCstTypeDistance,	reference6,		constraint87.Mode = catCstModeDrivingDimension
reference83)			Dim reference90 As Reference
constraint81.Mode = catCstModeDrivingDimension			Set
Dim reference84 As Reference			reference90
			=
Set	reference84	=	part 1.CreateReferenceFromObject(point2D52)
part 1.CreateReferenceFromObject(point2D46)			Dim constraint88 As Constraint
Dim constraint82 As Constraint			Set
			constraint88
Set	constraint82	=	constraints 1.AddBiEltCst(catCstTypeDistance,
constraints 1.AddBiEltCst(catCstTypeDistance,	reference6,		reference90)
reference84)			constraint88.Mode = catCstModeDrivingDimension
constraint82.Mode = catCstModeDrivingDimension			Dim reference91 As Reference
Dim reference85 As Reference			Set
			reference91
Set	reference85	=	part 1.CreateReferenceFromObject(point2D53)
part 1.CreateReferenceFromObject(point2D47)			Dim constraint89 As Constraint
Dim constraint83 As Constraint			Set
			constraint89
Set	constraint83	=	constraints 1.AddBiEltCst(catCstTypeDistance,
constraints 1.AddBiEltCst(catCstTypeDistance,	reference6,		reference91)
reference85)			constraint89.Mode = catCstModeDrivingDimension
constraint83.Mode = catCstModeDrivingDimension			Dim reference92 As Reference
Dim reference86 As Reference			Set
			reference92
Set	reference86	=	part 1.CreateReferenceFromObject(point2D54)
part 1.CreateReferenceFromObject(point2D48)			Dim constraint90 As Constraint
Dim constraint84 As Constraint			Set
			constraint90
Set	constraint84	=	constraints 1.AddBiEltCst(catCstTypeDistance,
constraints 1.AddBiEltCst(catCstTypeDistance,	reference6,		reference92)
reference86)			constraint90.Mode = catCstModeDrivingDimension
constraint84.Mode = catCstModeDrivingDimension			Dim reference93 As Reference
Dim reference87 As Reference			Set
			reference93
Set	reference87	=	part 1.CreateReferenceFromObject(point2D55)
part 1.CreateReferenceFromObject(point2D49)			Dim constraint91 As Constraint
Dim constraint85 As Constraint			Set
			constraint91
Set	constraint85	=	constraints 1.AddBiEltCst(catCstTypeDistance,
constraints 1.AddBiEltCst(catCstTypeDistance,	reference6,		reference93)
reference87)			constraint91.Mode = catCstModeDrivingDimension
constraint85.Mode = catCstModeDrivingDimension			Dim reference94 As Reference
Dim reference88 As Reference			Set
			reference94
Set	reference88	=	part 1.CreateReferenceFromObject(point2D56)
part 1.CreateReferenceFromObject(point2D50)			Dim constraint92 As Constraint
Dim constraint86 As Constraint			Set
			constraint92
Set	constraint86	=	constraints 1.AddBiEltCst(catCstTypeDistance,
constraints 1.AddBiEltCst(catCstTypeDistance,	reference6,		reference94)
reference88)			constraint92.Mode = catCstModeDrivingDimension
constraint86.Mode = catCstModeDrivingDimension			Dim reference95 As Reference
Dim reference89 As Reference			Set
			reference95
Set	reference89	=	part 1.CreateReferenceFromObject(point2D57)
part 1.CreateReferenceFromObject(point2D51)			Dim constraint93 As Constraint
Dim constraint87 As Constraint			Set
			constraint93
Set	constraint87	=	constraints 1.AddBiEltCst(catCstTypeDistance,
constraints 1.AddBiEltCst(catCstTypeDistance,	reference6,		reference95)
reference6,			constraint93.Mode = catCstModeDrivingDimension

```

Dim reference96 As Reference
Set reference96 = part1.CreateReferenceFromObject(point2D58)
Dim constraint94 As Constraint
Set constraint94 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference96)
constraint94.Mode = catCstModeDrivingDimension
Dim reference97 As Reference
Set reference97 = part1.CreateReferenceFromObject(point2D59)
Dim constraint95 As Constraint
Set constraint95 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference97)
constraint95.Mode = catCstModeDrivingDimension
Dim reference98 As Reference
Set reference98 = part1.CreateReferenceFromObject(point2D60)
Dim constraint96 As Constraint
Set constraint96 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference98)
constraint96.Mode = catCstModeDrivingDimension
.....
" Por último acotamos los puntos del Intradós por su
distancia al eje horizontal "
.....
Dim reference99 As Reference
Set reference99 = part1.CreateReferenceFromObject(point2D31)
Dim constraint97 As Constraint
Set constraint97 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference99)
constraint97.Mode = catCstModeDrivingDimension
Dim reference100 As Reference
Set reference100 = part1.CreateReferenceFromObject(point2D32)
Dim constraint98 As Constraint
Set constraint98 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference100)
constraint98.Mode = catCstModeDrivingDimension
Dim reference101 As Reference
Set reference101 = part1.CreateReferenceFromObject(point2D33)
Dim constraint99 As Constraint
Set constraint99 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference101)
constraint99.Mode = catCstModeDrivingDimension
Dim reference102 As Reference
Set reference102 = part1.CreateReferenceFromObject(point2D34)
Dim constraint100 As Constraint
Set constraint100 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference102)
constraint100.Mode = catCstModeDrivingDimension
Dim reference103 As Reference
Set reference103 = part1.CreateReferenceFromObject(point2D35)
Dim constraint101 As Constraint
Set constraint101 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference103)
constraint101.Mode = catCstModeDrivingDimension
Dim reference104 As Reference
Set reference104 = part1.CreateReferenceFromObject(point2D36)
Dim constraint102 As Constraint
Set constraint102 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference104)
constraint102.Mode = catCstModeDrivingDimension
Dim reference105 As Reference
Set reference105 = part1.CreateReferenceFromObject(point2D37)
Dim constraint103 As Constraint
Set constraint103 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference105)
constraint103.Mode = catCstModeDrivingDimension
Dim reference106 As Reference
Set reference106 = part1.CreateReferenceFromObject(point2D38)
Dim constraint104 As Constraint
Set constraint104 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference106)
constraint104.Mode = catCstModeDrivingDimension
Dim reference107 As Reference
Set reference107 = part1.CreateReferenceFromObject(point2D39)
Dim constraint105 As Constraint
Set constraint105 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference107)
constraint105.Mode = catCstModeDrivingDimension

```

Dim reference108 As Reference			part1.CreateReferenceFromObject(point2D46)
Set	reference108	=	Dim constraint112 As Constraint
part1.CreateReferenceFromObject(point2D40)			Set
Dim constraint106 As Constraint			constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference114)
Set	constraint106	=	constraint112.Mode = catCstModeDrivingDimension
constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference108)			Dim reference115 As Reference
constraint106.Mode = catCstModeDrivingDimension			Set
Dim reference109 As Reference			reference115
Set	reference109	=	part1.CreateReferenceFromObject(point2D47)
part1.CreateReferenceFromObject(point2D41)			Dim constraint113 As Constraint
Dim constraint107 As Constraint			Set
Set	constraint107	=	constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference115)
constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference109)			constraint113.Mode = catCstModeDrivingDimension
constraint107.Mode = catCstModeDrivingDimension			Dim reference116 As Reference
Dim reference110 As Reference			Set
Set	reference110	=	reference116
part1.CreateReferenceFromObject(point2D42)			part1.CreateReferenceFromObject(point2D48)
Dim constraint108 As Constraint			Dim constraint114 As Constraint
Set	constraint108	=	Set
constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference110)			constraint114
constraint108.Mode = catCstModeDrivingDimension			constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference116)
Dim reference111 As Reference			constraint114.Mode = catCstModeDrivingDimension
Set	reference111	=	Dim reference117 As Reference
part1.CreateReferenceFromObject(point2D43)			Set
Dim constraint109 As Constraint			reference117
Set	constraint109	=	part1.CreateReferenceFromObject(point2D49)
constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference111)			Dim constraint115 As Constraint
constraint109.Mode = catCstModeDrivingDimension			Set
Dim reference112 As Reference			constraint115
Set	reference112	=	constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference117)
part1.CreateReferenceFromObject(point2D44)			constraint115.Mode = catCstModeDrivingDimension
Dim constraint110 As Constraint			Dim reference118 As Reference
Set	constraint110	=	Set
constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference112)			reference118
constraint110.Mode = catCstModeDrivingDimension			part1.CreateReferenceFromObject(point2D50)
Dim reference113 As Reference			Dim constraint116 As Constraint
Set	reference113	=	Set
part1.CreateReferenceFromObject(point2D45)			constraint116
Dim constraint111 As Constraint			constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference118)
Set	constraint111	=	constraint116.Mode = catCstModeDrivingDimension
constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference113)			Dim reference119 As Reference
constraint111.Mode = catCstModeDrivingDimension			Set
Dim reference114 As Reference			reference119
Set	reference114	=	part1.CreateReferenceFromObject(point2D51)
			Dim constraint117 As Constraint
			Set
			constraint117
			constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference119)
			constraint117.Mode = catCstModeDrivingDimension
			Dim reference120 As Reference
			Set
			reference120
			part1.CreateReferenceFromObject(point2D52)
			Dim constraint118 As Constraint

```

Set constraint118 = constraint124.Mode = catCstModeDrivingDimension
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference120)
Dim reference127 As Reference
Set reference127 = part1.CreateReferenceFromObject(point2D59)
constraint118.Mode = catCstModeDrivingDimension
Dim constraint125 As Constraint
Dim reference121 As Reference
Set reference121 = part1.CreateReferenceFromObject(point2D53)
Set constraint125 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference127)
Dim constraint119 As Constraint
constraint125.Mode = catCstModeDrivingDimension
Dim reference128 As Reference
Set reference128 = part1.CreateReferenceFromObject(point2D60)
Dim constraint126 As Constraint
Set constraint126 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference128)
Dim constraint120 As Constraint
constraint126.Mode = catCstModeDrivingDimension
part1.CreateReferenceFromObject(point2D54)
" Acotamos el control point del borde de ataque "
Dim constraint120 As Constraint
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference122)
Dim reference129 As Reference
constraint120.Mode = catCstModeDrivingDimension
Set reference129 = part1.CreateReferenceFromObject(point2D61)
Dim reference123 As Reference
part1.CreateReferenceFromObject(point2D55)
Set reference129 = part1.CreateReferenceFromObject(point2D61)
Dim constraint121 As Constraint
Dim constraint127 As Constraint
Set constraint121 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference123)
Set constraint127 = constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference129)
constraint121.Mode = catCstModeDrivingDimension
constraint127.Mode = catCstModeDrivingDimension
Dim reference124 As Reference
Dim length127 As Length
Set length127 = constraint127.Dimension
length127.Value = 0
Set reference124 = part1.CreateReferenceFromObject(point2D56)
Dim reference130 As Reference
Set reference130 = part1.CreateReferenceFromObject(point2D61)
Dim constraint122 As Constraint
Dim constraint128 As Constraint
Set constraint122 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference124)
Set constraint128 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference130)
constraint122.Mode = catCstModeDrivingDimension
constraint128.Mode = catCstModeDrivingDimension
Dim reference125 As Reference
Dim length128 As Length
Set length128 = constraint128.Dimension
Dim formula5 As Formula
Set formula5 = relations.CreateFormula("Punto borde de
ataque encastre", "Borde de ataque", length128, "Cuerda")
Set reference125 = part1.CreateReferenceFromObject(point2D57)
' Una vez creado todos los puntos del perfil, realizamos un
spline alrededor de todos los puntos en un nuevo sketch.
Dim constraint123 As Constraint
' En dicho spline hemos unido todos los puntos exceptuando
los situados en el borde de salida. Ya que si unimos
Set constraint123 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference125)
Dim formula5 As Formula
constraint123.Mode = catCstModeDrivingDimension
' estos dos puntos en el mismo spline este se distorsiona.
Dim reference126 As Reference
Set reference126 = part1.CreateReferenceFromObject(point2D58)
Dim constraint124 As Constraint
Set constraint124 = constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference126)

```

```

Dim arrayOfObject1(60)
Set arrayOfObject1(0) = point2D30
Set arrayOfObject1(1) = point2D29
Set arrayOfObject1(2) = point2D28
Set arrayOfObject1(3) = point2D27
Set arrayOfObject1(4) = point2D26
Set arrayOfObject1(5) = point2D25
Set arrayOfObject1(6) = point2D24
Set arrayOfObject1(7) = point2D23
Set arrayOfObject1(8) = point2D22
Set arrayOfObject1(9) = point2D21
Set arrayOfObject1(10) = point2D20
Set arrayOfObject1(11) = point2D19
Set arrayOfObject1(12) = point2D18
Set arrayOfObject1(13) = point2D17
Set arrayOfObject1(14) = point2D16
Set arrayOfObject1(15) = point2D15
Set arrayOfObject1(16) = point2D14
Set arrayOfObject1(17) = point2D13
Set arrayOfObject1(18) = point2D12
Set arrayOfObject1(19) = point2D11
Set arrayOfObject1(20) = point2D10
Set arrayOfObject1(21) = point2D9
Set arrayOfObject1(22) = point2D8
Set arrayOfObject1(23) = point2D7
Set arrayOfObject1(24) = point2D6
Set arrayOfObject1(25) = point2D5
Set arrayOfObject1(26) = point2D4
Set arrayOfObject1(27) = point2D3
Set arrayOfObject1(28) = point2D2
Set arrayOfObject1(29) = point2D1
Set arrayOfObject1(30) = point2D61
Set arrayOfObject1(31) = point2D31
Set arrayOfObject1(32) = point2D32
Set arrayOfObject1(33) = point2D33
Set arrayOfObject1(34) = point2D34
Set arrayOfObject1(35) = point2D35
Set arrayOfObject1(36) = point2D36
Set arrayOfObject1(37) = point2D37
Set arrayOfObject1(38) = point2D38
Set arrayOfObject1(39) = point2D39
Set arrayOfObject1(40) = point2D40
Set arrayOfObject1(41) = point2D41
Set arrayOfObject1(42) = point2D42
Set arrayOfObject1(43) = point2D43
Set arrayOfObject1(44) = point2D44
Set arrayOfObject1(45) = point2D45
Set arrayOfObject1(46) = point2D46
Set arrayOfObject1(47) = point2D47
Set arrayOfObject1(48) = point2D48
Set arrayOfObject1(49) = point2D49
Set arrayOfObject1(50) = point2D50
Set arrayOfObject1(51) = point2D51
Set arrayOfObject1(52) = point2D52
Set arrayOfObject1(53) = point2D53
Set arrayOfObject1(54) = point2D54
Set arrayOfObject1(55) = point2D55
Set arrayOfObject1(56) = point2D56
Set arrayOfObject1(57) = point2D57
Set arrayOfObject1(58) = point2D58
Set arrayOfObject1(59) = point2D59
Set arrayOfObject1(60) = point2D60
Dim spline2D1 As Spline2D
Set factory2D1temp = factory2D1
Set          spline2D1          =
factory2D1temp.CreateSpline(arrayOfObject1)
spline2D1.ReportName = 71
.....
' Unimos los dos puntos del borde de salida mediante otro
spline el cual creará una línea recta entre estos dos puntos
' aunque lo ideal sería unirlos mediante un arco de
circunferencia el cual aun no sabemos hacer.
.....
Dim arrayOfObject2(1)
Set arrayOfObject2(0) = point2D30
Set arrayOfObject2(1) = point2D60
Dim spline2D2 As Spline2D
Set factory2D2temp = factory2D1
Set          spline2D2          =
factory2D2temp.CreateSpline(arrayOfObject2)
spline2D2.ReportName = 72
.....
' Cerramos el sketch
sketch1.CloseEdition
' Creamos las Regla a través de las cuales tendremos
parametrizados nuestros perfiles.
' Vamos a crear una regla para punto del perfil.
' Posteriormente modificamos la Regla e introducimos las

```

lineas de códigos obtenidas con MATLAB.

' Al ser lineas de códigos muy largas, estas no caben en una sola línea, por lo cual habrá que poner al final de cada línea

' " & _ para poder continuar debajo y cuando queramos que cambie de línea & vbCrLf &

Dim rule1 As Rule

Set rule1 = relations.CreateProgram("rule1", "", "")

' Modificamos la Regla e introducimos todas las líneas de códigos obtenidas con MATLAB

' Al ser muchas líneas de códigos estas no caben en una sola línea, por lo cual habrá que poner al final de cada línea

' " & _ para poder continuar debajo y cuando queramos que cambie de línea & vbCrLf &

rule1.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (20/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.7\Offset = (20/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/1
0)**2)*(NACA2/10)-(20/1000))))" else
PartBody\Sketch.1\Offset.7\Offset = (20/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(20/1000))))" & vbCrLf &
```

```
"if (20/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.37\Offset =
(Cuerda+(NACA1/100)*(20/1000*Cuerda)*(2*(NACA2/10)
-
20/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2
969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/1
0)**2)*(NACA2/10)-(20/1000))))" else
PartBody\Sketch.1\Offset.37\Offset =
(Cuerda+(NACA1/100)*(Cuerda-20/1000*Cuerda)*(-
2*(NACA2/10)+20/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(20/1000))))"
```

Dim rule2 As Rule

Set rule2 = relations.CreateProgram("rule2", "", "")

rule2.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (40/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.8\Offset = (40/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(40/1000)-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/1
0)**2)*(NACA2/10)-(40/1000))))" else
PartBody\Sketch.1\Offset.8\Offset = (40/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(40/1000)-
0.1260*(40/1000)-
```

```
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(40/1000))))" & vbCrLf &
```

```
"if (40/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.38\Offset =
(Cuerda+(NACA1/100)*(40/1000*Cuerda)*(2*(NACA2/10)
-
40/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2
969*sqrt(40/1000)-0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/1
0)**2)*(NACA2/10)-(40/1000))))" else
PartBody\Sketch.1\Offset.38\Offset =
(Cuerda+(NACA1/100)*(Cuerda-40/1000*Cuerda)*(-
2*(NACA2/10)+40/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
40/1000)-0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(40/1000))))"
```

Dim rule3 As Rule

Set rule3 = relations.CreateProgram("rule3", "", "")

rule3.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (60/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.9\Offset = (60/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/1
0)**2)*(NACA2/10)-(60/1000))))" else
PartBody\Sketch.1\Offset.9\Offset = (60/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(60/1000))))" & vbCrLf &
```

```
"if (60/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.39\Offset =
(Cuerda+(NACA1/100)*(60/1000*Cuerda)*(2*(NACA2/10)
-
60/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2
969*sqrt(60/1000)-0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/1
0)**2)*(NACA2/10)-(60/1000))))" else
PartBody\Sketch.1\Offset.39\Offset =
(Cuerda+(NACA1/100)*(Cuerda-60/1000*Cuerda)*(-
2*(NACA2/10)+60/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
60/1000)-0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(60/1000))))"
```

Dim rule4 As Rule

Set rule4 = relations.CreateProgram("rule4", "", "")

rule4.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (80/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.10\Offset = (80/1000*Cuerda -
```

```

NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(80/1000))))))
else
PartBody\Sketch.1\Offset.10\Offset = (80/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(80/1000))))))" & vbCrLf &
-
"if (80/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.40\Offset =
(Cuerda+(NACA1/100)*(80/1000*Cuerda)*(2*(NACA2/10)
-
80/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2
969*sqrt(80/1000)-0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(80/1000))))))
else
PartBody\Sketch.1\Offset.40\Offset =
(Cuerda+(NACA1/100)*(Cuerda-80/1000*Cuerda)*(-
2*(NACA2/10)+80/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
80/1000)-0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(80/1000))))))"

```

Dim rule5 As Rule

Set rule5 = relations.CreateProgram("rule5", "", "")

rule5.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```

"if (100/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.11\Offset = (100/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(100/1000))))))
else
PartBody\Sketch.1\Offset.11\Offset = (100/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(100/1000))))))" & vbCrLf &
_

```

```

"if (100/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.41\Offset =
(Cuerda+(NACA1/100)*(100/1000*Cuerda)*(2*(NACA2/10)
)-
100/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(100/1000)-0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(100/1000))))))
else
PartBody\Sketch.1\Offset.41\Offset =
(Cuerda+(NACA1/100)*(Cuerda-100/1000*Cuerda)*(-
2*(NACA2/10)+100/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
100/1000)-0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(100/1000))))))"

```

Dim rule6 As Rule

Set rule6 = relations.CreateProgram("rule6", "", "")

rule6.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```

"if (136/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.12\Offset = (136/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(136/1000))))))
else
PartBody\Sketch.1\Offset.12\Offset = (136/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(136/1000))))))" & vbCrLf &
_

```

```

"if (136/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.42\Offset =
(Cuerda+(NACA1/100)*(136/1000*Cuerda)*(2*(NACA2/10)
)-
136/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(136/1000)-0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(136/1000))))))
else
PartBody\Sketch.1\Offset.42\Offset =
(Cuerda+(NACA1/100)*(Cuerda-136/1000*Cuerda)*(-
2*(NACA2/10)+136/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
136/1000)-0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(136/1000))))))"

```

Dim rule7 As Rule

Set rule7 = relations.CreateProgram("rule7", "", "")

rule7.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```

"if (172/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.13\Offset = (172/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(172/1000))))))
else
PartBody\Sketch.1\Offset.13\Offset = (172/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(172/1000))))))" & vbCrLf &
_

```

```

"if (172/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.43\Offset =
(Cuerda+(NACA1/100)*(172/1000*Cuerda)*(2*(NACA2/10)
)-
172/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(172/1000)-0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(172/1000))))))
else

```



```

PartBody\Sketch.1\Offset.43\Offset = (Cuerda+((NACA1/100)*(Cuerda-172/1000*Cuerda)*(-
2*(NACA2/10)+172/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
172/1000)-0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(172/1000))))"
Dim rule8 As Rule
Set rule8 = relations.CreateProgram("rule8", "", "")
rule8.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (208/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.14\Offset = (208/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(208/1000))))" else
PartBody\Sketch.1\Offset.14\Offset = (208/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(208/1000))))" & vbCrLf
& _
"if (208/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.44\Offset =
(Cuerda+((NACA1/100)*(208/1000*Cuerda)*(2*(NACA2/10
)-
208/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(208/1000)-0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(208/1000))))" else
PartBody\Sketch.1\Offset.44\Offset =
(Cuerda+((NACA1/100)*(Cuerda-208/1000*Cuerda)*(-
2*(NACA2/10)+208/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
208/1000)-0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(208/1000))))"
Dim rule9 As Rule
Set rule9 = relations.CreateProgram("rule9", "", "")
rule9.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (244/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.15\Offset = (244/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(244/1000))))" else
PartBody\Sketch.1\Offset.15\Offset = (244/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(244/1000))))" & vbCrLf
& _
"if (244/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.45\Offset =
(Cuerda+((NACA1/100)*(244/1000*Cuerda)*(2*(NACA2/10
)-
244/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(244/1000)-0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(244/1000))))" else
PartBody\Sketch.1\Offset.45\Offset =
(Cuerda+((NACA1/100)*(Cuerda-244/1000*Cuerda)*(-
2*(NACA2/10)+244/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
244/1000)-0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(244/1000))))"
Dim rule10 As Rule
Set rule10 = relations.CreateProgram("rule10", "", "")
rule10.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (280/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.16\Offset = (280/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(280/1000))))" else
PartBody\Sketch.1\Offset.16\Offset = (280/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(280/1000))))" & vbCrLf
& _
"if (280/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.46\Offset =
(Cuerda+((NACA1/100)*(280/1000*Cuerda)*(2*(NACA2/10
)-
280/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(280/1000)-0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(280/1000))))" else
PartBody\Sketch.1\Offset.46\Offset =
(Cuerda+((NACA1/100)*(Cuerda-280/1000*Cuerda)*(-
2*(NACA2/10)+280/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
280/1000)-0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(280/1000))))"
Dim rule11 As Rule
Set rule11 = relations.CreateProgram("rule11", "", "")
rule11.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (316/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.17\Offset = (316/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(316/1000))))" else
PartBody\Sketch.1\Offset.17\Offset = (316/1000*Cuerda -

```

```
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10))**2)*((NACA2/10)-(316/1000))))" & vbCrLf
& _
```

```
"if (316/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.47\Offset =
(Cuerda+((NACA1/100)*(316/1000*Cuerda)*(2*(NACA2/10
)-
316/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(316/1000)-0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(316/1000))))" else
PartBody\Sketch.1\Offset.47\Offset =
(Cuerda+((NACA1/100)*(Cuerda-316/1000*Cuerda)*(-
2*(NACA2/10)+316/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
316/1000)-0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10))**2)*((NACA2/10)-(316/1000))))"
```

Dim rule12 As Rule

Set rule12 = relations.CreateProgram("rule12", "", "")

```
rule12.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (352/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.18\Offset = (352/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(352/1000))))" else
PartBody\Sketch.1\Offset.18\Offset = (352/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10))**2)*((NACA2/10)-(352/1000))))" & vbCrLf
& _
```

```
"if (352/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.48\Offset =
(Cuerda+((NACA1/100)*(352/1000*Cuerda)*(2*(NACA2/10
)-
352/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(352/1000)-0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(352/1000))))" else
PartBody\Sketch.1\Offset.48\Offset =
(Cuerda+((NACA1/100)*(Cuerda-352/1000*Cuerda)*(-
2*(NACA2/10)+352/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
352/1000)-0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10))**2)*((NACA2/10)-(352/1000))))"
```

Dim rule13 As Rule

Set rule13 = relations.CreateProgram("rule13", "", "")

```
rule13.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (388/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.19\Offset = (388/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(388/1000))))" else
PartBody\Sketch.1\Offset.19\Offset = (388/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10))**2)*((NACA2/10)-(388/1000))))" & vbCrLf
& _
```

```
"if (388/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.49\Offset =
(Cuerda+((NACA1/100)*(388/1000*Cuerda)*(2*(NACA2/10
)-
388/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(388/1000)-0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(388/1000))))" else
PartBody\Sketch.1\Offset.49\Offset =
(Cuerda+((NACA1/100)*(Cuerda-388/1000*Cuerda)*(-
2*(NACA2/10)+388/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
388/1000)-0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10))**2)*((NACA2/10)-(388/1000))))"
```

Dim rule14 As Rule

Set rule14 = relations.CreateProgram("rule14", "", "")

```
rule14.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (424/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.20\Offset = (424/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(424/1000))))" else
PartBody\Sketch.1\Offset.20\Offset = (424/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10))**2)*((NACA2/10)-(424/1000))))" & vbCrLf
& _
```

```
"if (424/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.50\Offset =
(Cuerda+((NACA1/100)*(424/1000*Cuerda)*(2*(NACA2/10
)-
424/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(424/1000)-0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(424/1000))))" else
PartBody\Sketch.1\Offset.50\Offset =
(Cuerda+((NACA1/100)*(Cuerda-424/1000*Cuerda)*(-
2*(NACA2/10)+424/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
424/1000)-0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
```

0.1015*(424/1000)**4)*cos(atan(2*(NACA1/100)/((1-(NACA2/10))**2)*((NACA2/10)-(424/1000))))"

Dim rule15 As Rule

Set rule15 = relations.CreateProgram("rule15", "", "")

rule15.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

"if (460/1000*Cuerda) <= (Cuerda*(NACA2/10)) PartBody\Sketch.1\Offset.21\Offset = (460/1000*Cuerda - NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(460/1000)-0.1260*(460/1000)-0.3516*(460/1000)**2+0.2843*(460/1000)**3-0.1015*(460/1000)**4)*sin(atan(2*(NACA1/100)/(NACA2/10)**2)*((NACA2/10)-(460/1000))))" else PartBody\Sketch.1\Offset.21\Offset = (460/1000*Cuerda - NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(460/1000)-0.1260*(460/1000)-0.3516*(460/1000)**2+0.2843*(460/1000)**3-0.1015*(460/1000)**4)*sin(atan(2*(NACA1/100)/((1-(NACA2/10))**2)*((NACA2/10)-(460/1000))))" & vbCrLf & _

"if (460/1000*Cuerda) <= (Cuerda*(NACA2/10)) PartBody\Sketch.1\Offset.51\Offset = (Cuerda+(NACA1/100)*(460/1000*Cuerda)*(2*(NACA2/10)-460/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(460/1000)-0.1260*(460/1000)-0.3516*(460/1000)**2+0.2843*(460/1000)**3-0.1015*(460/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(460/1000))))" else PartBody\Sketch.1\Offset.51\Offset = (Cuerda+(NACA1/100)*(Cuerda-460/1000*Cuerda)*(-2*(NACA2/10)+460/1000+1)/(1-NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(460/1000)-0.1260*(460/1000)-0.3516*(460/1000)**2+0.2843*(460/1000)**3-0.1015*(460/1000)**4)*cos(atan(2*(NACA1/100)/((1-(NACA2/10))**2)*((NACA2/10)-(460/1000))))"

Dim rule16 As Rule

Set rule16 = relations.CreateProgram("rule16", "", "")

rule16.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

"if (496/1000*Cuerda) <= (Cuerda*(NACA2/10)) PartBody\Sketch.1\Offset.22\Offset = (496/1000*Cuerda - NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(496/1000)-0.1260*(496/1000)-0.3516*(496/1000)**2+0.2843*(496/1000)**3-0.1015*(496/1000)**4)*sin(atan(2*(NACA1/100)/(NACA2/10)**2)*((NACA2/10)-(496/1000))))" else PartBody\Sketch.1\Offset.22\Offset = (496/1000*Cuerda - NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(496/1000)-0.1260*(496/1000)-0.3516*(496/1000)**2+0.2843*(496/1000)**3-0.1015*(496/1000)**4)*sin(atan(2*(NACA1/100)/((1-(NACA2/10))**2)*((NACA2/10)-(496/1000))))" & vbCrLf & _

"if (496/1000*Cuerda) <= (Cuerda*(NACA2/10)) PartBody\Sketch.1\Offset.52\Offset = (Cuerda+(NACA1/100)*(496/1000*Cuerda)*(2*(NACA2/10)-496/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(496/1000)-0.1260*(496/1000)-0.3516*(496/1000)**2+0.2843*(496/1000)**3-

0.1015*(496/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(496/1000))))" else PartBody\Sketch.1\Offset.52\Offset = (Cuerda+(NACA1/100)*(Cuerda-496/1000*Cuerda)*(-2*(NACA2/10)+496/1000+1)/(1-NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(496/1000)-0.1260*(496/1000)-0.3516*(496/1000)**2+0.2843*(496/1000)**3-0.1015*(496/1000)**4)*cos(atan(2*(NACA1/100)/((1-(NACA2/10))**2)*((NACA2/10)-(496/1000))))"

Dim rule17 As Rule

Set rule17 = relations.CreateProgram("rule17", "", "")

rule17.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

"if (532/1000*Cuerda) <= (Cuerda*(NACA2/10)) PartBody\Sketch.1\Offset.23\Offset = (532/1000*Cuerda - NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(532/1000)-0.1260*(532/1000)-0.3516*(532/1000)**2+0.2843*(532/1000)**3-0.1015*(532/1000)**4)*sin(atan(2*(NACA1/100)/(NACA2/10)**2)*((NACA2/10)-(532/1000))))" else PartBody\Sketch.1\Offset.23\Offset = (532/1000*Cuerda - NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(532/1000)-0.1260*(532/1000)-0.3516*(532/1000)**2+0.2843*(532/1000)**3-0.1015*(532/1000)**4)*sin(atan(2*(NACA1/100)/((1-(NACA2/10))**2)*((NACA2/10)-(532/1000))))" & vbCrLf & _

"if (532/1000*Cuerda) <= (Cuerda*(NACA2/10)) PartBody\Sketch.1\Offset.53\Offset = (Cuerda+(NACA1/100)*(532/1000*Cuerda)*(2*(NACA2/10)-532/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(532/1000)-0.1260*(532/1000)-0.3516*(532/1000)**2+0.2843*(532/1000)**3-0.1015*(532/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(532/1000))))" else PartBody\Sketch.1\Offset.53\Offset = (Cuerda+(NACA1/100)*(Cuerda-532/1000*Cuerda)*(-2*(NACA2/10)+532/1000+1)/(1-NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(532/1000)-0.1260*(532/1000)-0.3516*(532/1000)**2+0.2843*(532/1000)**3-0.1015*(532/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(532/1000))))"

Dim rule18 As Rule

Set rule18 = relations.CreateProgram("rule18", "", "")

rule18.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

"if (568/1000*Cuerda) <= (Cuerda*(NACA2/10)) PartBody\Sketch.1\Offset.24\Offset = (568/1000*Cuerda - NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(568/1000)-0.1260*(568/1000)-0.3516*(568/1000)**2+0.2843*(568/1000)**3-0.1015*(568/1000)**4)*sin(atan(2*(NACA1/100)/(NACA2/10)**2)*((NACA2/10)-(568/1000))))" else PartBody\Sketch.1\Offset.24\Offset = (568/1000*Cuerda - NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(568/1000)-0.1260*(568/1000)-0.3516*(568/1000)**2+0.2843*(568/1000)**3-0.1015*(568/1000)**4)*sin(atan(2*(NACA1/100)/((1-(NACA2/10))**2)*((NACA2/10)-(568/1000))))" & vbCrLf & _

```
"if (568/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.54\Offset =
(Cuerda+((NACA1/100)*(568/1000*Cuerda)*(2*(NACA2/10
)-
568/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(568/1000)-0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(568/1000))))" else
PartBody\Sketch.1\Offset.54\Offset =
(Cuerda+((NACA1/100)*(Cuerda-568/1000*Cuerda)*(-
2*(NACA2/10)+568/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
568/1000)-0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(568/1000))))"
```

Dim rule19 As Rule

Set rule19 = relations.CreateProgram("rule19", "", "")

rule19.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (604/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.25\Offset = (604/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(604/1000))))" else
PartBody\Sketch.1\Offset.25\Offset = (604/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(604/1000))))" & vbCrLf & _
```

```
"if (604/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.55\Offset =
(Cuerda+((NACA1/100)*(604/1000*Cuerda)*(2*(NACA2/10
)-
604/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(604/1000)-0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(604/1000))))" else
PartBody\Sketch.1\Offset.55\Offset =
(Cuerda+((NACA1/100)*(Cuerda-604/1000*Cuerda)*(-
2*(NACA2/10)+604/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
604/1000)-0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(604/1000))))"
```

Dim rule20 As Rule

Set rule20 = relations.CreateProgram("rule20", "", "")

rule20.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (640/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.26\Offset = (640/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
```

```
10)**2)*(NACA2/10)-(640/1000))))" else
PartBody\Sketch.1\Offset.26\Offset = (640/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(640/1000))))" & vbCrLf & _
```

```
"if (640/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.56\Offset =
(Cuerda+((NACA1/100)*(640/1000*Cuerda)*(2*(NACA2/10
)-
640/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(640/1000)-0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(640/1000))))" else
PartBody\Sketch.1\Offset.56\Offset =
(Cuerda+((NACA1/100)*(Cuerda-640/1000*Cuerda)*(-
2*(NACA2/10)+640/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
640/1000)-0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(640/1000))))"
```

Dim rule21 As Rule

Set rule21 = relations.CreateProgram("rule21", "", "")

rule21.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (676/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.27\Offset = (676/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(676/1000)-
0.1260*(676/1000)-
0.3516*(676/1000)**2+0.2843*(676/1000)**3-
0.1015*(676/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(676/1000))))" else
PartBody\Sketch.1\Offset.27\Offset = (676/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(676/1000)-
0.1260*(676/1000)-
0.3516*(676/1000)**2+0.2843*(676/1000)**3-
0.1015*(676/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(676/1000))))" & vbCrLf & _
```

```
"if (676/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.57\Offset =
(Cuerda+((NACA1/100)*(676/1000*Cuerda)*(2*(NACA2/10
)-
676/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(676/1000)-0.1260*(676/1000)-
0.3516*(676/1000)**2+0.2843*(676/1000)**3-
0.1015*(676/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(676/1000))))" else
PartBody\Sketch.1\Offset.57\Offset =
(Cuerda+((NACA1/100)*(Cuerda-676/1000*Cuerda)*(-
2*(NACA2/10)+676/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
676/1000)-0.1260*(676/1000)-
0.3516*(676/1000)**2+0.2843*(676/1000)**3-
0.1015*(676/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(676/1000))))"
```

Dim rule22 As Rule

Set rule22 = relations.CreateProgram("rule22", "", "")

rule22.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (712/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.28\Offset = (712/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(712/1000)))) else
PartBody\Sketch.1\Offset.28\Offset = (712/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(712/1000))))" & vbCrLf
& _
```

```
"if (712/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.58\Offset =
(Cuerda+(NACA1/100)*(712/1000*Cuerda)*(2*(NACA2/10
)-
712/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(712/1000)-0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(712/1000)))) else
PartBody\Sketch.1\Offset.58\Offset =
(Cuerda+(NACA1/100)*(Cuerda-712/1000*Cuerda)*(-
2*(NACA2/10)+712/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
712/1000)-0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(712/1000))))"
```

Dim rule23 As Rule

Set rule23 = relations.CreateProgram("rule23", "", "")

rule23.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (748/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.29\Offset = (748/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(748/1000)))) else
PartBody\Sketch.1\Offset.29\Offset = (748/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(748/1000))))" & vbCrLf
& _
```

```
"if (748/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.59\Offset =
(Cuerda+(NACA1/100)*(748/1000*Cuerda)*(2*(NACA2/10
)-
748/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(748/1000)-0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(748/1000)))) else
PartBody\Sketch.1\Offset.59\Offset =
(Cuerda+(NACA1/100)*(Cuerda-748/1000*Cuerda)*(-
2*(NACA2/10)+748/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
```

```
748/1000)-0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(748/1000))))"
```

Dim rule24 As Rule

Set rule24 = relations.CreateProgram("rule24", "", "")

rule24.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (784/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.30\Offset = (784/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(784/1000)))) else
PartBody\Sketch.1\Offset.30\Offset = (784/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(784/1000))))" & vbCrLf
& _
```

```
"if (784/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.60\Offset =
(Cuerda+(NACA1/100)*(784/1000*Cuerda)*(2*(NACA2/10
)-
784/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(784/1000)-0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(784/1000)))) else
PartBody\Sketch.1\Offset.60\Offset =
(Cuerda+(NACA1/100)*(Cuerda-784/1000*Cuerda)*(-
2*(NACA2/10)+784/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
784/1000)-0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(784/1000))))"
```

Dim rule25 As Rule

Set rule25 = relations.CreateProgram("rule25", "", "")

rule25.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (820/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.31\Offset = (820/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(820/1000)))) else
PartBody\Sketch.1\Offset.31\Offset = (820/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(820/1000))))" & vbCrLf
& _
```

```
"if (820/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.61\Offset =
(Cuerda+(NACA1/100)*(820/1000*Cuerda)*(2*(NACA2/10
)-
820/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
```

```

2969*sqrt(820/1000)-0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(820/1000))))" else
PartBody\Sketch.1\Offset.61\Offset =
(Cuerda+(NACA1/100)*(Cuerda-820/1000*Cuerda)*(-
2*(NACA2/10)+820/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
820/1000)-0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(820/1000))))"

```

Dim rule26 As Rule

Set rule26 = relations.CreateProgram("rule26", "", "")

rule26.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```

"if (856/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.32\Offset = (856/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(856/1000))))" else
PartBody\Sketch.1\Offset.32\Offset = (856/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(856/1000))))" & vbCrLf
& _

```

```

"if (856/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.62\Offset =
(Cuerda+(NACA1/100)*(856/1000*Cuerda)*(2*(NACA2/10
)-
856/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(856/1000)-0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(856/1000))))" else
PartBody\Sketch.1\Offset.62\Offset =
(Cuerda+(NACA1/100)*(Cuerda-856/1000*Cuerda)*(-
2*(NACA2/10)+856/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
856/1000)-0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(856/1000))))"

```

Dim rule27 As Rule

Set rule27 = relations.CreateProgram("rule27", "", "")

rule27.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```

"if (892/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.33\Offset = (892/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(892/1000))))" else
PartBody\Sketch.1\Offset.33\Offset = (892/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-

```

```

0.1015*(892/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(892/1000))))" & vbCrLf
& _

```

```

"if (892/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.63\Offset =
(Cuerda+(NACA1/100)*(892/1000*Cuerda)*(2*(NACA2/10
)-
892/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(892/1000)-0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(892/1000))))" else
PartBody\Sketch.1\Offset.63\Offset =
(Cuerda+(NACA1/100)*(Cuerda-892/1000*Cuerda)*(-
2*(NACA2/10)+892/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
892/1000)-0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(892/1000))))"

```

Dim rule28 As Rule

Set rule28 = relations.CreateProgram("rule28", "", "")

rule28.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```

"if (928/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.34\Offset = (928/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(928/1000))))" else
PartBody\Sketch.1\Offset.34\Offset = (928/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(928/1000))))" & vbCrLf
& _

```

```

"if (928/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.64\Offset =
(Cuerda+(NACA1/100)*(928/1000*Cuerda)*(2*(NACA2/10
)-
928/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(928/1000)-0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(928/1000))))" else
PartBody\Sketch.1\Offset.64\Offset =
(Cuerda+(NACA1/100)*(Cuerda-928/1000*Cuerda)*(-
2*(NACA2/10)+928/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
928/1000)-0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(928/1000))))"

```

Dim rule29 As Rule

Set rule29 = relations.CreateProgram("rule29", "", "")

rule29.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```

"if (964/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.35\Offset = (964/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(964/1000)-

```

```

0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(964/1000))))))" else
PartBody\Sket.ch.1\Offset.35\Offset = (964/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(964/1000))))))" & vbCrLf
& _

"if (964/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sket.ch.1\Offset.65\Offset =
(Cuerda+((NACA1/100)*(964/1000*Cuerda)*(2*(NACA2/10
)-
964/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(964/1000)-0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(964/1000))))))" else
PartBody\Sket.ch.1\Offset.65\Offset =
(Cuerda+((NACA1/100)*(Cuerda-964/1000*Cuerda)*(-
2*(NACA2/10)+964/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
964/1000)-0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(964/1000))))))"

Dim rule30 As Rule

Set rule30 = relations.CreateProgram("rule30", "", "")

rule30.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _

"if (1000/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sket.ch.1\Offset.36\Offset = (1000/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2
/10)**2)*((NACA2/10)-(1000/1000))))))" else
PartBody\Sket.ch.1\Offset.36\Offset = (1000/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(1000/1000))))))" & vbCrLf
& _

"if (1000/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sket.ch.1\Offset.66\Offset =
(Cuerda+((NACA1/100)*(1000/1000*Cuerda)*(2*(NACA2/1
0)-
1000/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(
0.2969*sqrt(1000/1000)-0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*cos(atan(2*(NACA1/100)/((NACA
2/10)**2)*((NACA2/10)-(1000/1000))))))" else
PartBody\Sket.ch.1\Offset.66\Offset =
(Cuerda+((NACA1/100)*(Cuerda-1000/1000*Cuerda)*(-
2*(NACA2/10)+1000/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
1000/1000)-0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(1000/1000))))))"

Dim rule31 As Rule

```

```

Set rule31 = relations.CreateProgram("rule31", "", "")

rule31.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _

"if (20/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sket.ch.1\Offset.67\Offset = (20/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(20/1000))))))" else
PartBody\Sket.ch.1\Offset.67\Offset = (20/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(20/1000))))))" & vbCrLf &
_

"if (20/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sket.ch.1\Offset.97\Offset =
(Cuerda+((NACA1/100)*(20/1000*Cuerda)*(2*(NACA2/10
)-20/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(20/1000))))))" else
PartBody\Sket.ch.1\Offset.97\Offset =
(Cuerda
+((NACA1/100)*(Cuerda-20/1000*Cuerda)*(-
2*(NACA2/10)+20/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(20/1000))))))"

Dim rule32 As Rule

Set rule32 = relations.CreateProgram("rule32", "", "")

rule32.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _

"if (40/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sket.ch.1\Offset.68\Offset = (40/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(40/1000)-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(40/1000))))))" else
PartBody\Sket.ch.1\Offset.68\Offset = (40/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(40/1000)-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(40/1000))))))" & vbCrLf &
_

"if (40/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sket.ch.1\Offset.98\Offset =
(Cuerda+((NACA1/100)*(40/1000*Cuerda)*(2*(NACA2/10
)-40/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(40/1000)-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(40/1000))))))" else
PartBody\Sket.ch.1\Offset.98\Offset =
(Cuerda
+((NACA1/100)*(Cuerda-40/1000*Cuerda)*(-
2*(NACA2/10)+40/1000+1)/(1-NACA2/10)**2)-

```

```
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(40/1000)-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(40/1000))))"
```

Dim rule33 As Rule

Set rule33 = relations.CreateProgram("rule33", "", "")

rule33.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (60/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.69\Offset = (60/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4*sin(atan(2*(NACA1/100)/((NACA2/1
0)**2)*(NACA2/10)-(60/1000))))" else
PartBody\Sketch.1\Offset.69\Offset = (60/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(60/1000))))" & vbCrLf & _
```

```
"if (60/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.99\Offset =
(Cuerda+(NACA1/100)*(60/1000*Cuerda)*(2*(NACA2/10)
-60/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/1
0)**2)*(NACA2/10)-(60/1000))))" else
PartBody\Sketch.1\Offset.99\Offset = (Cuerda
+((NACA1/100)*(Cuerda-60/1000*Cuerda)*(-
2*(NACA2/10)+60/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(60/1000))))"
```

Dim rule34 As Rule

Set rule34 = relations.CreateProgram("rule34", "", "")

rule34.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (80/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.70\Offset = (80/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4*sin(atan(2*(NACA1/100)/((NACA2/1
0)**2)*(NACA2/10)-(80/1000))))" else
PartBody\Sketch.1\Offset.70\Offset = (80/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(80/1000))))" & vbCrLf & _
```

```
"if (80/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.100\Offset =
(Cuerda+(NACA1/100)*(80/1000*Cuerda)*(2*(NACA2/10)
-80/1000)/(NACA2/10)**2)-
```

```
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/1
0)**2)*(NACA2/10)-(80/1000))))" else
PartBody\Sketch.1\Offset.100\Offset = (Cuerda
+((NACA1/100)*(Cuerda-80/1000*Cuerda)*(-
2*(NACA2/10)+80/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(80/1000))))"
```

Dim rule35 As Rule

Set rule35 = relations.CreateProgram("rule35", "", "")

rule35.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (100/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.71\Offset = (100/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(100/1000))))" else
PartBody\Sketch.1\Offset.71\Offset = (100/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(100/1000))))" & vbCrLf & _
```

```
"if (100/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.101\Offset =
(Cuerda+(NACA1/100)*(100/1000*Cuerda)*(2*(NACA2/10)
-100/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(100/1000))))" else
PartBody\Sketch.1\Offset.101\Offset = (Cuerda
+((NACA1/100)*(Cuerda-100/1000*Cuerda)*(-
2*(NACA2/10)+100/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(100/1000))))"
```

Dim rule36 As Rule

Set rule36 = relations.CreateProgram("rule36", "", "")

rule36.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (136/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.72\Offset = (136/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(136/1000))))" else
PartBody\Sketch.1\Offset.72\Offset = (136/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
```



```
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(136/1000))))" & vbCrLf
& _
```

```
"if (136/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.102\Offset =
(Cuerda+((NACA1/100)*(136/1000*Cuerda)*(2*(NACA2/10
)-136/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(136/1000))))" else
PartBody\Sketch.1\Offset.102\Offset = (Cuerda
+((NACA1/100)*(Cuerda-136/1000*Cuerda)*(-
2*(NACA2/10)+136/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(136/1000))))"
```

Dim rule37 As Rule

Set rule37 = relations.CreateProgram("rule37", "", "")

```
rule37.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (172/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.73\Offset = (172/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(172/1000))))" else
PartBody\Sketch.1\Offset.73\Offset = (172/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(172/1000))))" & vbCrLf
& _
```

```
"if (172/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.103\Offset =
(Cuerda+((NACA1/100)*(172/1000*Cuerda)*(2*(NACA2/10
)-172/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(172/1000))))" else
PartBody\Sketch.1\Offset.103\Offset = (Cuerda
+((NACA1/100)*(Cuerda-172/1000*Cuerda)*(-
2*(NACA2/10)+172/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(172/1000))))"
```

Dim rule38 As Rule

Set rule38 = relations.CreateProgram("rule38", "", "")

```
rule38.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (208/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.74\Offset = (208/1000*Cuerda +
```

```
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(208/1000))))" else
PartBody\Sketch.1\Offset.74\Offset = (208/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(208/1000))))" & vbCrLf
& _
```

```
"if (208/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.104\Offset =
(Cuerda+((NACA1/100)*(208/1000*Cuerda)*(2*(NACA2/10
)-208/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(208/1000))))" else
PartBody\Sketch.1\Offset.104\Offset = (Cuerda
+((NACA1/100)*(Cuerda-208/1000*Cuerda)*(-
2*(NACA2/10)+208/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(208/1000))))"
```

Dim rule39 As Rule

Set rule39 = relations.CreateProgram("rule39", "", "")

```
rule39.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (244/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.75\Offset = (244/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(244/1000))))" else
PartBody\Sketch.1\Offset.75\Offset = (244/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(244/1000))))" & vbCrLf
& _
```

```
"if (244/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.105\Offset =
(Cuerda+((NACA1/100)*(244/1000*Cuerda)*(2*(NACA2/10
)-244/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(244/1000))))" else
PartBody\Sketch.1\Offset.105\Offset = (Cuerda
+((NACA1/100)*(Cuerda-244/1000*Cuerda)*(-
2*(NACA2/10)+244/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(244/1000))))"
```

Dim rule40 As Rule

Set rule40 = relations.CreateProgram("rule40", "", "")

rule40.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (280/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.76\Offset = (280/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(280/1000)))) else
PartBody\Sketch.1\Offset.76\Offset = (280/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(280/1000))))" & vbCrLf
& _
```

```
"if (280/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.106\Offset =
(Cuerda+((NACA1/100)*(280/1000*Cuerda)*(2*(NACA2/10)-
280/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(280/1000)))) else
PartBody\Sketch.1\Offset.106\Offset = (Cuerda
+((NACA1/100)*(Cuerda-280/1000*Cuerda)*(-
2*(NACA2/10)+280/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(280/1000))))" & vbCrLf
& _
```

Dim rule41 As Rule

Set rule41 = relations.CreateProgram("rule41", "", "")

rule41.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (316/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.77\Offset = (316/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(316/1000)))) else
PartBody\Sketch.1\Offset.77\Offset = (316/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(316/1000))))" & vbCrLf
& _
```

```
"if (316/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.107\Offset =
(Cuerda+((NACA1/100)*(316/1000*Cuerda)*(2*(NACA2/10)-
316/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(316/1000)))) else
```

```
PartBody\Sketch.1\Offset.107\Offset = (Cuerda
+((NACA1/100)*(Cuerda-316/1000*Cuerda)*(-
2*(NACA2/10)+316/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(316/1000))))" & vbCrLf
& _
```

Dim rule42 As Rule

Set rule42 = relations.CreateProgram("rule42", "", "")

rule42.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (352/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.78\Offset = (352/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(352/1000)))) else
PartBody\Sketch.1\Offset.78\Offset = (352/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(352/1000))))" & vbCrLf
& _
```

```
"if (352/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.108\Offset =
(Cuerda+((NACA1/100)*(352/1000*Cuerda)*(2*(NACA2/10)-
352/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(352/1000)))) else
PartBody\Sketch.1\Offset.108\Offset =
(Cuerda+((NACA1/100)*(Cuerda-352/1000*Cuerda)*(-
2*(NACA2/10)+352/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(352/1000))))" & vbCrLf
& _
```

Dim rule43 As Rule

Set rule43 = relations.CreateProgram("rule43", "", "")

rule43.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (388/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.79\Offset = (388/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(388/1000)))) else
PartBody\Sketch.1\Offset.79\Offset = (388/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(388/1000))))" & vbCrLf
& _
```

```
"if (388/1000*Cuerda) <= (Cuerda*(NACA2/10))
```

```

PartBody\Sketch.1\Offset.109\Offset =
(Cuerda+((NACA1/100)*(388/1000*Cuerda)*(2*(NACA2/10
)-388/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(388/1000)))) else
PartBody\Sketch.1\Offset.109\Offset =
(Cuerda+((NACA1/100)*(Cuerda-388/1000*Cuerda)*(-
2*(NACA2/10)+388/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(388/1000))))"
Dim rule44 As Rule
Set rule44 = relations.CreateProgram("rule44", "", "")
rule44.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (424/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.80\Offset = (424/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4)*sin(atan(2*(NACA1/100)/(NACA2/
10)**2)*((NACA2/10)-(424/1000)))) else
PartBody\Sketch.1\Offset.80\Offset = (424/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(424/1000))))" & vbCrLf
& _
"if (424/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.110\Offset =
(Cuerda+((NACA1/100)*(424/1000*Cuerda)*(2*(NACA2/10
)-424/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(424/1000)))) else
PartBody\Sketch.1\Offset.110\Offset =
(Cuerda+((NACA1/100)*(Cuerda-424/1000*Cuerda)*(-
2*(NACA2/10)+424/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(424/1000))))"
Dim rule45 As Rule
Set rule45 = relations.CreateProgram("rule45", "", "")
rule45.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (460/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.81\Offset = (460/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(460/1000)-
0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4)*sin(atan(2*(NACA1/100)/(NACA2/
10)**2)*((NACA2/10)-(460/1000)))) else
PartBody\Sketch.1\Offset.81\Offset = (460/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(460/1000)-
0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(460/1000))))" & vbCrLf
& _
"if (460/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.111\Offset =
(Cuerda+((NACA1/100)*(460/1000*Cuerda)*(2*(NACA2/10
)-460/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(460/1000)-
0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(460/1000)))) else
PartBody\Sketch.1\Offset.111\Offset =
(Cuerda+((NACA1/100)*(Cuerda-460/1000*Cuerda)*(-
2*(NACA2/10)+460/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(460/1000)-
0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(460/1000))))"
Dim rule46 As Rule
Set rule46 = relations.CreateProgram("rule46", "", "")
rule46.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (496/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.82\Offset = (496/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(496/1000)-
0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4)*sin(atan(2*(NACA1/100)/(NACA2/
10)**2)*((NACA2/10)-(496/1000)))) else
PartBody\Sketch.1\Offset.82\Offset = (496/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(496/1000)-
0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(496/1000))))" & vbCrLf
& _
"if (496/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.112\Offset =
(Cuerda+((NACA1/100)*(496/1000*Cuerda)*(2*(NACA2/10
)-496/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(496/1000)-
0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(496/1000)))) else
PartBody\Sketch.1\Offset.112\Offset =
(Cuerda+((NACA1/100)*(Cuerda-496/1000*Cuerda)*(-
2*(NACA2/10)+496/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(496/1000)-
0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(496/1000))))"
Dim rule47 As Rule
Set rule47 = relations.CreateProgram("rule47", "", "")
rule47.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _

```

```
"if (532/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.83\Offset = (532/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(532/1000)))) else
PartBody\Sketch.1\Offset.83\Offset = (532/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(532/1000))))" & vbCrLf
& _
```

```
"if (532/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.113\Offset =
(Cuerda+((NACA1/100)*(532/1000*Cuerda)*(2*(NACA2/10
)-532/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(532/1000)))) else
PartBody\Sketch.1\Offset.113\Offset =
(Cuerda+((NACA1/100)*(Cuerda-532/1000*Cuerda)*(-
2*(NACA2/10)+532/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(532/1000))))"
```

Dim rule48 As Rule

Set rule48 = relations.CreateProgram("rule48", "", "")

rule48.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (568/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.84\Offset = (568/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(568/1000)))) else
PartBody\Sketch.1\Offset.84\Offset = (568/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(568/1000))))" & vbCrLf
& _
```

```
"if (568/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.114\Offset =
(Cuerda+((NACA1/100)*(568/1000*Cuerda)*(2*(NACA2/10
)-568/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(568/1000)))) else
PartBody\Sketch.1\Offset.114\Offset =
(Cuerda+((NACA1/100)*(Cuerda-568/1000*Cuerda)*(-
2*(NACA2/10)+568/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
```

```
0.1015*(568/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(568/1000))))"
```

Dim rule49 As Rule

Set rule49 = relations.CreateProgram("rule49", "", "")

rule49.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (604/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.85\Offset = (604/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(604/1000)))) else
PartBody\Sketch.1\Offset.85\Offset = (604/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(604/1000))))" & vbCrLf
& _
```

```
"if (604/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.115\Offset =
(Cuerda+((NACA1/100)*(604/1000*Cuerda)*(2*(NACA2/10
)-604/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(604/1000)))) else
PartBody\Sketch.1\Offset.115\Offset =
(Cuerda+((NACA1/100)*(Cuerda-604/1000*Cuerda)*(-
2*(NACA2/10)+604/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(604/1000))))"
```

Dim rule50 As Rule

Set rule50 = relations.CreateProgram("rule50", "", "")

rule50.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (640/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.86\Offset = (640/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(640/1000)))) else
PartBody\Sketch.1\Offset.86\Offset = (640/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(640/1000))))" & vbCrLf
& _
```

```
"if (640/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.116\Offset =
(Cuerda+((NACA1/100)*(640/1000*Cuerda)*(2*(NACA2/10
)-640/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
```

```
0.1015*(640/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(640/1000))))
else
PartBody\Sketch.1\Offset.116\Offset =
(Cuerda+((NACA1/100)*(Cuerda-640/1000*Cuerda)*(-2*(NACA2/10)+640/1000+1)/(1-NACA2/10)**2)-NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(640/1000)-0.1260*(640/1000)-0.3516*(640/1000)**2+0.2843*(640/1000)**3-0.1015*(640/1000)**4*cos(atan(2*(NACA1/100)/((1-(NACA2/10)**2)*((NACA2/10)-(640/1000))))))"
```

Dim rule51 As Rule

Set rule51 = relations.CreateProgram("rule51", "", "")

rule51.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (676/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.87\Offset = (676/1000*Cuerda + NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(676/1000)-0.1260*(676/1000)-0.3516*(676/1000)**2+0.2843*(676/1000)**3-0.1015*(676/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(676/1000))))
else
PartBody\Sketch.1\Offset.87\Offset = (676/1000*Cuerda + NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(676/1000)-0.1260*(676/1000)-0.3516*(676/1000)**2+0.2843*(676/1000)**3-0.1015*(676/1000)**4)*sin(atan(2*(NACA1/100)/((1-(NACA2/10)**2)*((NACA2/10)-(676/1000)))))" & vbCrLf & _
```

```
"if (676/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.117\Offset =
(Cuerda+((NACA1/100)*(676/1000*Cuerda)*(2*(NACA2/10)-676/1000)/(NACA2/10)**2)-NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(676/1000)-0.1260*(676/1000)-0.3516*(676/1000)**2+0.2843*(676/1000)**3-0.1015*(676/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(676/1000))))
else
PartBody\Sketch.1\Offset.117\Offset =
(Cuerda+((NACA1/100)*(Cuerda-676/1000*Cuerda)*(-2*(NACA2/10)+676/1000+1)/(1-NACA2/10)**2)-NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(676/1000)-0.1260*(676/1000)-0.3516*(676/1000)**2+0.2843*(676/1000)**3-0.1015*(676/1000)**4)*cos(atan(2*(NACA1/100)/((1-(NACA2/10)**2)*((NACA2/10)-(676/1000)))))"
```

Dim rule52 As Rule

Set rule52 = relations.CreateProgram("rule52", "", "")

rule52.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (712/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.88\Offset = (712/1000*Cuerda + NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(712/1000)-0.1260*(712/1000)-0.3516*(712/1000)**2+0.2843*(712/1000)**3-0.1015*(712/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(712/1000))))
else
PartBody\Sketch.1\Offset.88\Offset = (712/1000*Cuerda + NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(712/1000)-0.1260*(712/1000)-0.3516*(712/1000)**2+0.2843*(712/1000)**3-0.1015*(712/1000)**4)*sin(atan(2*(NACA1/100)/((1-(NACA2/10)**2)*((NACA2/10)-(712/1000))))" & vbCrLf & _
```

```
"if (712/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.118\Offset =
(Cuerda+((NACA1/100)*(712/1000*Cuerda)*(2*(NACA2/10)-712/1000)/(NACA2/10)**2)-NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(712/1000)-0.1260*(712/1000)-0.3516*(712/1000)**2+0.2843*(712/1000)**3-0.1015*(712/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(712/1000))))
else
PartBody\Sketch.1\Offset.118\Offset =
(Cuerda+((NACA1/100)*(Cuerda-712/1000*Cuerda)*(-2*(NACA2/10)+712/1000+1)/(1-NACA2/10)**2)-NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(712/1000)-0.1260*(712/1000)-0.3516*(712/1000)**2+0.2843*(712/1000)**3-0.1015*(712/1000)**4)*cos(atan(2*(NACA1/100)/((1-(NACA2/10)**2)*((NACA2/10)-(712/1000))))"
```

Dim rule53 As Rule

Set rule53 = relations.CreateProgram("rule53", "", "")

rule53.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (748/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.89\Offset = (748/1000*Cuerda + NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(748/1000)-0.1260*(748/1000)-0.3516*(748/1000)**2+0.2843*(748/1000)**3-0.1015*(748/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(748/1000))))
else
PartBody\Sketch.1\Offset.89\Offset = (748/1000*Cuerda + NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(748/1000)-0.1260*(748/1000)-0.3516*(748/1000)**2+0.2843*(748/1000)**3-0.1015*(748/1000)**4)*sin(atan(2*(NACA1/100)/((1-(NACA2/10)**2)*((NACA2/10)-(748/1000))))" & vbCrLf & _
```

```
"if (748/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.119\Offset =
(Cuerda+((NACA1/100)*(748/1000*Cuerda)*(2*(NACA2/10)-748/1000)/(NACA2/10)**2)-NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(748/1000)-0.1260*(748/1000)-0.3516*(748/1000)**2+0.2843*(748/1000)**3-0.1015*(748/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(748/1000))))
else
PartBody\Sketch.1\Offset.119\Offset =
(Cuerda+((NACA1/100)*(Cuerda-748/1000*Cuerda)*(-2*(NACA2/10)+748/1000+1)/(1-NACA2/10)**2)-NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(748/1000)-0.1260*(748/1000)-0.3516*(748/1000)**2+0.2843*(748/1000)**3-0.1015*(748/1000)**4)*cos(atan(2*(NACA1/100)/((1-(NACA2/10)**2)*((NACA2/10)-(748/1000))))"
```

Dim rule54 As Rule

Set rule54 = relations.CreateProgram("rule54", "", "")

rule54.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (784/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.90\Offset = (784/1000*Cuerda + NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(784/1000)-0.1260*(784/1000)-0.3516*(784/1000)**2+0.2843*(784/1000)**3-0.1015*(784/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(784/1000))))
else
```

```
PartBody\Sketch.1\Offset.90\Offset = (784/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(784/1000))))" & vbCrLf
& _
```

```
"if (784/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.120\Offset =
(Cuerda+(NACA1/100)*(784/1000*Cuerda)*(2*(NACA2/10
)-784/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(784/1000))))" else
PartBody\Sketch.1\Offset.120\Offset =
(Cuerda+(NACA1/100)*(Cuerda-784/1000*Cuerda)*(-
2*(NACA2/10)+784/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(784/1000))))"
```

Dim rule55 As Rule

Set rule55 = relations.CreateProgram("rule55", "", "")

rule55.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (820/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.91\Offset = (820/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(820/1000))))" else
PartBody\Sketch.1\Offset.91\Offset = (820/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(820/1000))))" & vbCrLf
& _
```

```
"if (820/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.121\Offset =
(Cuerda+(NACA1/100)*(820/1000*Cuerda)*(2*(NACA2/10
)-820/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(820/1000))))" else
PartBody\Sketch.1\Offset.121\Offset =
(Cuerda+(NACA1/100)*(Cuerda-820/1000*Cuerda)*(-
2*(NACA2/10)+820/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(820/1000))))"
```

Dim rule56 As Rule

Set rule56 = relations.CreateProgram("rule56", "", "")

rule56.Modify "/*Rule created by Francisco Javier Jiménez

Calvo*/" & vbCrLf & _

```
"if (856/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.92\Offset = (856/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(856/1000))))" else
PartBody\Sketch.1\Offset.92\Offset = (856/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(856/1000))))" & vbCrLf
& _
```

```
"if (856/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.122\Offset =
(Cuerda+(NACA1/100)*(856/1000*Cuerda)*(2*(NACA2/10
)-856/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(856/1000))))" else
PartBody\Sketch.1\Offset.122\Offset =
(Cuerda+(NACA1/100)*(Cuerda-856/1000*Cuerda)*(-
2*(NACA2/10)+856/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(856/1000))))"
```

Dim rule57 As Rule

Set rule57 = relations.CreateProgram("rule57", "", "")

rule57.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (892/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.93\Offset = (892/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(892/1000))))" else
PartBody\Sketch.1\Offset.93\Offset = (892/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(892/1000))))" & vbCrLf
& _
```

```
"if (892/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.123\Offset =
(Cuerda+(NACA1/100)*(892/1000*Cuerda)*(2*(NACA2/10
)-892/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(892/1000))))" else
PartBody\Sketch.1\Offset.123\Offset =
(Cuerda+(NACA1/100)*(Cuerda-892/1000*Cuerda)*(-
2*(NACA2/10)+892/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
```

```

0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(892/1000))))"
Dim rule58 As Rule
Set rule58 = relations.CreateProgram("rule58", "", "")
rule58.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (928/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.94\Offset = (928/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4)*sin(atan(2*(NACA1/100)/(NACA2/
10)**2*(NACA2/10)-(928/1000))))" else
PartBody\Sketch.1\Offset.94\Offset = (928/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(928/1000))))" & vbCrLf
& _
"if (928/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.124\Offset =
(Cuerda+(NACA1/100)*(928/1000*Cuerda)*(2*(NACA2/10
)-928/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2*(NACA2/10)-(928/1000))))" else
PartBody\Sketch.1\Offset.124\Offset =
(Cuerda+(NACA1/100)*(Cuerda-928/1000*Cuerda)*(-
2*(NACA2/10)+928/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(928/1000))))"
Dim rule59 As Rule
Set rule59 = relations.CreateProgram("rule59", "", "")
rule59.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (964/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.95\Offset = (964/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*sin(atan(2*(NACA1/100)/(NACA2/
10)**2*(NACA2/10)-(964/1000))))" else
PartBody\Sketch.1\Offset.95\Offset = (964/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(964/1000))))" & vbCrLf
& _
"if (964/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.125\Offset =
(Cuerda+(NACA1/100)*(964/1000*Cuerda)*(2*(NACA2/10
)-964/1000)/(NACA2/10)**2)-

```

```

NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2*(NACA2/10)-(964/1000))))" else
PartBody\Sketch.1\Offset.125\Offset =
(Cuerda+(NACA1/100)*(Cuerda-964/1000*Cuerda)*(-
2*(NACA2/10)+964/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(964/1000))))"
Dim rule60 As Rule
Set rule60 = relations.CreateProgram("rule60", "", "")
rule60.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (1000/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.96\Offset = (1000/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*sin(atan(2*(NACA1/100)/(NACA2
/10)**2*(NACA2/10)-(1000/1000))))" else
PartBody\Sketch.1\Offset.96\Offset = (1000/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(1000/1000))))" & vbCrLf
& _
"if (1000/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.126\Offset =
(Cuerda+(NACA1/100)*(1000/1000*Cuerda)*(2*(NACA2/1
0)-1000/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*cos(atan(2*(NACA1/100)/((NACA
2/10)**2*(NACA2/10)-(1000/1000))))" else
PartBody\Sketch.1\Offset.126\Offset =
(Cuerda+(NACA1/100)*(Cuerda-1000/1000*Cuerda)*(-
2*(NACA2/10)+1000/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(1000/1000))))"
'Se actualiza el documento y se cierra la interface
CATIA.ActiveDocument.Part.Update
Unload Me
End Sub

```

ANEXO 3. CÓDIGO DE LA ESTRUCTURA PRINCIPAL PARA DIBUJAR UN ALA

```
' Para que el programa funcione bien debemos estar trabajando
en un documento .CATPart y que no existan parámetros

' que se llamen iguales a los que creamos con nuestro programa
ya que entrarían en conflicto. Para evitar todos

' estos problemas vamos a hacer que este programa siempre se
ejecute en un nuevo documento .CATPart.

' De esta forma el programa pueda arrancar sin necesidad de
tener un part abierto, de forma que al ejecutarse se

' abra un nuevo part de forma automática.

' Al crearse un nuevo part cada vez que se ejecuta el programa,
no es necesario comprobar que estamos trabajando con

' archivo .CATPart ni que en dicho archivo existan ya
parámetros que se llamen igual que los que vamos a crear

' ya que al ser un nuevo archivo estará totalmente vacío.

' Normalmente se suele nombrar al objeto con el mismo
nombre de la orden seguida de un número para identificar

' rápidamente que estamos haciendo, esta es la lógica que
seguiremos en nuestro programa.

' Creamos y abrimos el nuevo Part
Dim documents1 As Documents
Dim partDocument1 As PartDocument
Dim part1 As Part
Set documents1 = CATIA.Documents
Set partDocument1 = documents1.Add("Part") ' Se carga
un documento activo part para trabajar
Set part1 = partDocument1.Part ' Se establece el
part1 como entorno para trabajar

' Activamos las relaciones y parámetros
Dim relations As relations
Dim parameters As parameters
Set relations = part1.relations
Set parameters = part1.parameters

' Asignamos los valores introducidos a cada parámetro
NACA1Valor = TextBox1
NACA2Valor = TextBox2
NACA34Valor = TextBox3
CuerdaValor = TextBox4
NACA1TValor = TextBox5
NACA2TValor = TextBox6
NACA34TValor = TextBox7
EstrechamientoValor = TextBox8
IncidenciaValor = TextBox9
DiedroValor = TextBox10
```

```
FlechaValor = TextBox11
SemienvergaduraValor = TextBox12
TorsionValor = TextBox13

' Se comprueba que los valores introducidos son correctos, si
estos no lo están se lanza un mensaje

' de advertencia y se para el programa para que los
modifiquemos y volvamos a ejecutar el programa.

' NACA1 debe ser un número entero positivo de una cifra
If NACA1Valor > -1 And NACA1Valor < 10 Then
Else
MsgBox "El valor introducido en NACA 1 en perfil encastre
debe estar comprendido entre 0 y 9."
End ' Este End se pone para parar el programa
si se encuentra el error
End If

' NACA2 debe ser un número entero positivo de una cifra
If NACA2Valor > -1 And NACA2Valor < 10 Then
Else
MsgBox "El valor introducido en NACA 2 en perfil encastre
debe estar comprendido entre 0 y 9."
End
End If

' NACA34 debe ser un número entero positivo este puede ir
desde el 0 al 99.
If NACA34Valor > -1 And NACA34Valor < 100 Then
Else
MsgBox "El valor introducido en NACA 34 en perfil encastre
debe estar comprendido entre 0 y 99."
End
End If

' NACA1 raiz debe ser un número entero positivo de una cifra
If NACA1TValor > -1 And NACA1TValor < 10 Then
Else
MsgBox "El valor introducido en NACA 1 en perfil raiz debe
estar comprendido entre 0 y 9."
End ' Este End se pone para parar el programa
si se encuentra el error
End If

' NACA2 raiz debe ser un número entero positivo de una cifra
If NACA2TValor > -1 And NACA2TValor < 10 Then
Else
MsgBox "El valor introducido en NACA 2 en perfil raiz debe
estar comprendido entre 0 y 9."
```



```

End
End If
' NACA34 raiz debe ser un número entero positivo este puede
ir desde el 0 al 99.
If NACA34TValor > -1 And NACA34Valor < 100 Then
Else
MsgBox "El valor introducido en NACA 34 en perfil raiz debe
estar comprendido entre 0 y 99."
End
End If
' Cuerda debe ser un número entero positivo mayor que 0
If CuerdaValor > 0 Then
Else
MsgBox "El valor introducido en Cuerda debe ser mayor que
0."
End
End If
If DiedroValor > -46 And DiedroValor < 46 Then
Else
MsgBox "El valor introducido en Diedro debe estar
comprendido entre -45° y 45°."
End
End If
If IncidenciaValor > -0.001 And IncidenciaValor < 46 Then
Else
MsgBox "El valor introducido en Incidencia debe estar
comprendido entre 0° y 45°."
End
End If
If FlechaValor > -0.001 And FlechaValor < 46 Then
Else
MsgBox "El valor introducido en Flecha debe estar
comprendido entre 0° y 45°."
End
End If
If SemienvergaduraValor > 0 Then
Else
MsgBox "El valor introducido en Semienvergadura debe ser
mayor que 0."
End
End If
' Estrechamiento debe ser un numero entre 0 y 100 ya que
refleja el % de reducción de la cuerda en la punta del ala.
If EstrechamientoValor > -0.001 And EstrechamientoValor <
101 Then
Else
MsgBox "El valor introducido en Estrechamiento debe estar
comprendido entre 0 y 100."
End
End If
' Creamos los parámetros
Dim NACA1 As Parameter
Set NACA1 = parameters.CreateInteger("NACA1", 0)
Dim NACA2 As Parameter
Set NACA2 = parameters.CreateInteger("NACA2", 0)
Dim NACA34 As Parameter
Set NACA34 = parameters.CreateInteger("NACA34", 0)
Dim NACA1T As Parameter
Set NACA1T = parameters.CreateInteger("NACA1T", 0)
Dim NACA2T As Parameter
Set NACA2T = parameters.CreateInteger("NACA2T", 0)
Dim NACA34T As Parameter
Set NACA34T = parameters.CreateInteger("NACA34T", 0)
Dim Cuerda As Parameter
Set Cuerda = parameters.CreateDimension("Cuerda",
"LENGTH", 0)
Dim Diedro As Parameter
Set Diedro = parameters.CreateDimension("Diedro",
"ANGLE", 0)
Dim Incidencia As Parameter
Set Incidencia = parameters.CreateDimension("Incidencia",
"ANGLE", 0)
Dim Flecha As Parameter
Set Flecha = parameters.CreateDimension("Flecha",
"ANGLE", 0)
Dim Semienvergadura As Parameter
Set Semienvergadura = parameters.CreateDimension("Semienvergadura",
"LENGTH", 0)
Dim Estrechamiento As Parameter
Set Estrechamiento = parameters.CreateInteger("Estrechamiento", 0)
Dim Torsion As Parameter
Set Torsion = parameters.CreateDimension("Torsion",
"ANGLE", 0)
' Asignamos el valor correspondiente a cada parámetro
NACA1.Value = NACA1Valor
NACA2.Value = NACA2Valor

```

```

NACA34.Value = NACA34Valor
NACA1T.Value = NACA1TValor
NACA2T.Value = NACA2TValor
NACA34T.Value = NACA34TValor
Cuerda.Value = CuerdaValor
Diedro.Value = DiedroValor
Flecha.Value = FlechaValor
Semienvergadura.Value = SemienvergaduraValor
Estrechamiento.Value = EstrechamientoValor
Torsion.Value = TorsionValor
Incidencia.Value = IncidenciaValor
.....
"""          PERFIL ENCASTRE ALA          """
.....
' Ahora debemos crear 60 puntos, 30 para extradós y 30 para
intradós
' Definimos el entorno de trabajo
Dim bodies1 As Bodies
Dim body1 As Body
Dim sketches1 As Sketches
Dim originElements1 As OriginElements
Dim reference1 As Reference
Dim sketch1 As Sketch
' Establecimiento del entorno de trabajo
Set bodies1 = part1.Bodies
Set body1 = bodies1.Item("PartBody") ' Se activa el body1
como body de trabajo
Set sketches1 = body1.Sketches ' Se establece el sketch
Set originElements1 = part1.OriginElements ' Sistema de
referencia 3D
Set reference1 = originElements1.PlaneYZ ' Se establece el
plano YZ como plano de trabajo
Set sketch1 = sketches1.Add(reference1) ' Toma la
referencia1 como referencia para trabajar
' Trabajamos en el plano yz
Dim arrayOfVariantOfDouble1(8)
arrayOfVariantOfDouble1(0) = 0# ' Establece el vector
unidad x en el plano X
arrayOfVariantOfDouble1(1) = 0# ' Establece el vector
unidad y en el plano X
arrayOfVariantOfDouble1(2) = 0# ' Establece el vector
unidad z en el plano X
arrayOfVariantOfDouble1(3) = 0# ' Establece el vector
unidad x en el plano Y
arrayOfVariantOfDouble1(4) = 1# ' Establece el vector
unidad y en el plano Y
arrayOfVariantOfDouble1(5) = 0# ' Establece el vector
unidad z en el plano Y
arrayOfVariantOfDouble1(6) = 0# ' Establece el vector
unidad x en el plano Z
arrayOfVariantOfDouble1(7) = 0# ' Establece el vector
unidad y en el plano Z
arrayOfVariantOfDouble1(8) = 1# ' Establece el vector
unidad z en el plano Z
Set sketch1Variant = sketch1
sketch1Variant.SetAbsoluteAxisData
arrayOfVariantOfDouble1 'Sistema absoluto 3D para el sketch
part1.InWorkObject = sketch1
' Establecimiento del conjunto de herramientas 2D y asignarlo
al sketch de trabajo
Dim factory2D1 As Factory2D
Set factory2D1 = sketch1.OpenEdition()
' Se establecen los elementos geométricos
Dim geometricElements1 As GeometricElements
Set geometricElements1 = sketch1.GeometricElements
' Se define el sistema de ejes dentro del sketch
Dim axis2D1 As Axis2D
Set axis2D1 = geometricElements1.Item("AbsoluteAxis")
' Establecimiento de las direcciones horizontal y vertical
Dim line2D1 As Line2D
Set line2D1 = axis2D1.GetItem("HDirection")
line2D1.ReportName = 1
Dim line2D2 As Line2D
Set line2D2 = axis2D1.GetItem("VDirection")
line2D2.ReportName = 2
' Definimos los puntos con unos valores distribuidos
uniformemente. Estos valores
' no son los definitivos ya que estos seran modificados con la
rule.
Dim point2D1 As ControlPoint2D
Set point2D1 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D1.ReportName = 2 + 1
Dim point2D2 As ControlPoint2D
Set point2D2 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D2.ReportName = 2 + 2
Dim point2D3 As ControlPoint2D
Set point2D3 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D3.ReportName = 2 + 3
Dim point2D4 As ControlPoint2D

```

```

Set point2D4 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D4.ReportName = 2 + 4
Dim point2D5 As ControlPoint2D
Set point2D5 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D5.ReportName = 2 + 5
Dim point2D6 As ControlPoint2D
Set point2D6 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D6.ReportName = 2 + 6
Dim point2D7 As ControlPoint2D
Set point2D7 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D7.ReportName = 2 + 7
Dim point2D8 As ControlPoint2D
Set point2D8 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D8.ReportName = 2 + 8
Dim point2D9 As ControlPoint2D
Set point2D9 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D9.ReportName = 2 + 9
Dim point2D10 As ControlPoint2D
Set point2D10 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D10.ReportName = 2 + 10
Dim point2D11 As ControlPoint2D
Set point2D11 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D11.ReportName = 2 + 11
Dim point2D12 As ControlPoint2D
Set point2D12 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D12.ReportName = 2 + 12
Dim point2D13 As ControlPoint2D
Set point2D13 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D13.ReportName = 2 + 13
Dim point2D14 As ControlPoint2D
Set point2D14 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D14.ReportName = 2 + 14
Dim point2D15 As ControlPoint2D
Set point2D15 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D15.ReportName = 2 + 15
Dim point2D16 As ControlPoint2D
Set point2D16 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D16.ReportName = 2 + 16
Dim point2D17 As ControlPoint2D
Set point2D17 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D17.ReportName = 2 + 17
Dim point2D18 As ControlPoint2D
Set point2D18 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D18.ReportName = 2 + 18
Dim point2D19 As ControlPoint2D
Set point2D19 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D19.ReportName = 2 + 19
Dim point2D20 As ControlPoint2D
Set point2D20 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D20.ReportName = 2 + 20
Dim point2D21 As ControlPoint2D
Set point2D21 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D21.ReportName = 2 + 21
Dim point2D22 As ControlPoint2D
Set point2D22 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D22.ReportName = 2 + 22
Dim point2D23 As ControlPoint2D
Set point2D23 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D23.ReportName = 2 + 23
Dim point2D24 As ControlPoint2D
Set point2D24 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D24.ReportName = 2 + 24
Dim point2D25 As ControlPoint2D
Set point2D25 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D25.ReportName = 2 + 25
Dim point2D26 As ControlPoint2D
Set point2D26 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D26.ReportName = 2 + 26
Dim point2D27 As ControlPoint2D
Set point2D27 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D27.ReportName = 2 + 27
Dim point2D28 As ControlPoint2D

```

```

Set point2D28 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D28.ReportName = 2 + 28
Dim point2D29 As ControlPoint2D
Set point2D29 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D29.ReportName = 2 + 29
Dim point2D30 As ControlPoint2D
Set point2D30 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D30.ReportName = 2 + 30
Dim point2D31 As ControlPoint2D
Set point2D31 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D31.ReportName = 2 + 31
Dim point2D32 As ControlPoint2D
Set point2D32 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D32.ReportName = 2 + 32
Dim point2D33 As ControlPoint2D
Set point2D33 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D33.ReportName = 2 + 33
Dim point2D34 As ControlPoint2D
Set point2D34 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D34.ReportName = 2 + 34
Dim point2D35 As ControlPoint2D
Set point2D35 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D35.ReportName = 2 + 35
Dim point2D36 As ControlPoint2D
Set point2D36 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D36.ReportName = 2 + 36
Dim point2D37 As ControlPoint2D
Set point2D37 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D37.ReportName = 2 + 37
Dim point2D38 As ControlPoint2D
Set point2D38 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D38.ReportName = 2 + 38
Dim point2D39 As ControlPoint2D
Set point2D39 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D39.ReportName = 2 + 39

```

```

Dim point2D40 As ControlPoint2D
Set point2D40 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D40.ReportName = 2 + 40
Dim point2D41 As ControlPoint2D
Set point2D41 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D41.ReportName = 2 + 41
Dim point2D42 As ControlPoint2D
Set point2D42 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D42.ReportName = 2 + 42
Dim point2D43 As ControlPoint2D
Set point2D43 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D43.ReportName = 2 + 43
Dim point2D44 As ControlPoint2D
Set point2D44 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D44.ReportName = 2 + 44
Dim point2D45 As ControlPoint2D
Set point2D45 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D45.ReportName = 2 + 45
Dim point2D46 As ControlPoint2D
Set point2D46 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D46.ReportName = 2 + 46
Dim point2D47 As ControlPoint2D
Set point2D47 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D47.ReportName = 2 + 47
Dim point2D48 As ControlPoint2D
Set point2D48 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D48.ReportName = 2 + 48
Dim point2D49 As ControlPoint2D
Set point2D49 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D49.ReportName = 2 + 49
Dim point2D50 As ControlPoint2D
Set point2D50 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D50.ReportName = 2 + 50
Dim point2D51 As ControlPoint2D
Set point2D51 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)

```

```

point2D51.ReportName = 2 + 51
Dim point2D52 As ControlPoint2D
Set point2D52 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D52.ReportName = 2 + 52
Dim point2D53 As ControlPoint2D
Set point2D53 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D53.ReportName = 2 + 53
Dim point2D54 As ControlPoint2D
Set point2D54 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D54.ReportName = 2 + 54
Dim point2D55 As ControlPoint2D
Set point2D55 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D55.ReportName = 2 + 55
Dim point2D56 As ControlPoint2D
Set point2D56 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D56.ReportName = 2 + 56
Dim point2D57 As ControlPoint2D
Set point2D57 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D57.ReportName = 2 + 57
Dim point2D58 As ControlPoint2D
Set point2D58 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D58.ReportName = 2 + 58
Dim point2D59 As ControlPoint2D
Set point2D59 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D59.ReportName = 2 + 59
Dim point2D60 As ControlPoint2D
Set point2D60 = factory2D1.CreateControlPoint(99999999#,
CuerdaValor)
point2D60.ReportName = 2 + 60
.....
.....
' Creamos el punto del borde de ataque tanto como control
point
' como punto (pero de construcción) ya que este ultimo
' lo necesitamos para parametrizar las líneas de referencia para
poner la incidencia a nuestro perfil.
.....
.....
Dim point2D61 As ControlPoint2D
Set point2D61 = factory2D1.CreateControlPoint(SemienvergaduraValor,
CuerdaValor)
point2D61.ReportName = 63
Dim point2D62 As Point2D
Set point2D62 = factory2D1.CreatePoint(SemienvergaduraValor,
SemienvergaduraValor)
point2D62.ReportName = 64
point2D62.Construction = False
.....
.....
"          INCIDENCIA          "
.....
.....
' Los perfiles suelen tener una incidencia, ángulo, respecto a la
horizontal, por lo que crearemos unas nuevas
' líneas de referencia que formen un ángulo igual a la
incidencia y posteriormente acotaremos sobre estas líneas.
' Creamos las líneas que utilizaremos como referencia.
Comenzamos por la "vertical".
' En primer lugar acotamos y parametrizamos el punto del
borde de ataque.
Dim constraints1 As constraints
Set constraints1 = sketch1.constraints
Dim reference2 As Reference
Set reference2 = part1.CreateReferenceFromObject(line2D2)
'Referencia al eje vertical
Dim reference3 As Reference
Set reference3 = part1.CreateReferenceFromObject(line2D1)
'Referencia al eje horizontal
Dim reference4 As Reference
Set reference4 = part1.CreateReferenceFromObject(point2D62)
' Punto de
construcción
Dim constraint1 As Constraint
Set constraint1 = constraints1.AddBiEltCst(catCstTypeDistance,
reference2,
reference4)
constraint1.Mode = catCstModeDrivingDimension
Dim length1 As Length
Set length1 = constraint1.Dimension
Dim formula1 As Formula
Set formula1 = relations.CreateFormula("Punto borde de
ataque", "Borde de ataque", length1, "Semienvergadura")
Dim constraint2 As Constraint
Set constraint2 = constraints1.AddBiEltCst(catCstTypeDistance,
reference3,
reference4)
constraint2.Mode = catCstModeDrivingDimension

```

```

Dim length2 As Length
Set length2 = constraint2.Dimension

Dim formula2 As Formula
Set formula2 = relations.CreateFormula("Punto borde de
ataque", "Borde de ataque", length2, "Semienvergadura")
' Se crean los puntos de interes para crear los ejes de
coordenadas rotados.

Dim point2D63 As Point2D
Set point2D63 = factory2D1.CreatePoint(-7000#, -7000#)
point2D63.ReportName = 65

Dim point2D64 As Point2D
Set point2D64 = factory2D1.CreatePoint(7000#, 7000#)
point2D64.ReportName = 66
' Se crea la linea

Dim line2D3 As Line2D
Set line2D3 = factory2D1.CreateLine(-7000#, -7000#, 7000#,
7000#)
line2D3.ReportName = 67
line2D3.Construction = True
line2D3.StartPoint = point2D63
line2D3.EndPoint = point2D64
' Acotamos la linea. En primer lugar que forme un ángulo con
la dirección vertical y en segundo
' lugar que sea coincidente con el punto del borde de ataque.

Dim reference5 As Reference
Set reference5 = part1.CreateReferenceFromObject(line2D3)

Dim constraint3 As Constraint
Set constraint3 = constraints1.AddBiEltCst(catCstTypeAngle,
reference5, reference2)
constraint3.Mode = catCstModeDrivingDimension
constraint3.AngleSector = catCstAngleSector0

Dim angle3 As Angle
Set angle3 = constraint3.Dimension

Dim formula3 As Formula
Set formula3 = relations.CreateFormula("Linea referencia
vertical Incidencia", "Incidencia", angle3, "Incidencia")

Dim reference6 As Reference
Set reference6 = part1.CreateReferenceFromObject(line2D3)
'vertical cambiada

Dim reference7 As Reference
Set
reference7 =
part1.CreateReferenceFromObject(point2D62) 'Punto
construcción borde ataque

Dim constraint4 As Constraint
Set constraint4 = constraints1.AddBiEltCst(catCstTypeOn,
reference6, reference7) 'Constraint de coincidencia
constraint4.Mode = catCstModeDrivingDimension

```

```

' Linea "horizontal de referencia".
Dim point2D65 As Point2D
Set point2D65 = factory2D1.CreatePoint(-7000, 7000)
point2D65.ReportName = 68
Dim point2D66 As Point2D
Set point2D66 = factory2D1.CreatePoint(7000, -7000)
point2D66.ReportName = 69
Dim line2D4 As Line2D
Set line2D4 = factory2D1.CreateLine(-7000, 7000, 7000, -
7000)
line2D4.ReportName = 70
line2D4.Construction = True
line2D4.StartPoint = point2D65
line2D4.EndPoint = point2D66
Dim reference8 As Reference
Set reference8 = part1.CreateReferenceFromObject(line2D4)

Dim constraint5 As Constraint
Set
constraint5 =
constraints1.AddBiEltCst(catCstTypePerpendicularity,
reference8, reference6)
constraint5.Mode = catCstModeDrivingDimension

Dim constraint6 As Constraint
Set
constraint6 =
constraints1.AddBiEltCst(catCstTypeDistance, reference7,
reference8)
constraint6.Mode = catCstModeDrivingDimension

Dim length6 As Length
Set length6 = constraint6.Dimension

Dim formula4 As Formula
Set formula4 = relations.CreateFormula("Linea referencia
horizontal Incidencia", "Referencia horizontal Incidencia",
length6, "Semienvergadura")
.....
.....

' Definimos las constraints de todos los puntos acotando estos
en distancias a ambas direcciones del sketch.

' Debemos seguir una lógica para acotar los puntos ya que esta
afectará a nuestra Rule.
.....

''' En primer lugar vamos a acotar los puntos del Extradós
por su distancia al eje vertical '''
.....

Dim reference9 As Reference
Set reference9 = part1.CreateReferenceFromObject(point2D1)

Dim constraint7 As Constraint
Set
constraint7 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,

```

```

reference9)
constraint7.Mode = catCstModeDrivingDimension
Dim reference10 As Reference
Set          reference10          =
part1.CreateReferenceFromObject(point2D2)
Dim constraint8 As Constraint
Set          constraint8          =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference10)
constraint8.Mode = catCstModeDrivingDimension
Dim reference11 As Reference
Set          reference11          =
part1.CreateReferenceFromObject(point2D3)
Dim constraint9 As Constraint
Set          constraint9          =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference11)
constraint9.Mode = catCstModeDrivingDimension
Dim reference12 As Reference
Set          reference12          =
part1.CreateReferenceFromObject(point2D4)
Dim constraint10 As Constraint
Set          constraint10         =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference12)
constraint10.Mode = catCstModeDrivingDimension
Dim reference13 As Reference
Set          reference13          =
part1.CreateReferenceFromObject(point2D5)
Dim constraint11 As Constraint
Set          constraint11         =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference13)
constraint11.Mode = catCstModeDrivingDimension
Dim reference14 As Reference
Set          reference14          =
part1.CreateReferenceFromObject(point2D6)
Dim constraint12 As Constraint
Set          constraint12         =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference14)
constraint12.Mode = catCstModeDrivingDimension
Dim reference15 As Reference
Set          reference15          =
part1.CreateReferenceFromObject(point2D7)
Dim constraint13 As Constraint
Set          constraint13         =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference15)
constraint13.Mode = catCstModeDrivingDimension
Dim reference16 As Reference
Set          reference16          =
part1.CreateReferenceFromObject(point2D8)
Dim constraint14 As Constraint
Set          constraint14         =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference16)
constraint14.Mode = catCstModeDrivingDimension
Dim reference17 As Reference
Set          reference17          =
part1.CreateReferenceFromObject(point2D9)
Dim constraint15 As Constraint
Set          constraint15         =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference17)
constraint15.Mode = catCstModeDrivingDimension
Dim reference18 As Reference
Set          reference18          =
part1.CreateReferenceFromObject(point2D10)
Dim constraint16 As Constraint
Set          constraint16         =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference18)
constraint16.Mode = catCstModeDrivingDimension
Dim reference19 As Reference
Set          reference19          =
part1.CreateReferenceFromObject(point2D11)
Dim constraint17 As Constraint
Set          constraint17         =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference19)
constraint17.Mode = catCstModeDrivingDimension
Dim reference20 As Reference
Set          reference20          =
part1.CreateReferenceFromObject(point2D12)
Dim constraint18 As Constraint
Set          constraint18         =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference20)
constraint18.Mode = catCstModeDrivingDimension
Dim reference21 As Reference
Set          reference21          =
part1.CreateReferenceFromObject(point2D13)
Dim constraint19 As Constraint
Set          constraint19         =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference21)
constraint19.Mode = catCstModeDrivingDimension
Dim reference22 As Reference
Set          reference22          =

```

```

part1.CreateReferenceFromObject(point2D14)
Dim constraint20 As Constraint

Set constraint20 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference22)
constraint20.Mode = catCstModeDrivingDimension
Dim reference23 As Reference

Set reference23 =
part1.CreateReferenceFromObject(point2D15)
Dim constraint21 As Constraint

Set constraint21 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference23)
constraint21.Mode = catCstModeDrivingDimension
Dim reference24 As Reference

Set reference24 =
part1.CreateReferenceFromObject(point2D16)
Dim constraint22 As Constraint

Set constraint22 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference24)
constraint22.Mode = catCstModeDrivingDimension
Dim reference25 As Reference

Set reference25 =
part1.CreateReferenceFromObject(point2D17)
Dim constraint23 As Constraint

Set constraint23 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference25)
constraint23.Mode = catCstModeDrivingDimension
Dim reference26 As Reference

Set reference26 =
part1.CreateReferenceFromObject(point2D18)
Dim constraint24 As Constraint

Set constraint24 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference26)
constraint24.Mode = catCstModeDrivingDimension
Dim reference27 As Reference

Set reference27 =
part1.CreateReferenceFromObject(point2D19)
Dim constraint25 As Constraint

Set constraint25 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference27)
constraint25.Mode = catCstModeDrivingDimension
Dim reference28 As Reference

Set reference28 =
part1.CreateReferenceFromObject(point2D20)
Dim constraint26 As Constraint

Set constraint26 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference28)
constraint26.Mode = catCstModeDrivingDimension
Dim reference29 As Reference

Set reference29 =
part1.CreateReferenceFromObject(point2D21)
Dim constraint27 As Constraint

Set constraint27 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference29)
constraint27.Mode = catCstModeDrivingDimension
Dim reference30 As Reference

Set reference30 =
part1.CreateReferenceFromObject(point2D22)
Dim constraint28 As Constraint

Set constraint28 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference30)
constraint28.Mode = catCstModeDrivingDimension
Dim reference31 As Reference

Set reference31 =
part1.CreateReferenceFromObject(point2D23)
Dim constraint29 As Constraint

Set constraint29 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference31)
constraint29.Mode = catCstModeDrivingDimension
Dim reference32 As Reference

Set reference32 =
part1.CreateReferenceFromObject(point2D24)
Dim constraint30 As Constraint

Set constraint30 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference32)
constraint30.Mode = catCstModeDrivingDimension
Dim reference33 As Reference

Set reference33 =
part1.CreateReferenceFromObject(point2D25)
Dim constraint31 As Constraint

Set constraint31 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference33)
constraint31.Mode = catCstModeDrivingDimension
Dim reference34 As Reference

Set reference34 =
part1.CreateReferenceFromObject(point2D26)
Dim constraint32 As Constraint

Set constraint32 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference34)

```



```

reference34)
constraint32.Mode = catCstModeDrivingDimension
Dim reference35 As Reference
Set reference35 = part1.CreateReferenceFromObject(point2D27)
Dim constraint33 As Constraint
Set constraint33 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference35)
constraint33.Mode = catCstModeDrivingDimension
Dim reference36 As Reference
Set reference36 = part1.CreateReferenceFromObject(point2D28)
Dim constraint34 As Constraint
Set constraint34 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference36)
constraint34.Mode = catCstModeDrivingDimension
Dim reference37 As Reference
Set reference37 = part1.CreateReferenceFromObject(point2D29)
Dim constraint35 As Constraint
Set constraint35 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference37)
constraint35.Mode = catCstModeDrivingDimension
Dim reference38 As Reference
Set reference38 = part1.CreateReferenceFromObject(point2D30)
Dim constraint36 As Constraint
Set constraint36 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference38)
constraint36.Mode = catCstModeDrivingDimension
.....
'En segundo lugar vamos a acotar los puntos del Extradós por
su distancia al eje horizontal
.....
Dim reference39 As Reference
Set reference39 = part1.CreateReferenceFromObject(point2D1)
Dim constraint37 As Constraint
Set constraint37 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference39)
constraint37.Mode = catCstModeDrivingDimension
Dim reference40 As Reference
Set reference40 = part1.CreateReferenceFromObject(point2D2)
Dim constraint38 As Constraint
Set constraint38 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference40)
constraint38.Mode = catCstModeDrivingDimension
Dim reference41 As Reference
Set reference41 = part1.CreateReferenceFromObject(point2D3)
Dim constraint39 As Constraint
Set constraint39 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference41)
constraint39.Mode = catCstModeDrivingDimension
Dim reference42 As Reference
Set reference42 = part1.CreateReferenceFromObject(point2D4)
Dim constraint40 As Constraint
Set constraint40 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference42)
constraint40.Mode = catCstModeDrivingDimension
Dim reference43 As Reference
Set reference43 = part1.CreateReferenceFromObject(point2D5)
Dim constraint41 As Constraint
Set constraint41 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference43)
constraint41.Mode = catCstModeDrivingDimension
Dim reference44 As Reference
Set reference44 = part1.CreateReferenceFromObject(point2D6)
Dim constraint42 As Constraint
Set constraint42 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference44)
constraint42.Mode = catCstModeDrivingDimension
Dim reference45 As Reference
Set reference45 = part1.CreateReferenceFromObject(point2D7)
Dim constraint43 As Constraint
Set constraint43 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference45)
constraint43.Mode = catCstModeDrivingDimension
Dim reference46 As Reference
Set reference46 = part1.CreateReferenceFromObject(point2D8)
Dim constraint44 As Constraint
Set constraint44 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference46)

```

constraint44.Mode = catCstModeDrivingDimension		Set	reference53	=	part1.CreateReferenceFromObject(point2D15)
Dim reference47 As Reference					Dim constraint51 As Constraint
Set	reference47	=			Set
part1.CreateReferenceFromObject(point2D9)			constraint51	=	constraints1.AddBiEltCst(catCstTypeDistance, reference8,
Dim constraint45 As Constraint			reference53)		constraint51.Mode = catCstModeDrivingDimension
Set	constraint45	=			Dim reference54 As Reference
constraints1.AddBiEltCst(catCstTypeDistance, reference8,	reference47)				Set
constraint45.Mode = catCstModeDrivingDimension			reference54	=	part1.CreateReferenceFromObject(point2D16)
Dim reference48 As Reference					Dim constraint52 As Constraint
Set	reference48	=			Set
part1.CreateReferenceFromObject(point2D10)			constraint52	=	constraints1.AddBiEltCst(catCstTypeDistance, reference8,
Dim constraint46 As Constraint			reference54)		constraint52.Mode = catCstModeDrivingDimension
Set	constraint46	=			Dim reference55 As Reference
constraints1.AddBiEltCst(catCstTypeDistance, reference8,	reference48)				Set
constraint46.Mode = catCstModeDrivingDimension			reference55	=	part1.CreateReferenceFromObject(point2D17)
Dim reference49 As Reference					Dim constraint53 As Constraint
Set	reference49	=			Set
part1.CreateReferenceFromObject(point2D11)			constraint53	=	constraints1.AddBiEltCst(catCstTypeDistance, reference8,
Dim constraint47 As Constraint			reference55)		constraint53.Mode = catCstModeDrivingDimension
Set	constraint47	=			Dim reference56 As Reference
constraints1.AddBiEltCst(catCstTypeDistance, reference8,	reference49)				Set
constraint47.Mode = catCstModeDrivingDimension			reference56	=	part1.CreateReferenceFromObject(point2D18)
Dim reference50 As Reference					Dim constraint54 As Constraint
Set	reference50	=			Set
part1.CreateReferenceFromObject(point2D12)			constraint54	=	constraints1.AddBiEltCst(catCstTypeDistance, reference8,
Dim constraint48 As Constraint			reference56)		constraint54.Mode = catCstModeDrivingDimension
Set	constraint48	=			Dim reference57 As Reference
constraints1.AddBiEltCst(catCstTypeDistance, reference8,	reference50)				Set
constraint48.Mode = catCstModeDrivingDimension			reference57	=	part1.CreateReferenceFromObject(point2D19)
Dim reference51 As Reference					Dim constraint55 As Constraint
Set	reference51	=			Set
part1.CreateReferenceFromObject(point2D13)			constraint55	=	constraints1.AddBiEltCst(catCstTypeDistance, reference8,
Dim constraint49 As Constraint			reference57)		constraint55.Mode = catCstModeDrivingDimension
Set	constraint49	=			Dim reference58 As Reference
constraints1.AddBiEltCst(catCstTypeDistance, reference8,	reference51)				Set
constraint49.Mode = catCstModeDrivingDimension			reference58	=	part1.CreateReferenceFromObject(point2D20)
Dim reference52 As Reference					Dim constraint56 As Constraint
Set	reference52	=			Set
part1.CreateReferenceFromObject(point2D14)			constraint56	=	constraints1.AddBiEltCst(catCstTypeDistance, reference8,
Dim constraint50 As Constraint			reference58)		constraint56.Mode = catCstModeDrivingDimension
Set	constraint50	=			Dim reference59 As Reference
constraints1.AddBiEltCst(catCstTypeDistance, reference8,	reference52)				Set
constraint50.Mode = catCstModeDrivingDimension			reference59	=	part1.CreateReferenceFromObject(point2D21)
Dim reference53 As Reference					

```

Dim constraint57 As Constraint
Set constraint57 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference59)
constraint57.Mode = catCstModeDrivingDimension
Dim reference60 As Reference
Set reference60 = part1.CreateReferenceFromObject(point2D22)
Dim constraint58 As Constraint
Set constraint58 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference60)
constraint58.Mode = catCstModeDrivingDimension
Dim reference61 As Reference
Set reference61 = part1.CreateReferenceFromObject(point2D23)
Dim constraint59 As Constraint
Set constraint59 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference61)
constraint59.Mode = catCstModeDrivingDimension
Dim reference62 As Reference
Set reference62 = part1.CreateReferenceFromObject(point2D24)
Dim constraint60 As Constraint
Set constraint60 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference62)
constraint60.Mode = catCstModeDrivingDimension
Dim reference63 As Reference
Set reference63 = part1.CreateReferenceFromObject(point2D25)
Dim constraint61 As Constraint
Set constraint61 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference63)
constraint61.Mode = catCstModeDrivingDimension
Dim reference64 As Reference
Set reference64 = part1.CreateReferenceFromObject(point2D26)
Dim constraint62 As Constraint
Set constraint62 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference64)
constraint62.Mode = catCstModeDrivingDimension
Dim reference65 As Reference
Set reference65 = part1.CreateReferenceFromObject(point2D27)
Dim constraint63 As Constraint
Set constraint63 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference65)
constraint63.Mode = catCstModeDrivingDimension
Dim reference66 As Reference
Set reference66 = part1.CreateReferenceFromObject(point2D28)
Dim constraint64 As Constraint
Set constraint64 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference66)
constraint64.Mode = catCstModeDrivingDimension
Dim reference67 As Reference
Set reference67 = part1.CreateReferenceFromObject(point2D29)
Dim constraint65 As Constraint
Set constraint65 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference67)
constraint65.Mode = catCstModeDrivingDimension
Dim reference68 As Reference
Set reference68 = part1.CreateReferenceFromObject(point2D30)
Dim constraint66 As Constraint
Set constraint66 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference68)
constraint66.Mode = catCstModeDrivingDimension
.....
'En tercer lugar vamos a acotar los puntos del Intradós por su distancia al eje vertical
.....
Dim reference69 As Reference
Set reference69 = part1.CreateReferenceFromObject(point2D31)
Dim constraint67 As Constraint
Set constraint67 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference69)
constraint67.Mode = catCstModeDrivingDimension
Dim reference70 As Reference
Set reference70 = part1.CreateReferenceFromObject(point2D32)
Dim constraint68 As Constraint
Set constraint68 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference70)
constraint68.Mode = catCstModeDrivingDimension
Dim reference71 As Reference
Set reference71 = part1.CreateReferenceFromObject(point2D33)

```

Dim constraint69 As Constraint
Set constraint69 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference71)
constraint69.Mode = catCstModeDrivingDimension
Dim reference72 As Reference
Set reference72 =
part1.CreateReferenceFromObject(point2D34)
Dim constraint70 As Constraint
Set constraint70 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference72)
constraint70.Mode = catCstModeDrivingDimension
Dim reference73 As Reference
Set reference73 =
part1.CreateReferenceFromObject(point2D35)
Dim constraint71 As Constraint
Set constraint71 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference73)
constraint71.Mode = catCstModeDrivingDimension
Dim reference74 As Reference
Set reference74 =
part1.CreateReferenceFromObject(point2D36)
Dim constraint72 As Constraint
Set constraint72 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference74)
constraint72.Mode = catCstModeDrivingDimension
Dim reference75 As Reference
Set reference75 =
part1.CreateReferenceFromObject(point2D37)
Dim constraint73 As Constraint
Set constraint73 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference75)
constraint73.Mode = catCstModeDrivingDimension
Dim reference76 As Reference
Set reference76 =
part1.CreateReferenceFromObject(point2D38)
Dim constraint74 As Constraint
Set constraint74 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference76)
constraint74.Mode = catCstModeDrivingDimension
Dim reference77 As Reference
Set reference77 =
part1.CreateReferenceFromObject(point2D39)
Dim constraint75 As Constraint
Set constraint75 =

constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference77)
constraint75.Mode = catCstModeDrivingDimension
Dim reference78 As Reference
Set reference78 =
part1.CreateReferenceFromObject(point2D40)
Dim constraint76 As Constraint
Set constraint76 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference78)
constraint76.Mode = catCstModeDrivingDimension
Dim reference79 As Reference
Set reference79 =
part1.CreateReferenceFromObject(point2D41)
Dim constraint77 As Constraint
Set constraint77 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference79)
constraint77.Mode = catCstModeDrivingDimension
Dim reference80 As Reference
Set reference80 =
part1.CreateReferenceFromObject(point2D42)
Dim constraint78 As Constraint
Set constraint78 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference80)
constraint78.Mode = catCstModeDrivingDimension
Dim reference81 As Reference
Set reference81 =
part1.CreateReferenceFromObject(point2D43)
Dim constraint79 As Constraint
Set constraint79 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference81)
constraint79.Mode = catCstModeDrivingDimension
Dim reference82 As Reference
Set reference82 =
part1.CreateReferenceFromObject(point2D44)
Dim constraint80 As Constraint
Set constraint80 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference82)
constraint80.Mode = catCstModeDrivingDimension
Dim reference83 As Reference
Set reference83 =
part1.CreateReferenceFromObject(point2D45)
Dim constraint81 As Constraint
Set constraint81 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference83)

```

constraint81.Mode = catCstModeDrivingDimension
Dim reference84 As Reference
Set reference84 = part1.CreateReferenceFromObject(point2D46)
Dim constraint82 As Constraint
Set constraint82 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference84)
constraint82.Mode = catCstModeDrivingDimension
Dim reference85 As Reference
Set reference85 = part1.CreateReferenceFromObject(point2D47)
Dim constraint83 As Constraint
Set constraint83 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference85)
constraint83.Mode = catCstModeDrivingDimension
Dim reference86 As Reference
Set reference86 = part1.CreateReferenceFromObject(point2D48)
Dim constraint84 As Constraint
Set constraint84 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference86)
constraint84.Mode = catCstModeDrivingDimension
Dim reference87 As Reference
Set reference87 = part1.CreateReferenceFromObject(point2D49)
Dim constraint85 As Constraint
Set constraint85 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference87)
constraint85.Mode = catCstModeDrivingDimension
Dim reference88 As Reference
Set reference88 = part1.CreateReferenceFromObject(point2D50)
Dim constraint86 As Constraint
Set constraint86 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference88)
constraint86.Mode = catCstModeDrivingDimension
Dim reference89 As Reference
Set reference89 = part1.CreateReferenceFromObject(point2D51)
Dim constraint87 As Constraint
Set constraint87 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference89)
constraint87.Mode = catCstModeDrivingDimension
Dim reference90 As Reference
Set reference90 = part1.CreateReferenceFromObject(point2D52)
Dim constraint88 As Constraint
Set constraint88 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference90)
constraint88.Mode = catCstModeDrivingDimension
Dim reference91 As Reference
Set reference91 = part1.CreateReferenceFromObject(point2D53)
Dim constraint89 As Constraint
Set constraint89 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference91)
constraint89.Mode = catCstModeDrivingDimension
Dim reference92 As Reference
Set reference92 = part1.CreateReferenceFromObject(point2D54)
Dim constraint90 As Constraint
Set constraint90 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference92)
constraint90.Mode = catCstModeDrivingDimension
Dim reference93 As Reference
Set reference93 = part1.CreateReferenceFromObject(point2D55)
Dim constraint91 As Constraint
Set constraint91 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference93)
constraint91.Mode = catCstModeDrivingDimension
Dim reference94 As Reference
Set reference94 = part1.CreateReferenceFromObject(point2D56)
Dim constraint92 As Constraint
Set constraint92 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference94)
constraint92.Mode = catCstModeDrivingDimension
Dim reference95 As Reference
Set reference95 = part1.CreateReferenceFromObject(point2D57)
Dim constraint93 As Constraint
Set constraint93 = constraints1.AddBiEltCst(catCstTypeDistance, reference6, reference95)
constraint93.Mode = catCstModeDrivingDimension
Dim reference96 As Reference
Set reference96 = part1.CreateReferenceFromObject(point2D58)

```

```

Dim constraint94 As Constraint
Set constraint94 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference96)
constraint94.Mode = catCstModeDrivingDimension
Dim reference97 As Reference
Set reference97 =
part1.CreateReferenceFromObject(point2D59)
Dim constraint95 As Constraint
Set constraint95 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference97)
constraint95.Mode = catCstModeDrivingDimension
Dim reference98 As Reference
Set reference98 =
part1.CreateReferenceFromObject(point2D60)
Dim constraint96 As Constraint
Set constraint96 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference98)
constraint96.Mode = catCstModeDrivingDimension
.....
" Por último acotamos los puntos del Intradós por su
distancia al eje horizontal "
.....
Dim reference99 As Reference
Set reference99 =
part1.CreateReferenceFromObject(point2D31)
Dim constraint97 As Constraint
Set constraint97 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference99)
constraint97.Mode = catCstModeDrivingDimension
Dim reference100 As Reference
Set reference100 =
part1.CreateReferenceFromObject(point2D32)
Dim constraint98 As Constraint
Set constraint98 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference100)
constraint98.Mode = catCstModeDrivingDimension
Dim reference101 As Reference
Set reference101 =
part1.CreateReferenceFromObject(point2D33)
Dim constraint99 As Constraint
Set constraint99 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference101)
constraint99.Mode = catCstModeDrivingDimension
Dim reference102 As Reference
Set reference102 =
part1.CreateReferenceFromObject(point2D34)
Dim constraint100 As Constraint
Set constraint100 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference102)
constraint100.Mode = catCstModeDrivingDimension
Dim reference103 As Reference
Set reference103 =
part1.CreateReferenceFromObject(point2D35)
Dim constraint101 As Constraint
Set constraint101 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference103)
constraint101.Mode = catCstModeDrivingDimension
Dim reference104 As Reference
Set reference104 =
part1.CreateReferenceFromObject(point2D36)
Dim constraint102 As Constraint
Set constraint102 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference104)
constraint102.Mode = catCstModeDrivingDimension
Dim reference105 As Reference
Set reference105 =
part1.CreateReferenceFromObject(point2D37)
Dim constraint103 As Constraint
Set constraint103 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference105)
constraint103.Mode = catCstModeDrivingDimension
Dim reference106 As Reference
Set reference106 =
part1.CreateReferenceFromObject(point2D38)
Dim constraint104 As Constraint
Set constraint104 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference106)
constraint104.Mode = catCstModeDrivingDimension
Dim reference107 As Reference
Set reference107 =
part1.CreateReferenceFromObject(point2D39)
Dim constraint105 As Constraint
Set constraint105 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference107)
constraint105.Mode = catCstModeDrivingDimension
Dim reference108 As Reference
Set reference108 =
part1.CreateReferenceFromObject(point2D40)

```

```

Dim constraint106 As Constraint
Set constraint106 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference108)
constraint106.Mode = catCstModeDrivingDimension
Dim reference109 As Reference
Set reference109 = part1.CreateReferenceFromObject(point2D41)
Dim constraint107 As Constraint
Set constraint107 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference109)
constraint107.Mode = catCstModeDrivingDimension
Dim reference110 As Reference
Set reference110 = part1.CreateReferenceFromObject(point2D42)
Dim constraint108 As Constraint
Set constraint108 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference110)
constraint108.Mode = catCstModeDrivingDimension
Dim reference111 As Reference
Set reference111 = part1.CreateReferenceFromObject(point2D43)
Dim constraint109 As Constraint
Set constraint109 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference111)
constraint109.Mode = catCstModeDrivingDimension
Dim reference112 As Reference
Set reference112 = part1.CreateReferenceFromObject(point2D44)
Dim constraint110 As Constraint
Set constraint110 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference112)
constraint110.Mode = catCstModeDrivingDimension
Dim reference113 As Reference
Set reference113 = part1.CreateReferenceFromObject(point2D45)
Dim constraint111 As Constraint
Set constraint111 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference113)
constraint111.Mode = catCstModeDrivingDimension
Dim reference114 As Reference
Set reference114 = part1.CreateReferenceFromObject(point2D46)
Dim constraint112 As Constraint
Set constraint112 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference114)
constraint112.Mode = catCstModeDrivingDimension
Dim reference115 As Reference
Set reference115 = part1.CreateReferenceFromObject(point2D47)
Dim constraint113 As Constraint
Set constraint113 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference115)
constraint113.Mode = catCstModeDrivingDimension
Dim reference116 As Reference
Set reference116 = part1.CreateReferenceFromObject(point2D48)
Dim constraint114 As Constraint
Set constraint114 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference116)
constraint114.Mode = catCstModeDrivingDimension
Dim reference117 As Reference
Set reference117 = part1.CreateReferenceFromObject(point2D49)
Dim constraint115 As Constraint
Set constraint115 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference117)
constraint115.Mode = catCstModeDrivingDimension
Dim reference118 As Reference
Set reference118 = part1.CreateReferenceFromObject(point2D50)
Dim constraint116 As Constraint
Set constraint116 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference118)
constraint116.Mode = catCstModeDrivingDimension
Dim reference119 As Reference
Set reference119 = part1.CreateReferenceFromObject(point2D51)
Dim constraint117 As Constraint
Set constraint117 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference119)
constraint117.Mode = catCstModeDrivingDimension
Dim reference120 As Reference
Set reference120 = part1.CreateReferenceFromObject(point2D52)
Dim constraint118 As Constraint
Set constraint118 = constraints1.AddBiEltCst(catCstTypeDistance, reference8, reference120)
constraint118.Mode = catCstModeDrivingDimension

```

```

constraint118.Mode = catCstModeDrivingDimension
Dim reference121 As Reference
Set reference121 =
part1.CreateReferenceFromObject(point2D53)
Dim constraint119 As Constraint
Set constraint119 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference121)
constraint119.Mode = catCstModeDrivingDimension
Dim reference122 As Reference
Set reference122 =
part1.CreateReferenceFromObject(point2D54)
Dim constraint120 As Constraint
Set constraint120 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference122)
constraint120.Mode = catCstModeDrivingDimension
Dim reference123 As Reference
Set reference123 =
part1.CreateReferenceFromObject(point2D55)
Dim constraint121 As Constraint
Set constraint121 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference123)
constraint121.Mode = catCstModeDrivingDimension
Dim reference124 As Reference
Set reference124 =
part1.CreateReferenceFromObject(point2D56)
Dim constraint122 As Constraint
Set constraint122 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference124)
constraint122.Mode = catCstModeDrivingDimension
Dim reference125 As Reference
Set reference125 =
part1.CreateReferenceFromObject(point2D57)
Dim constraint123 As Constraint
Set constraint123 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference125)
constraint123.Mode = catCstModeDrivingDimension
Dim reference126 As Reference
Set reference126 =
part1.CreateReferenceFromObject(point2D58)
Dim constraint124 As Constraint
Set constraint124 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference126)
constraint124.Mode = catCstModeDrivingDimension
Dim reference127 As Reference
Set reference127 =
part1.CreateReferenceFromObject(point2D59)
Dim constraint125 As Constraint
Set constraint125 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference127)
constraint125.Mode = catCstModeDrivingDimension
Dim reference128 As Reference
Set reference128 =
part1.CreateReferenceFromObject(point2D60)
Dim constraint126 As Constraint
Set constraint126 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference128)
constraint126.Mode = catCstModeDrivingDimension
" Acotamos el control point del borde de ataque "
Dim reference129 As Reference
Set reference129 =
part1.CreateReferenceFromObject(point2D61)
Dim constraint127 As Constraint
Set constraint127 =
constraints1.AddBiEltCst(catCstTypeDistance, reference6,
reference129)
constraint127.Mode = catCstModeDrivingDimension
Dim length127 As Length
Set length127 = constraint127.Dimension
length127.Value = 0
Dim reference130 As Reference
Set reference130 =
part1.CreateReferenceFromObject(point2D61)
Dim constraint128 As Constraint
Set constraint128 =
constraints1.AddBiEltCst(catCstTypeDistance, reference8,
reference130)
constraint128.Mode = catCstModeDrivingDimension
Dim length128 As Length
Set length128 = constraint128.Dimension
Dim formula5 As Formula
Set formula5 = relations.CreateFormula("Punto borde de
ataque encastre", "Borde de ataque", length128,
"Semienvergadura")
' Una vez creado y acotados todos los puntos del perfil,
realizamos un spline alrededor de todos los puntos en un nuevo
' sketch. En dicho spline hemos unido todos los puntos
exceptuando los situados en el borde de salida. Ya que si
unimos
' estos dos puntos en el mismo spline este se distorsiona.

```



```

Dim arrayOfObject1(60)
Set arrayOfObject1(0) = point2D30
Set arrayOfObject1(1) = point2D29
Set arrayOfObject1(2) = point2D28
Set arrayOfObject1(3) = point2D27
Set arrayOfObject1(4) = point2D26
Set arrayOfObject1(5) = point2D25
Set arrayOfObject1(6) = point2D24
Set arrayOfObject1(7) = point2D23
Set arrayOfObject1(8) = point2D22
Set arrayOfObject1(9) = point2D21
Set arrayOfObject1(10) = point2D20
Set arrayOfObject1(11) = point2D19
Set arrayOfObject1(12) = point2D18
Set arrayOfObject1(13) = point2D17
Set arrayOfObject1(14) = point2D16
Set arrayOfObject1(15) = point2D15
Set arrayOfObject1(16) = point2D14
Set arrayOfObject1(17) = point2D13
Set arrayOfObject1(18) = point2D12
Set arrayOfObject1(19) = point2D11
Set arrayOfObject1(20) = point2D10
Set arrayOfObject1(21) = point2D9
Set arrayOfObject1(22) = point2D8
Set arrayOfObject1(23) = point2D7
Set arrayOfObject1(24) = point2D6
Set arrayOfObject1(25) = point2D5
Set arrayOfObject1(26) = point2D4
Set arrayOfObject1(27) = point2D3
Set arrayOfObject1(28) = point2D2
Set arrayOfObject1(29) = point2D1
Set arrayOfObject1(30) = point2D61
Set arrayOfObject1(31) = point2D31
Set arrayOfObject1(32) = point2D32
Set arrayOfObject1(33) = point2D33
Set arrayOfObject1(34) = point2D34
Set arrayOfObject1(35) = point2D35
Set arrayOfObject1(36) = point2D36
Set arrayOfObject1(37) = point2D37
Set arrayOfObject1(38) = point2D38
Set arrayOfObject1(39) = point2D39
Set arrayOfObject1(40) = point2D40
Set arrayOfObject1(41) = point2D41

Set arrayOfObject1(42) = point2D42
Set arrayOfObject1(43) = point2D43
Set arrayOfObject1(44) = point2D44
Set arrayOfObject1(45) = point2D45
Set arrayOfObject1(46) = point2D46
Set arrayOfObject1(47) = point2D47
Set arrayOfObject1(48) = point2D48
Set arrayOfObject1(49) = point2D49
Set arrayOfObject1(50) = point2D50
Set arrayOfObject1(51) = point2D51
Set arrayOfObject1(52) = point2D52
Set arrayOfObject1(53) = point2D53
Set arrayOfObject1(54) = point2D54
Set arrayOfObject1(55) = point2D55
Set arrayOfObject1(56) = point2D56
Set arrayOfObject1(57) = point2D57
Set arrayOfObject1(58) = point2D58
Set arrayOfObject1(59) = point2D59
Set arrayOfObject1(60) = point2D60

Dim spline2D1 As Spline2D
Set factory2D1temp = factory2D1
Set spline2D1 = factory2D1temp.CreateSpline(arrayOfObject1)
spline2D1.ReportName = 71
' Unimos los dos puntos del borde de salida mediante otro spline el cual creará una línea recta entre estos dos puntos '
' aunque lo ideal sería unirlos mediante un arco de circunferencia el cual aun no sabemos hacer. '

Dim arrayOfObject2(1)
Set arrayOfObject2(0) = point2D30
Set arrayOfObject2(1) = point2D60
Dim spline2D2 As Spline2D
Set factory2D2temp = factory2D1
Set spline2D2 = factory2D2temp.CreateSpline(arrayOfObject2)
spline2D2.ReportName = 72
' Cerramos el sketch
sketch1.CloseEdition
' Creamos las Regla a través de las cuales tendremos parametrizados nuestros perfiles.
' Vamos a crear una regla para punto del perfil.
' Posteriormente modificamos la Regla e introducimos las líneas de códigos obtenidas con MATLAB.

```

' Al ser líneas de códigos muy largas, estas no caben en una sola línea, por lo cual habrá que poner al final de cada línea

' " & _ para poder continuar debajo y cuando queramos que cambie de línea & vbCrLf & _

Dim rule1 As Rule

Set rule1 = relations.CreateProgram("rule1", "", "")

' Modificamos la Regla e introducimos todas las líneas de códigos obtenidas con MATLAB

' Al ser muchas líneas de códigos estas no caben en una sola línea, por lo cual habrá que poner al final de cada línea

' " & _ para poder continuar debajo y cuando queramos que cambie de línea & vbCrLf & _

rule1.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (20/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.7\Offset = (20/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(20/1000))))" else
PartBody\Sketch.1\Offset.7\Offset = (20/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(20/1000))))" & vbCrLf & _
```

```
"if (20/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.37\Offset =
(Semienvergadura+((NACA1/100)*(20/1000*Cuerda)*(2*(N
ACA2/10)-
20/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2
969*sqrt(20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(20/1000))))" else
PartBody\Sketch.1\Offset.37\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
20/1000*Cuerda)*(-2*(NACA2/10)+20/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
20/1000)-0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(20/1000))))" & vbCrLf & _
```

Dim rule2 As Rule

Set rule2 = relations.CreateProgram("rule2", "", "")

rule2.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (40/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.8\Offset = (40/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(40/1000)-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(40/1000))))" else
PartBody\Sketch.1\Offset.8\Offset = (40/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(40/1000)-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
```

```
0.1015*(40/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(40/1000))))" & vbCrLf & _
```

```
"if (40/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.38\Offset =
(Semienvergadura+((NACA1/100)*(40/1000*Cuerda)*(2*(N
ACA2/10)-
40/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2
969*sqrt(40/1000)-0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(40/1000))))" else
PartBody\Sketch.1\Offset.38\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
40/1000*Cuerda)*(-2*(NACA2/10)+40/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
40/1000)-0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(40/1000))))" & vbCrLf & _
```

Dim rule3 As Rule

Set rule3 = relations.CreateProgram("rule3", "", "")

rule3.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (60/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.9\Offset = (60/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(60/1000))))" else
PartBody\Sketch.1\Offset.9\Offset = (60/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(60/1000))))" & vbCrLf & _
```

```
"if (60/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.39\Offset =
(Semienvergadura+((NACA1/100)*(60/1000*Cuerda)*(2*(N
ACA2/10)-
60/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2
969*sqrt(60/1000)-0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(60/1000))))" else
PartBody\Sketch.1\Offset.39\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
60/1000*Cuerda)*(-2*(NACA2/10)+60/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
60/1000)-0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(60/1000))))" & vbCrLf & _
```

Dim rule4 As Rule

Set rule4 = relations.CreateProgram("rule4", "", "")

rule4.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (80/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.10\Offset = (80/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(80/1000)-
```

```
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(80/1000))))))" else
PartBody\Sketch.1\Offset.10\Offset = (80/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(80/1000))))))" & vbCrLf &
-
```

```
"if (80/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.40\Offset =
(Semienvergadura+((NACA1/100)*(80/1000*Cuerda)*(2*(N
ACA2/10)-
80/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2
969*sqrt(80/1000)-0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(80/1000))))))" else
PartBody\Sketch.1\Offset.40\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
80/1000*Cuerda)*(-2*(NACA2/10)+80/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
80/1000)-0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(80/1000))))))"
```

Dim rule5 As Rule

```
Set rule5 = relations.CreateProgram("rule5", "", "")
```

```
rule5.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (100/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.11\Offset = (100/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(100/1000))))))" else
PartBody\Sketch.1\Offset.11\Offset = (100/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(100/1000))))))" & vbCrLf
& _
```

```
"if (100/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.41\Offset =
(Semienvergadura+((NACA1/100)*(100/1000*Cuerda)*(2*(
NACA2/10)-
100/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(100/1000)-0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(100/1000))))))" else
PartBody\Sketch.1\Offset.41\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
100/1000*Cuerda)*(-2*(NACA2/10)+100/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
100/1000)-0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(100/1000))))))"
```

Dim rule6 As Rule

```
Set rule6 = relations.CreateProgram("rule6", "", "")
```

```
rule6.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (136/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.12\Offset = (136/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(136/1000))))))" else
PartBody\Sketch.1\Offset.12\Offset = (136/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(136/1000))))))" & vbCrLf
& _
```

```
"if (136/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.42\Offset =
(Semienvergadura+((NACA1/100)*(136/1000*Cuerda)*(2*(
NACA2/10)-
136/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(136/1000)-0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(136/1000))))))" else
PartBody\Sketch.1\Offset.42\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
136/1000*Cuerda)*(-2*(NACA2/10)+136/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
136/1000)-0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(136/1000))))))"
```

Dim rule7 As Rule

```
Set rule7 = relations.CreateProgram("rule7", "", "")
```

```
rule7.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (172/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.13\Offset = (172/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(172/1000))))))" else
PartBody\Sketch.1\Offset.13\Offset = (172/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(172/1000))))))" & vbCrLf
& _
```

```
"if (172/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.43\Offset =
(Semienvergadura+((NACA1/100)*(172/1000*Cuerda)*(2*(
NACA2/10)-
172/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(172/1000)-0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(172/1000))))))" else
PartBody\Sketch.1\Offset.43\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
172/1000*Cuerda)*(-2*(NACA2/10)+172/1000+1)/(1-
```

```
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
172/1000)-0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(172/1000))))))"
```

Dim rule8 As Rule

```
Set rule8 = relations.CreateProgram("rule8", "", "")
```

```
rule8.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (208/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.14\Offset = (208/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(208/1000))))" else
PartBody\Sketch.1\Offset.14\Offset = (208/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(208/1000))))" & vbCrLf
& _
```

```
"if (208/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.44\Offset =
(Semienvergadura+((NACA1/100)*(208/1000*Cuerda)*(2*(
NACA2/10)-
208/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(208/1000)-0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(208/1000))))" else
PartBody\Sketch.1\Offset.44\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
208/1000*Cuerda)*(-2*(NACA2/10)+208/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
208/1000)-0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(208/1000))))" & vbCrLf
& _
```

Dim rule9 As Rule

```
Set rule9 = relations.CreateProgram("rule9", "", "")
```

```
rule9.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (244/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.15\Offset = (244/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(244/1000))))" else
PartBody\Sketch.1\Offset.15\Offset = (244/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(244/1000))))" & vbCrLf
& _
```

```
"if (244/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.45\Offset =
(Semienvergadura+((NACA1/100)*(244/1000*Cuerda)*(2*(
NACA2/10)-
```

```
244/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(244/1000)-0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(244/1000))))" else
PartBody\Sketch.1\Offset.45\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
244/1000*Cuerda)*(-2*(NACA2/10)+244/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
244/1000)-0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(244/1000))))" & vbCrLf
& _
```

Dim rule10 As Rule

```
Set rule10 = relations.CreateProgram("rule10", "", "")
```

```
rule10.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (280/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.16\Offset = (280/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(280/1000))))" else
PartBody\Sketch.1\Offset.16\Offset = (280/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(280/1000))))" & vbCrLf
& _
```

```
"if (280/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.46\Offset =
(Semienvergadura+((NACA1/100)*(280/1000*Cuerda)*(2*(
NACA2/10)-
280/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(280/1000)-0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(280/1000))))" else
PartBody\Sketch.1\Offset.46\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
280/1000*Cuerda)*(-2*(NACA2/10)+280/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
280/1000)-0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(280/1000))))" & vbCrLf
& _
```

Dim rule11 As Rule

```
Set rule11 = relations.CreateProgram("rule11", "", "")
```

```
rule11.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (316/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.17\Offset = (316/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(316/1000))))" else
PartBody\Sketch.1\Offset.17\Offset = (316/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
```

```
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(316/1000))))" & vbCrLf
& _
```

```
"if (316/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.47\Offset =
(Semienvergadura+(NACA1/100)*(316/1000*Cuerda)*(2*(
NACA2/10)-
316/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(316/1000)-0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(316/1000))))" else
PartBody\Sketch.1\Offset.47\Offset =
(Semienvergadura+(NACA1/100)*(Cuerda-
316/1000*Cuerda)*(-2*(NACA2/10)+316/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
316/1000)-0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(316/1000))))"
```

Dim rule12 As Rule

Set rule12 = relations.CreateProgram("rule12", "", "")

```
rule12.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (352/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.18\Offset = (352/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(352/1000))))" else
PartBody\Sketch.1\Offset.18\Offset = (352/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(352/1000))))" & vbCrLf
& _
```

```
"if (352/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.48\Offset =
(Semienvergadura+(NACA1/100)*(352/1000*Cuerda)*(2*(
NACA2/10)-
352/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(352/1000)-0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(352/1000))))" else
PartBody\Sketch.1\Offset.48\Offset =
(Semienvergadura+(NACA1/100)*(Cuerda-
352/1000*Cuerda)*(-2*(NACA2/10)+352/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
352/1000)-0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(352/1000))))"
```

Dim rule13 As Rule

Set rule13 = relations.CreateProgram("rule13", "", "")

```
rule13.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (388/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.19\Offset = (388/1000*Cuerda -
```

```
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(388/1000))))" else
PartBody\Sketch.1\Offset.19\Offset = (388/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(388/1000))))" & vbCrLf
& _
```

```
"if (388/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.49\Offset =
(Semienvergadura+(NACA1/100)*(388/1000*Cuerda)*(2*(
NACA2/10)-
388/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(388/1000)-0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(388/1000))))" else
PartBody\Sketch.1\Offset.49\Offset =
(Semienvergadura+(NACA1/100)*(Cuerda-
388/1000*Cuerda)*(-2*(NACA2/10)+388/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
388/1000)-0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(388/1000))))"
```

Dim rule14 As Rule

Set rule14 = relations.CreateProgram("rule14", "", "")

```
rule14.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (424/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.20\Offset = (424/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(424/1000))))" else
PartBody\Sketch.1\Offset.20\Offset = (424/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(424/1000))))" & vbCrLf
& _
```

```
"if (424/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.50\Offset =
(Semienvergadura+(NACA1/100)*(424/1000*Cuerda)*(2*(
NACA2/10)-
424/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(424/1000)-0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(424/1000))))" else
PartBody\Sketch.1\Offset.50\Offset =
(Semienvergadura+(NACA1/100)*(Cuerda-
424/1000*Cuerda)*(-2*(NACA2/10)+424/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
424/1000)-0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(424/1000))))"
```

Dim rule15 As Rule

Set rule15 = relations.CreateProgram("rule15", "", "")

rule15.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (460/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.21\Offset = (460/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(460/1000)-
0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(460/1000)))) else
PartBody\Sketch.1\Offset.21\Offset = (460/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(460/1000)-
0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(460/1000))))" & vbCrLf
& _
```

```
"if (460/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.51\Offset =
(Semienvergadura+((NACA1/100)*(460/1000*Cuerda)*(2*(
NACA2/10)-
460/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(460/1000)-0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(460/1000)))) else
PartBody\Sketch.1\Offset.51\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
460/1000*Cuerda)*(-2*(NACA2/10)+460/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
460/1000)-0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(460/1000))))"
```

Dim rule16 As Rule

Set rule16 = relations.CreateProgram("rule16", "", "")

rule16.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (496/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.22\Offset = (496/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(496/1000)-
0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(496/1000)))) else
PartBody\Sketch.1\Offset.22\Offset = (496/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(496/1000)-
0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(496/1000))))" & vbCrLf
& _
```

```
"if (496/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.52\Offset =
(Semienvergadura+((NACA1/100)*(496/1000*Cuerda)*(2*(
NACA2/10)-
496/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(496/1000)-0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(496/1000)))) else
```

```
PartBody\Sketch.1\Offset.52\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
496/1000*Cuerda)*(-2*(NACA2/10)+496/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
496/1000)-0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(496/1000))))"
```

Dim rule17 As Rule

Set rule17 = relations.CreateProgram("rule17", "", "")

rule17.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (532/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.23\Offset = (532/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(532/1000)))) else
PartBody\Sketch.1\Offset.23\Offset = (532/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(532/1000))))" & vbCrLf
& _
```

```
"if (532/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.53\Offset =
(Semienvergadura+((NACA1/100)*(532/1000*Cuerda)*(2*(
NACA2/10)-
532/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(532/1000)-0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(532/1000)))) else
PartBody\Sketch.1\Offset.53\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
532/1000*Cuerda)*(-2*(NACA2/10)+532/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
532/1000)-0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(532/1000))))"
```

Dim rule18 As Rule

Set rule18 = relations.CreateProgram("rule18", "", "")

rule18.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (568/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.24\Offset = (568/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(568/1000)))) else
PartBody\Sketch.1\Offset.24\Offset = (568/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(568/1000))))" & vbCrLf
& _
```

```
"if (568/1000*Cuerda) <= (Cuerda*(NACA2/10))
```



```
"if (712/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.28\Offset = (712/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(712/1000)))) else
PartBody\Sketch.1\Offset.28\Offset = (712/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(712/1000))))" & vbCrLf
& _
```

```
"if (712/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.58\Offset =
(Semienvergadura+((NACA1/100)*(712/1000*Cuerda)*(2*(
NACA2/10)-
712/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(712/1000)-0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(712/1000)))) else
PartBody\Sketch.1\Offset.58\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
712/1000*Cuerda)*(-2*(NACA2/10)+712/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
712/1000)-0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(712/1000))))"
```

Dim rule23 As Rule

Set rule23 = relations.CreateProgram("rule23", "", "")

rule23.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (748/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.29\Offset = (748/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(748/1000)))) else
PartBody\Sketch.1\Offset.29\Offset = (748/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(748/1000))))" & vbCrLf
& _
```

```
"if (748/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.59\Offset =
(Semienvergadura+((NACA1/100)*(748/1000*Cuerda)*(2*(
NACA2/10)-
748/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(748/1000)-0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(748/1000)))) else
PartBody\Sketch.1\Offset.59\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
748/1000*Cuerda)*(-2*(NACA2/10)+748/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
748/1000)-0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
```

```
0.1015*(748/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(748/1000))))"
```

Dim rule24 As Rule

Set rule24 = relations.CreateProgram("rule24", "", "")

rule24.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (784/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.30\Offset = (784/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(784/1000)))) else
PartBody\Sketch.1\Offset.30\Offset = (784/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(784/1000))))" & vbCrLf
& _
```

```
"if (784/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.60\Offset =
(Semienvergadura+((NACA1/100)*(784/1000*Cuerda)*(2*(
NACA2/10)-
784/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(784/1000)-0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(784/1000)))) else
PartBody\Sketch.1\Offset.60\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
784/1000*Cuerda)*(-2*(NACA2/10)+784/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
784/1000)-0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(784/1000))))"
```

Dim rule25 As Rule

Set rule25 = relations.CreateProgram("rule25", "", "")

rule25.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (820/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.31\Offset = (820/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(820/1000)))) else
PartBody\Sketch.1\Offset.31\Offset = (820/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(820/1000))))" & vbCrLf
& _
```

```
"if (820/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.61\Offset =
(Semienvergadura+((NACA1/100)*(820/1000*Cuerda)*(2*(
NACA2/10)-
820/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(820/1000)-0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
```



```
0.1015*(820/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(820/1000))))
PartBody\Sketch.1\Offset.61\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-820/1000*Cuerda)*(-2*(NACA2/10)+820/1000+1)/(1-NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(820/1000)-0.1260*(820/1000)-0.3516*(820/1000)**2+0.2843*(820/1000)**3-0.1015*(820/1000)**4*cos(atan(2*(NACA1/100)/((1-(NACA2/10)**2)*((NACA2/10)-(820/1000))))))"
```

Dim rule26 As Rule

Set rule26 = relations.CreateProgram("rule26", "", "")

rule26.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (856/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.32\Offset = (856/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(856/1000))))
else
PartBody\Sketch.1\Offset.32\Offset = (856/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*sin(atan(2*(NACA1/100)/((1-(NACA2/10)**2)*((NACA2/10)-(856/1000)))))" & vbCrLf & _
```

```
"if (856/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.62\Offset =
(Semienvergadura+((NACA1/100)*(856/1000*Cuerda)*(2*(NACA2/10)-856/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(856/1000)-0.1260*(856/1000)-0.3516*(856/1000)**2+0.2843*(856/1000)**3-0.1015*(856/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(856/1000))))
else
PartBody\Sketch.1\Offset.62\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-856/1000*Cuerda)*(-2*(NACA2/10)+856/1000+1)/(1-NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(856/1000)-0.1260*(856/1000)-0.3516*(856/1000)**2+0.2843*(856/1000)**3-0.1015*(856/1000)**4)*cos(atan(2*(NACA1/100)/((1-(NACA2/10)**2)*((NACA2/10)-(856/1000)))))"
```

Dim rule27 As Rule

Set rule27 = relations.CreateProgram("rule27", "", "")

rule27.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (892/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.33\Offset = (892/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(892/1000))))
else
PartBody\Sketch.1\Offset.33\Offset = (892/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4)*sin(atan(2*(NACA1/100)/((1-(NACA2/10)**2)*((NACA2/10)-(892/1000))))" & vbCrLf & _
```

```
"if (892/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.63\Offset =
(Semienvergadura+((NACA1/100)*(892/1000*Cuerda)*(2*(NACA2/10)-892/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(892/1000)-0.1260*(892/1000)-0.3516*(892/1000)**2+0.2843*(892/1000)**3-0.1015*(892/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(892/1000))))
else
PartBody\Sketch.1\Offset.63\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-892/1000*Cuerda)*(-2*(NACA2/10)+892/1000+1)/(1-NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(892/1000)-0.1260*(892/1000)-0.3516*(892/1000)**2+0.2843*(892/1000)**3-0.1015*(892/1000)**4)*cos(atan(2*(NACA1/100)/((1-(NACA2/10)**2)*((NACA2/10)-(892/1000))))"
```

Dim rule28 As Rule

Set rule28 = relations.CreateProgram("rule28", "", "")

rule28.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (928/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.34\Offset = (928/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(928/1000))))
else
PartBody\Sketch.1\Offset.34\Offset = (928/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4)*sin(atan(2*(NACA1/100)/((1-(NACA2/10)**2)*((NACA2/10)-(928/1000))))" & vbCrLf & _
```

```
"if (928/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.64\Offset =
(Semienvergadura+((NACA1/100)*(928/1000*Cuerda)*(2*(NACA2/10)-928/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(928/1000)-0.1260*(928/1000)-0.3516*(928/1000)**2+0.2843*(928/1000)**3-0.1015*(928/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(928/1000))))
else
PartBody\Sketch.1\Offset.64\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-928/1000*Cuerda)*(-2*(NACA2/10)+928/1000+1)/(1-NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(928/1000)-0.1260*(928/1000)-0.3516*(928/1000)**2+0.2843*(928/1000)**3-0.1015*(928/1000)**4)*cos(atan(2*(NACA1/100)/((1-(NACA2/10)**2)*((NACA2/10)-(928/1000))))"
```

Dim rule29 As Rule

Set rule29 = relations.CreateProgram("rule29", "", "")

rule29.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (964/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.35\Offset = (964/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/10)**2)*((NACA2/10)-(964/1000))))
else
```

```
PartBody\Sketch.1\Offset.35\Offset = (964/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(964/1000))))" & vbCrLf
& _
```

```
"if (964/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.65\Offset =
(Semienvergadura+((NACA1/100)*(964/1000*Cuerda)*(2*(
NACA2/10)-
964/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.
2969*sqrt(964/1000)-0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(964/1000))))" else
PartBody\Sketch.1\Offset.65\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
964/1000*Cuerda)*(-2*(NACA2/10)+964/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
964/1000)-0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(964/1000))))"
```

Dim rule30 As Rule

Set rule30 = relations.CreateProgram("rule30", "", "")

```
rule30.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (1000/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.36\Offset = (1000/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2
/10)**2)*((NACA2/10)-(1000/1000))))" else
PartBody\Sketch.1\Offset.36\Offset = (1000/1000*Cuerda -
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(1000/1000))))" & vbCrLf
& _
```

```
"if (1000/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.66\Offset =
(Semienvergadura+((NACA1/100)*(1000/1000*Cuerda)*(2*(
NACA2/10)-
1000/1000)/(NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(
0.2969*sqrt(1000/1000)-0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*cos(atan(2*(NACA1/100)/((NACA
2/10)**2)*((NACA2/10)-(1000/1000))))" else
PartBody\Sketch.1\Offset.66\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
1000/1000*Cuerda)*(-2*(NACA2/10)+1000/1000+1)/(1-
NACA2/10)**2)+NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(
1000/1000)-0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(1000/1000))))"
```

Dim rule31 As Rule

Set rule31 = relations.CreateProgram("rule31", "", "")

```
rule31.Modify "/*Rule created by Francisco Javier Jiménez
```

```
Calvo*/" & vbCrLf & _
```

```
"if (20/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.67\Offset = (20/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(20/1000))))" else
PartBody\Sketch.1\Offset.67\Offset = (20/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(20/1000))))" & vbCrLf &
_
```

```
"if (20/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.97\Offset =
(Semienvergadura+((NACA1/100)*(20/1000*Cuerda)*(2*(N
ACA2/10)-20/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(20/1000))))" else
PartBody\Sketch.1\Offset.97\Offset = (Semienvergadura
+((NACA1/100)*(Cuerda-20/1000*Cuerda)*(-
2*(NACA2/10)+20/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(20/1000))))"
```

Dim rule32 As Rule

Set rule32 = relations.CreateProgram("rule32", "", "")

```
rule32.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (40/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.68\Offset = (40/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(40/1000)-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(40/1000))))" else
PartBody\Sketch.1\Offset.68\Offset = (40/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(40/1000)-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(40/1000))))" & vbCrLf &
_
```

```
"if (40/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.98\Offset =
(Semienvergadura+((NACA1/100)*(40/1000*Cuerda)*(2*(N
ACA2/10)-40/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(40/1000)-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/1
0)**2)*((NACA2/10)-(40/1000))))" else
PartBody\Sketch.1\Offset.98\Offset = (Semienvergadura
+((NACA1/100)*(Cuerda-40/1000*Cuerda)*(-
2*(NACA2/10)+40/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(40/1000)-
0.1260*(40/1000)-
```

```
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(40/1000))))"
```

Dim rule33 As Rule

```
Set rule33 = relations.CreateProgram("rule33", "", "")
```

```
rule33.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (60/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.69\Offset = (60/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/1
0)**2)*(NACA2/10)-(60/1000)))) else
PartBody\Sketch.1\Offset.69\Offset = (60/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(60/1000))))" & vbCrLf & _
```

```
"if (60/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.99\Offset =
(Semienvergadura+((NACA1/100)*(60/1000*Cuerda)*(2*(N
ACA2/10)-60/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/1
0)**2)*(NACA2/10)-(60/1000)))) else
PartBody\Sketch.1\Offset.99\Offset = (Semienvergadura
+((NACA1/100)*(Cuerda-60/1000*Cuerda)*(-
2*(NACA2/10)+60/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(60/1000))))"
```

Dim rule34 As Rule

```
Set rule34 = relations.CreateProgram("rule34", "", "")
```

```
rule34.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (80/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.70\Offset = (80/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/1
0)**2)*(NACA2/10)-(80/1000)))) else
PartBody\Sketch.1\Offset.70\Offset = (80/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(80/1000))))" & vbCrLf & _
```

```
"if (80/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.100\Offset =
(Semienvergadura+((NACA1/100)*(80/1000*Cuerda)*(2*(N
ACA2/10)-80/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
```

```
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/1
0)**2)*(NACA2/10)-(80/1000)))) else
PartBody\Sketch.1\Offset.100\Offset = (Semienvergadura
+((NACA1/100)*(Cuerda-80/1000*Cuerda)*(-
2*(NACA2/10)+80/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(80/1000))))"
```

Dim rule35 As Rule

```
Set rule35 = relations.CreateProgram("rule35", "", "")
```

```
rule35.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (100/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.71\Offset = (100/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(100/1000)))) else
PartBody\Sketch.1\Offset.71\Offset = (100/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(100/1000))))" & vbCrLf & _
```

```
"if (100/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.101\Offset =
(Semienvergadura+((NACA1/100)*(100/1000*Cuerda)*(2*(
NACA2/10)-100/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(100/1000)))) else
PartBody\Sketch.1\Offset.101\Offset = (Semienvergadura
+((NACA1/100)*(Cuerda-100/1000*Cuerda)*(-
2*(NACA2/10)+100/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(100/1000))))"
```

Dim rule36 As Rule

```
Set rule36 = relations.CreateProgram("rule36", "", "")
```

```
rule36.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (136/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.72\Offset = (136/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(136/1000)))) else
PartBody\Sketch.1\Offset.72\Offset = (136/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(136/1000))))" & vbCrLf & _
```

```
& _
"if (136/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.102\Offset =
(Semienvergadura+((NACA1/100)*(136/1000*Cuerda)*(2*(
NACA2/10)-136/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(136/1000)))))) else
PartBody\Sketch.1\Offset.102\Offset = (Semienvergadura
+((NACA1/100)*(Cuerda-136/1000*Cuerda)*(-
2*(NACA2/10)+136/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(136/1000))))))"
```

Dim rule37 As Rule

Set rule37 = relations.CreateProgram("rule37", "", "")

rule37.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (172/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.73\Offset = (172/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(172/1000)))))) else
PartBody\Sketch.1\Offset.73\Offset = (172/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(172/1000))))))" & vbCrLf
& _
```

```
"if (172/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.103\Offset =
(Semienvergadura+((NACA1/100)*(172/1000*Cuerda)*(2*(
NACA2/10)-172/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(172/1000)))))) else
PartBody\Sketch.1\Offset.103\Offset = (Semienvergadura
+((NACA1/100)*(Cuerda-172/1000*Cuerda)*(-
2*(NACA2/10)+172/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(172/1000))))))"
```

Dim rule38 As Rule

Set rule38 = relations.CreateProgram("rule38", "", "")

rule38.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (208/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.74\Offset = (208/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
```

```
0.1015*(208/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(208/1000)))))) else
PartBody\Sketch.1\Offset.74\Offset = (208/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(208/1000))))))" & vbCrLf
& _
```

```
"if (208/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.104\Offset =
(Semienvergadura+((NACA1/100)*(208/1000*Cuerda)*(2*(
NACA2/10)-208/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(208/1000)))))) else
PartBody\Sketch.1\Offset.104\Offset = (Semienvergadura
+((NACA1/100)*(Cuerda-208/1000*Cuerda)*(-
2*(NACA2/10)+208/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(208/1000))))))"
```

Dim rule39 As Rule

Set rule39 = relations.CreateProgram("rule39", "", "")

rule39.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (244/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.75\Offset = (244/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(244/1000)))))) else
PartBody\Sketch.1\Offset.75\Offset = (244/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(244/1000))))))" & vbCrLf
& _
```

```
"if (244/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.105\Offset =
(Semienvergadura+((NACA1/100)*(244/1000*Cuerda)*(2*(
NACA2/10)-244/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(244/1000)))))) else
PartBody\Sketch.1\Offset.105\Offset = (Semienvergadura
+((NACA1/100)*(Cuerda-244/1000*Cuerda)*(-
2*(NACA2/10)+244/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(244/1000))))))"
```

Dim rule40 As Rule

Set rule40 = relations.CreateProgram("rule40", "", "")

rule40.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (280/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.76\Offset = (280/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(280/1000))))
else
PartBody\Sketch.1\Offset.76\Offset = (280/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(280/1000))))" & vbCrLf
& _
```

```
"if (280/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.106\Offset =
(Semienvergadura+((NACA1/100)*(280/1000*Cuerda)*(2*(
NACA2/10)-280/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(280/1000))))
else
PartBody\Sketch.1\Offset.106\Offset = (Semienvergadura
+((NACA1/100)*(Cuerda-280/1000*Cuerda)*(-
2*(NACA2/10)+280/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(280/1000))))" & vbCrLf
& _
```

Dim rule41 As Rule

Set rule41 = relations.CreateProgram("rule41", "", "")

rule41.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (316/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.77\Offset = (316/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(316/1000))))
else
PartBody\Sketch.1\Offset.77\Offset = (316/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(316/1000))))" & vbCrLf
& _
```

```
"if (316/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.107\Offset =
(Semienvergadura+((NACA1/100)*(316/1000*Cuerda)*(2*(
NACA2/10)-316/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(316/1000))))
else
PartBody\Sketch.1\Offset.107\Offset = (Semienvergadura
+((NACA1/100)*(Cuerda-316/1000*Cuerda)*(-
2*(NACA2/10)+316/1000+1)/(1-NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(316/1000))))" & vbCrLf
& _
```

```
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(316/1000))))" & vbCrLf
& _
```

Dim rule42 As Rule

Set rule42 = relations.CreateProgram("rule42", "", "")

rule42.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (352/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.78\Offset = (352/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(352/1000))))
else
PartBody\Sketch.1\Offset.78\Offset = (352/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(352/1000))))" & vbCrLf
& _
```

```
"if (352/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.108\Offset =
(Semienvergadura+((NACA1/100)*(352/1000*Cuerda)*(2*(
NACA2/10)-352/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(352/1000))))
else
PartBody\Sketch.1\Offset.108\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
352/1000*Cuerda)*(-2*(NACA2/10)+352/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(352/1000))))" & vbCrLf
& _
```

Dim rule43 As Rule

Set rule43 = relations.CreateProgram("rule43", "", "")

rule43.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (388/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.79\Offset = (388/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(388/1000))))
else
PartBody\Sketch.1\Offset.79\Offset = (388/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(388/1000))))" & vbCrLf
& _
```

```
"if (388/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.109\Offset =
(Semienvergadura+((NACA1/100)*(388/1000*Cuerda)*(2*(
NACA2/10)-388/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(388/1000))))" & vbCrLf
& _
```

```

NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(388/1000))))    else
PartBody\Sketch.1\Offset.109\Offset    =
(Semienvergadura+((NACA1/100)*(Cuerda-
388/1000*Cuerda)*(-2*(NACA2/10)+388/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(388/1000))))"

```

Dim rule44 As Rule

Set rule44 = relations.CreateProgram("rule44", "", "")

rule44.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```

"if (424/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.80\Offset = (424/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(424/1000))))    else
PartBody\Sketch.1\Offset.80\Offset = (424/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(424/1000))))" & vbCrLf
& _

```

```

"if (424/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.110\Offset    =
(Semienvergadura+((NACA1/100)*(424/1000*Cuerda)*(2*(
NACA2/10)-424/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(424/1000))))    else
PartBody\Sketch.1\Offset.110\Offset    =
(Semienvergadura+((NACA1/100)*(Cuerda-
424/1000*Cuerda)*(-2*(NACA2/10)+424/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(424/1000))))"

```

Dim rule45 As Rule

Set rule45 = relations.CreateProgram("rule45", "", "")

rule45.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```

"if (460/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.81\Offset = (460/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(460/1000)-
0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(460/1000))))    else
PartBody\Sketch.1\Offset.81\Offset = (460/1000*Cuerda +

```

```

NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(460/1000)-
0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(460/1000))))" & vbCrLf
& _

```

```

"if (460/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.111\Offset    =
(Semienvergadura+((NACA1/100)*(460/1000*Cuerda)*(2*(
NACA2/10)-460/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(460/1000)-
0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(460/1000))))    else
PartBody\Sketch.1\Offset.111\Offset    =
(Semienvergadura+((NACA1/100)*(Cuerda-
460/1000*Cuerda)*(-2*(NACA2/10)+460/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(460/1000)-
0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(460/1000))))"

```

Dim rule46 As Rule

Set rule46 = relations.CreateProgram("rule46", "", "")

rule46.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```

"if (496/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.82\Offset = (496/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(496/1000)-
0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(496/1000))))    else
PartBody\Sketch.1\Offset.82\Offset = (496/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(496/1000)-
0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(496/1000))))" & vbCrLf
& _

```

```

"if (496/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.112\Offset    =
(Semienvergadura+((NACA1/100)*(496/1000*Cuerda)*(2*(
NACA2/10)-496/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(496/1000)-
0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(496/1000))))    else
PartBody\Sketch.1\Offset.112\Offset    =
(Semienvergadura+((NACA1/100)*(Cuerda-
496/1000*Cuerda)*(-2*(NACA2/10)+496/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(496/1000)-
0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(496/1000))))"

```

Dim rule47 As Rule

Set rule47 = relations.CreateProgram("rule47", "", "")

rule47.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (532/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.83\Offset = (532/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(532/1000)))) else
PartBody\Sketch.1\Offset.83\Offset = (532/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(532/1000))))" & vbCrLf
& _
```

```
"if (532/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.113\Offset =
(Semienvergadura+((NACA1/100)*(532/1000*Cuerda)*(2*(
NACA2/10)-532/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(532/1000)))) else
PartBody\Sketch.1\Offset.113\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
532/1000*Cuerda)*(-2*(NACA2/10)+532/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(532/1000))))"
```

Dim rule48 As Rule

Set rule48 = relations.CreateProgram("rule48", "", "")

rule48.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (568/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.84\Offset = (568/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(568/1000)))) else
PartBody\Sketch.1\Offset.84\Offset = (568/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(568/1000))))" & vbCrLf
& _
```

```
"if (568/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.114\Offset =
(Semienvergadura+((NACA1/100)*(568/1000*Cuerda)*(2*(
NACA2/10)-568/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(568/1000)))) else
PartBody\Sketch.1\Offset.114\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
568/1000*Cuerda)*(-2*(NACA2/10)+568/1000+1)/(1-
```

```
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(568/1000))))"
```

Dim rule49 As Rule

Set rule49 = relations.CreateProgram("rule49", "", "")

rule49.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (604/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.85\Offset = (604/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(604/1000)))) else
PartBody\Sketch.1\Offset.85\Offset = (604/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(604/1000))))" & vbCrLf
& _
```

```
"if (604/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.115\Offset =
(Semienvergadura+((NACA1/100)*(604/1000*Cuerda)*(2*(
NACA2/10)-604/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(604/1000)))) else
PartBody\Sketch.1\Offset.115\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
604/1000*Cuerda)*(-2*(NACA2/10)+604/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(604/1000))))"
```

Dim rule50 As Rule

Set rule50 = relations.CreateProgram("rule50", "", "")

rule50.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (640/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.86\Offset = (640/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*(NACA2/10)-(640/1000)))) else
PartBody\Sketch.1\Offset.86\Offset = (640/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*(NACA2/10)-(640/1000))))" & vbCrLf
& _
```

```
"if (640/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.116\Offset =
```

```
(Semienvergadura+((NACA1/100)*(640/1000*Cuerda)*(2*(
NACA2/10)-640/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(640/1000))))))" else
PartBody\Sketch.1\Offset.116\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
640/1000*Cuerda)*(-2*(NACA2/10)+640/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(640/1000))))))"
```

Dim rule51 As Rule

Set rule51 = relations.CreateProgram("rule51", "", "")

rule51.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (676/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.87\Offset = (676/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(676/1000)-
0.1260*(676/1000)-
0.3516*(676/1000)**2+0.2843*(676/1000)**3-
0.1015*(676/1000)**4*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(676/1000))))" else
PartBody\Sketch.1\Offset.87\Offset = (676/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(676/1000)-
0.1260*(676/1000)-
0.3516*(676/1000)**2+0.2843*(676/1000)**3-
0.1015*(676/1000)**4*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(676/1000))))" & vbCrLf
& _
```

```
"if (676/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.117\Offset =
(Semienvergadura+((NACA1/100)*(676/1000*Cuerda)*(2*(
NACA2/10)-676/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(676/1000)-
0.1260*(676/1000)-
0.3516*(676/1000)**2+0.2843*(676/1000)**3-
0.1015*(676/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(676/1000))))" else
PartBody\Sketch.1\Offset.117\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
676/1000*Cuerda)*(-2*(NACA2/10)+676/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(676/1000)-
0.1260*(676/1000)-
0.3516*(676/1000)**2+0.2843*(676/1000)**3-
0.1015*(676/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(676/1000))))"
```

Dim rule52 As Rule

Set rule52 = relations.CreateProgram("rule52", "", "")

rule52.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (712/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.88\Offset = (712/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4*sin(atan(2*(NACA1/100)/((NACA2/
```

```
10)**2)*((NACA2/10)-(712/1000))))" else
PartBody\Sketch.1\Offset.88\Offset = (712/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(712/1000))))" & vbCrLf
& _
```

```
"if (712/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.118\Offset =
(Semienvergadura+((NACA1/100)*(712/1000*Cuerda)*(2*(
NACA2/10)-712/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(712/1000))))" else
PartBody\Sketch.1\Offset.118\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
712/1000*Cuerda)*(-2*(NACA2/10)+712/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(712/1000))))"
```

Dim rule53 As Rule

Set rule53 = relations.CreateProgram("rule53", "", "")

rule53.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (748/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.89\Offset = (748/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(748/1000))))" else
PartBody\Sketch.1\Offset.89\Offset = (748/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(748/1000))))" & vbCrLf
& _
```

```
"if (748/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.119\Offset =
(Semienvergadura+((NACA1/100)*(748/1000*Cuerda)*(2*(
NACA2/10)-748/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(748/1000))))" else
PartBody\Sketch.1\Offset.119\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
748/1000*Cuerda)*(-2*(NACA2/10)+748/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(748/1000))))"
```

Dim rule54 As Rule


```

Set rule54 = relations.CreateProgram("rule54", "", "")
rule54.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _
"if (784/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.90\Offset = (784/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(784/1000))))
else
PartBody\Sketch.1\Offset.90\Offset = (784/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10))**2)*((NACA2/10)-(784/1000))))" & vbCrLf
& _
"if (784/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.120\Offset =
(Semienvergadura+((NACA1/100)*(784/1000*Cuerda)*(2*(
NACA2/10)-784/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(784/1000))))
else
PartBody\Sketch.1\Offset.120\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
784/1000*Cuerda)*(-2*(NACA2/10)+784/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10))**2)*((NACA2/10)-(784/1000))))"
Dim rule55 As Rule
Set rule55 = relations.CreateProgram("rule55", "", "")
rule55.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _
"if (820/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.91\Offset = (820/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(820/1000))))
else
PartBody\Sketch.1\Offset.91\Offset = (820/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10))**2)*((NACA2/10)-(820/1000))))" & vbCrLf
& _
"if (820/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.121\Offset =
(Semienvergadura+((NACA1/100)*(820/1000*Cuerda)*(2*(
NACA2/10)-820/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(820/1000))))
else
PartBody\Sketch.1\Offset.121\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-

```

```

820/1000*Cuerda)*(-2*(NACA2/10)+820/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10))**2)*((NACA2/10)-(820/1000))))"

```

Dim rule56 As Rule

```
Set rule56 = relations.CreateProgram("rule56", "", "")
```

```
rule56.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _
```

```

"if (856/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.92\Offset = (856/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(856/1000))))
else
PartBody\Sketch.1\Offset.92\Offset = (856/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10))**2)*((NACA2/10)-(856/1000))))" & vbCrLf
& _

```

```

"if (856/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.122\Offset =
(Semienvergadura+((NACA1/100)*(856/1000*Cuerda)*(2*(
NACA2/10)-856/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(856/1000))))
else
PartBody\Sketch.1\Offset.122\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
856/1000*Cuerda)*(-2*(NACA2/10)+856/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*cos(atan(2*(NACA1/100)/((1-
(NACA2/10))**2)*((NACA2/10)-(856/1000))))"

```

Dim rule57 As Rule

```
Set rule57 = relations.CreateProgram("rule57", "", "")
```

```
rule57.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _
```

```

"if (892/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.93\Offset = (892/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4)*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(892/1000))))
else
PartBody\Sketch.1\Offset.93\Offset = (892/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4)*sin(atan(2*(NACA1/100)/((1-
(NACA2/10))**2)*((NACA2/10)-(892/1000))))" & vbCrLf
& _

```

```
"if (892/1000*Cuerda) <= (Cuerda*(NACA2/10))
```

```

PartBody\Sketch.1\Offset.123\Offset =
(Semienvergadura+((NACA1/100)*(892/1000*Cuerda)*(2*(
NACA2/10)-892/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(892/1000)))))) else
PartBody\Sketch.1\Offset.123\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
892/1000*Cuerda)*(-2*(NACA2/10)+892/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(892/1000))))))"
Dim rule58 As Rule
Set rule58 = relations.CreateProgram("rule58", "", "")
rule58.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (928/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.94\Offset = (928/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(928/1000)))))) else
PartBody\Sketch.1\Offset.94\Offset = (928/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(928/1000))))))" & vbCrLf
& _
"if (928/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.124\Offset =
(Semienvergadura+((NACA1/100)*(928/1000*Cuerda)*(2*(
NACA2/10)-928/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(928/1000)))))) else
PartBody\Sketch.1\Offset.124\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
928/1000*Cuerda)*(-2*(NACA2/10)+928/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(928/1000))))))"
Dim rule59 As Rule
Set rule59 = relations.CreateProgram("rule59", "", "")
rule59.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (964/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.95\Offset = (964/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4*sin(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(964/1000)))))) else
PartBody\Sketch.1\Offset.95\Offset = (964/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(964/1000))))))" & vbCrLf
& _
"if (964/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.125\Offset =
(Semienvergadura+((NACA1/100)*(964/1000*Cuerda)*(2*(
NACA2/10)-964/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4*cos(atan(2*(NACA1/100)/((NACA2/
10)**2)*((NACA2/10)-(964/1000)))))) else
PartBody\Sketch.1\Offset.125\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
964/1000*Cuerda)*(-2*(NACA2/10)+964/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(964/1000))))))"
Dim rule60 As Rule
Set rule60 = relations.CreateProgram("rule60", "", "")
rule60.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (1000/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.96\Offset = (1000/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4*sin(atan(2*(NACA1/100)/((NACA2
/10)**2)*((NACA2/10)-(1000/1000)))))) else
PartBody\Sketch.1\Offset.96\Offset = (1000/1000*Cuerda +
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4*sin(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(1000/1000))))))" & vbCrLf
& _
"if (1000/1000*Cuerda) <= (Cuerda*(NACA2/10))
PartBody\Sketch.1\Offset.126\Offset =
(Semienvergadura+((NACA1/100)*(1000/1000*Cuerda)*(2*(
NACA2/10)-1000/1000)/(NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4*cos(atan(2*(NACA1/100)/((NACA
2/10)**2)*((NACA2/10)-(1000/1000)))))) else
PartBody\Sketch.1\Offset.126\Offset =
(Semienvergadura+((NACA1/100)*(Cuerda-
1000/1000*Cuerda)*(-2*(NACA2/10)+1000/1000+1)/(1-
NACA2/10)**2)-
NACA34/(100*0.2)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4*cos(atan(2*(NACA1/100)/((1-
(NACA2/10)**2)*((NACA2/10)-(1000/1000))))))"

```



```

Dim point2D70 As ControlPoint2D
Set          point2D70          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D70.ReportName = 6 + 70
Dim point2D71 As ControlPoint2D
Set          point2D71          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D71.ReportName = 6 + 71
Dim point2D72 As ControlPoint2D
Set          point2D72          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D72.ReportName = 6 + 72
Dim point2D73 As ControlPoint2D
Set          point2D73          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D73.ReportName = 6 + 73
Dim point2D74 As ControlPoint2D
Set          point2D74          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D74.ReportName = 6 + 74
Dim point2D75 As ControlPoint2D
Set          point2D75          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D75.ReportName = 6 + 75
Dim point2D76 As ControlPoint2D
Set          point2D76          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D76.ReportName = 6 + 76
Dim point2D77 As ControlPoint2D
Set          point2D77          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D77.ReportName = 6 + 77
Dim point2D78 As ControlPoint2D
Set          point2D78          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D78.ReportName = 6 + 78
Dim point2D79 As ControlPoint2D
Set          point2D79          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D79.ReportName = 6 + 79
Dim point2D80 As ControlPoint2D
Set          point2D80          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D80.ReportName = 6 + 80
Dim point2D81 As ControlPoint2D
Set          point2D81          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D81.ReportName = 6 + 81
Dim point2D82 As ControlPoint2D
Set          point2D82          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D82.ReportName = 6 + 82
Dim point2D83 As ControlPoint2D
Set          point2D83          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D83.ReportName = 6 + 83
Dim point2D84 As ControlPoint2D
Set          point2D84          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D84.ReportName = 6 + 84
Dim point2D85 As ControlPoint2D
Set          point2D85          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D85.ReportName = 6 + 85
Dim point2D86 As ControlPoint2D
Set          point2D86          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D86.ReportName = 6 + 86
Dim point2D87 As ControlPoint2D
Set          point2D87          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D87.ReportName = 6 + 87
Dim point2D88 As ControlPoint2D
Set          point2D88          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D88.ReportName = 6 + 88
Dim point2D89 As ControlPoint2D
Set          point2D89          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D89.ReportName = 6 + 89
Dim point2D90 As ControlPoint2D
Set          point2D90          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D90.ReportName = 6 + 90
Dim point2D91 As ControlPoint2D
Set          point2D91          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D91.ReportName = 6 + 91
Dim point2D92 As ControlPoint2D
Set          point2D92          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D92.ReportName = 6 + 92
Dim point2D93 As ControlPoint2D
Set          point2D93          =

```

```

factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D93.ReportName = 6 + 93
Dim point2D94 As ControlPoint2D
Set point2D94 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D94.ReportName = 6 + 94
Dim point2D95 As ControlPoint2D
Set point2D95 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D95.ReportName = 6 + 95
Dim point2D96 As ControlPoint2D
Set point2D96 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D96.ReportName = 6 + 96
Dim point2D97 As ControlPoint2D
Set point2D97 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D97.ReportName = 6 + 97
Dim point2D98 As ControlPoint2D
Set point2D98 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D98.ReportName = 6 + 98
Dim point2D99 As ControlPoint2D
Set point2D99 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D99.ReportName = 6 + 99
Dim point2D100 As ControlPoint2D
Set point2D100 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D100.ReportName = 6 + 100
Dim point2D101 As ControlPoint2D
Set point2D101 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D101.ReportName = 6 + 101
Dim point2D102 As ControlPoint2D
Set point2D102 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D102.ReportName = 6 + 102
Dim point2D103 As ControlPoint2D
Set point2D103 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D103.ReportName = 6 + 103
Dim point2D104 As ControlPoint2D
Set point2D104 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D104.ReportName = 6 + 104
Dim point2D105 As ControlPoint2D
Set point2D105 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D105.ReportName = 6 + 105
Dim point2D106 As ControlPoint2D
Set point2D106 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D106.ReportName = 6 + 106
Dim point2D107 As ControlPoint2D
Set point2D107 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D107.ReportName = 6 + 107
Dim point2D108 As ControlPoint2D
Set point2D108 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D108.ReportName = 6 + 108
Dim point2D109 As ControlPoint2D
Set point2D109 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D109.ReportName = 6 + 109
Dim point2D110 As ControlPoint2D
Set point2D110 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D110.ReportName = 6 + 110
Dim point2D111 As ControlPoint2D
Set point2D111 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D111.ReportName = 6 + 111
Dim point2D112 As ControlPoint2D
Set point2D112 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D112.ReportName = 6 + 112
Dim point2D113 As ControlPoint2D
Set point2D113 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D113.ReportName = 6 + 113
Dim point2D114 As ControlPoint2D
Set point2D114 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D114.ReportName = 6 + 114
Dim point2D115 As ControlPoint2D
Set point2D115 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D115.ReportName = 6 + 115
Dim point2D116 As ControlPoint2D
Set point2D116 = factory2D2.CreateControlPoint(9999999999#, CuerdaValor)
point2D116.ReportName = 6 + 116
Dim point2D117 As ControlPoint2D

```

```

Set          point2D117          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)

point2D117.ReportName = 6 + 117

Dim point2D118 As ControlPoint2D

Set          point2D118          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)

point2D118.ReportName = 6 + 118

Dim point2D119 As ControlPoint2D

Set          point2D119          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)

point2D119.ReportName = 6 + 119

Dim point2D120 As ControlPoint2D

Set          point2D120          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)

point2D120.ReportName = 6 + 120

Dim point2D121 As ControlPoint2D

Set          point2D121          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)

point2D121.ReportName = 6 + 121

Dim point2D122 As ControlPoint2D

Set          point2D122          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)

point2D122.ReportName = 6 + 122

Dim point2D123 As ControlPoint2D

Set          point2D123          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)

point2D123.ReportName = 6 + 123

Dim point2D124 As ControlPoint2D

Set          point2D124          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)

point2D124.ReportName = 6 + 124

Dim point2D125 As ControlPoint2D

Set          point2D125          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)

point2D125.ReportName = 6 + 125

Dim point2D126 As ControlPoint2D

Set          point2D126          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)

point2D126.ReportName = 6 + 126

' Creamos el punto del borde de ataque tanto como control
point como en point, ya que nos hará falta

' posteriormente para crear la línea que define la extrusión.

Dim point2D127 As ControlPoint2D

Set          point2D127          =
factory2D2.CreateControlPoint(9999999999#, CuerdaValor)

point2D127.ReportName = 133

Dim point2D128 As Point2D

Set point2D128 = factory2D2.CreatePoint(9999999999#,
CuerdaValor)

point2D128.ReportName = 134

point2D128.Construction = False

.....

"          FLECHA Y TORSIÓN          "

.....

' Creamos las líneas de construcción de referencia para
parametrizar la torsión y flecha del ala. Para ello acotamos '
' el punto del borde de ataque (no el control point) y creamos
una línea coincidente con dicho punto y con un ángulo '
' con la vertical igual al ángulo de torsión. Por último creamos
una línea perpendicular a ésta última que pase por '
' el origen de coordenadas y por último acotamos el perfil
sobre estas líneas de referencia.

.....

' Acotamos el punto del borde de ataque para poder crear las
líneas de referencia

' para la torsión.

' Definimos las constraints del punto acotando este en
distancias a ambas direcciones del sketch.

Dim constraints2 As constraints

Set constraints2 = sketch2.constraints

Dim reference133 As Reference

Set          reference133          =
part1.CreateReferenceFromObject(line2D6) 'Referencia al eje
vertical

Dim reference134 As Reference

Set          reference134          =
part1.CreateReferenceFromObject(line2D5) 'Referencia al eje
horizontal

' Acotamos y parametrizamos la Flecha y el diedro.

Dim reference135 As Reference

Set          reference135          =
part1.CreateReferenceFromObject(point2D128)

Dim constraint129 As Constraint

Set          constraint129          =
constraints2.AddBiEltCst(catCstTypeDistance, reference133,
reference135)

constraint129.Mode = catCstModeDrivingDimension

Dim length129 As Length

Set length129 = constraint129.Dimension

Dim formula7 As Formula

Set formula7 = relations.CreateFormula("Punto borde de
ataque punta ala", "Borde de ataque", length129,
"Semienvergadura + Semienvergadura*tan(Flecha)")

```

```

Dim reference136 As Reference
Set reference136 = part1.CreateReferenceFromObject(point2D128)
Dim constraint130 As Constraint
Set constraint130 = constraints2.AddBiEltCst(catCstTypeDistance, reference134, reference135)
constraint130.Mode = catCstModeDrivingDimension
Dim length130 As Length
Set length130 = constraint130.Dimension
Dim formula8 As Formula
Set formula8 = relations.CreateFormula("Punto control borde de ataque raiz", "Borde de ataque", length130, "Semienvergadura + Semienvergadura*tan(Diedro)")
' Creamos las lineas que utilizaremos como referencia. Comenzamos por la "vertical".
' Se crean los puntos de interes.
Dim point2D129 As Point2D
Set point2D129 = factory2D2.CreatePoint(-7000#, -7000#)
point2D129.ReportName = 135
Dim point2D130 As Point2D
Set point2D130 = factory2D2.CreatePoint(7000#, 7000#)
point2D130.ReportName = 136
' Se crea la linea
Dim line2D7 As Line2D
Set line2D7 = factory2D2.CreateLine(-7000#, -7000#, 7000#, 7000#)
line2D7.ReportName = 137
line2D7.Construction = True
line2D7.StartPoint = point2D129
line2D7.EndPoint = point2D130
' Acotamos la linea. En primer lugar que forme un ángulo con la dirección vertical y en segundo
' lugar que sea coincidente con el punto del borde de ataque.
Dim reference137 As Reference
Set reference137 = part1.CreateReferenceFromObject(line2D7)
Dim constraint131 As Constraint
Set constraint131 = constraints2.AddBiEltCst(catCstTypeAngle, reference137, reference133)
constraint131.Mode = catCstModeDrivingDimension
constraint131.AngleSector = catCstAngleSector0
Dim angle131 As Angle
Set angle131 = constraint131.Dimension
Dim formula9 As Formula
Set formula9 = relations.CreateFormula("Linea referencia Torsión", "Torsión", angle131, "Torsion + Incidencia")

```

```

Dim reference138 As Reference
Set reference138 = part1.CreateReferenceFromObject(line2D7)
Dim reference139 As Reference
Set reference139 = part1.CreateReferenceFromObject(point2D128)
Dim constraint132 As Constraint
Set constraint132 = constraints2.AddBiEltCst(catCstTypeOn, reference138, reference139)
constraint132.Mode = catCstModeDrivingDimension
' Linea "horizontal de referencia".
Dim point2D131 As Point2D
Set point2D131 = factory2D2.CreatePoint(-7000, 7000)
point2D131.ReportName = 138
Dim point2D132 As Point2D
Set point2D132 = factory2D2.CreatePoint(7000, -7000)
point2D132.ReportName = 139
Dim line2D8 As Line2D
Set line2D8 = factory2D2.CreateLine(-7000, 7000, 7000, -7000)
line2D8.ReportName = 140
line2D8.Construction = True
line2D8.StartPoint = point2D131
line2D8.EndPoint = point2D132
Dim reference140 As Reference
Set reference140 = part1.CreateReferenceFromObject(line2D8)
Dim constraint133 As Constraint
Set constraint133 = constraints2.AddBiEltCst(catCstTypePerpendicularity, reference140, reference138)
constraint133.Mode = catCstModeDrivingDimension
Dim reference141 As Reference
Set reference141 = part1.CreateReferenceFromObject(point2D128)
Dim constraint134 As Constraint
Set constraint134 = constraints2.AddBiEltCst(catCstTypeDistance, reference140, reference141)
constraint134.Mode = catCstModeDrivingDimension
Dim length134 As Length
Set length134 = constraint134.Dimension
Dim formula10 As Formula
Set formula10 = relations.CreateFormula("Punto control borde de ataque punta ala", "Borde de ataque", length134, "Semienvergadura*tan(Diedro)+Semienvergadura")

```

' Acotamos los puntos del perfil del encastre respecto a las líneas de referencia nuevas.

Dim reference142 As Reference

Set reference142 =
part1.CreateReferenceFromObject(line2D7) ' Vertical

Dim reference143 As Reference

Set reference143 =
part1.CreateReferenceFromObject(line2D8) ' Horizontal

' En primer lugar vamos a acotar los puntos del Extradós por su distancia al eje vertical '

Dim reference144 As Reference

Set reference144 =
part1.CreateReferenceFromObject(point2D67)

Dim constraint135 As Constraint

Set constraint135 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference144)

constraint135.Mode = catCstModeDrivingDimension

Dim reference145 As Reference

Set reference145 =
part1.CreateReferenceFromObject(point2D68)

Dim constraint136 As Constraint

Set constraint136 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference145)

constraint136.Mode = catCstModeDrivingDimension

Dim reference146 As Reference

Set reference146 =
part1.CreateReferenceFromObject(point2D69)

Dim constraint137 As Constraint

Set constraint137 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference146)

constraint137.Mode = catCstModeDrivingDimension

Dim reference147 As Reference

Set reference147 =
part1.CreateReferenceFromObject(point2D70)

Dim constraint138 As Constraint

Set constraint138 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference147)

constraint138.Mode = catCstModeDrivingDimension

Dim reference148 As Reference

Set reference148 =
part1.CreateReferenceFromObject(point2D71)

Dim constraint139 As Constraint

Set constraint139 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,

reference148)

constraint139.Mode = catCstModeDrivingDimension

Dim reference149 As Reference

Set reference149 =
part1.CreateReferenceFromObject(point2D72)

Dim constraint140 As Constraint

Set constraint140 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference149)

constraint140.Mode = catCstModeDrivingDimension

Dim reference150 As Reference

Set reference150 =
part1.CreateReferenceFromObject(point2D73)

Dim constraint141 As Constraint

Set constraint141 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference150)

constraint141.Mode = catCstModeDrivingDimension

Dim reference151 As Reference

Set reference151 =
part1.CreateReferenceFromObject(point2D74)

Dim constraint142 As Constraint

Set constraint142 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference151)

constraint142.Mode = catCstModeDrivingDimension

Dim reference152 As Reference

Set reference152 =
part1.CreateReferenceFromObject(point2D75)

Dim constraint143 As Constraint

Set constraint143 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference152)

constraint143.Mode = catCstModeDrivingDimension

Dim reference153 As Reference

Set reference153 =
part1.CreateReferenceFromObject(point2D76)

Dim constraint144 As Constraint

Set constraint144 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference153)

constraint144.Mode = catCstModeDrivingDimension

Dim reference154 As Reference

Set reference154 =
part1.CreateReferenceFromObject(point2D77)

Dim constraint145 As Constraint

Set constraint145 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference154)

constraint145.Mode = catCstModeDrivingDimension


```

Dim reference155 As Reference
Set reference155 = part1.CreateReferenceFromObject(point2D78)
Dim constraint146 As Constraint
Set constraint146 = constraints2.AddBiEltCst(catCstTypeDistance, reference142, reference155)
constraint146.Mode = catCstModeDrivingDimension
Dim reference156 As Reference
Set reference156 = part1.CreateReferenceFromObject(point2D79)
Dim constraint147 As Constraint
Set constraint147 = constraints2.AddBiEltCst(catCstTypeDistance, reference142, reference156)
constraint147.Mode = catCstModeDrivingDimension
Dim reference157 As Reference
Set reference157 = part1.CreateReferenceFromObject(point2D80)
Dim constraint148 As Constraint
Set constraint148 = constraints2.AddBiEltCst(catCstTypeDistance, reference142, reference157)
constraint148.Mode = catCstModeDrivingDimension
Dim reference158 As Reference
Set reference158 = part1.CreateReferenceFromObject(point2D81)
Dim constraint149 As Constraint
Set constraint149 = constraints2.AddBiEltCst(catCstTypeDistance, reference142, reference158)
constraint149.Mode = catCstModeDrivingDimension
Dim reference159 As Reference
Set reference159 = part1.CreateReferenceFromObject(point2D82)
Dim constraint150 As Constraint
Set constraint150 = constraints2.AddBiEltCst(catCstTypeDistance, reference142, reference159)
constraint150.Mode = catCstModeDrivingDimension
Dim reference160 As Reference
Set reference160 = part1.CreateReferenceFromObject(point2D83)
Dim constraint151 As Constraint
Set constraint151 = constraints2.AddBiEltCst(catCstTypeDistance, reference142, reference160)
constraint151.Mode = catCstModeDrivingDimension
Dim reference161 As Reference
Set reference161 = part1.CreateReferenceFromObject(point2D84)
Dim constraint152 As Constraint
Set constraint152 = constraints2.AddBiEltCst(catCstTypeDistance, reference142, reference161)
constraint152.Mode = catCstModeDrivingDimension
Dim reference162 As Reference
Set reference162 = part1.CreateReferenceFromObject(point2D85)
Dim constraint153 As Constraint
Set constraint153 = constraints2.AddBiEltCst(catCstTypeDistance, reference142, reference162)
constraint153.Mode = catCstModeDrivingDimension
Dim reference163 As Reference
Set reference163 = part1.CreateReferenceFromObject(point2D86)
Dim constraint154 As Constraint
Set constraint154 = constraints2.AddBiEltCst(catCstTypeDistance, reference142, reference163)
constraint154.Mode = catCstModeDrivingDimension
Dim reference164 As Reference
Set reference164 = part1.CreateReferenceFromObject(point2D87)
Dim constraint155 As Constraint
Set constraint155 = constraints2.AddBiEltCst(catCstTypeDistance, reference142, reference164)
constraint155.Mode = catCstModeDrivingDimension
Dim reference165 As Reference
Set reference165 = part1.CreateReferenceFromObject(point2D88)
Dim constraint156 As Constraint
Set constraint156 = constraints2.AddBiEltCst(catCstTypeDistance, reference142, reference165)
constraint156.Mode = catCstModeDrivingDimension
Dim reference166 As Reference
Set reference166 = part1.CreateReferenceFromObject(point2D89)
Dim constraint157 As Constraint
Set constraint157 = constraints2.AddBiEltCst(catCstTypeDistance, reference142, reference166)
constraint157.Mode = catCstModeDrivingDimension
Dim reference167 As Reference
Set reference167 = part1.CreateReferenceFromObject(point2D90)
Dim constraint158 As Constraint

```

```

Set                constraint158                =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference167)
constraint158.Mode = catCstModeDrivingDimension
Dim reference168 As Reference
Set                reference168                =
part1.CreateReferenceFromObject(point2D91)
Dim constraint159 As Constraint
Set                constraint159                =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference168)
constraint159.Mode = catCstModeDrivingDimension
Dim reference169 As Reference
Set                reference169                =
part1.CreateReferenceFromObject(point2D92)
Dim constraint160 As Constraint
Set                constraint160                =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference169)
constraint160.Mode = catCstModeDrivingDimension
Dim reference170 As Reference
Set                reference170                =
part1.CreateReferenceFromObject(point2D93)
Dim constraint161 As Constraint
Set                constraint161                =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference170)
constraint161.Mode = catCstModeDrivingDimension
Dim reference171 As Reference
Set                reference171                =
part1.CreateReferenceFromObject(point2D94)
Dim constraint162 As Constraint
Set                constraint162                =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference171)
constraint162.Mode = catCstModeDrivingDimension
Dim reference172 As Reference
Set                reference172                =
part1.CreateReferenceFromObject(point2D95)
Dim constraint163 As Constraint
Set                constraint163                =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference172)
constraint163.Mode = catCstModeDrivingDimension
Dim reference173 As Reference
Set                reference173                =
part1.CreateReferenceFromObject(point2D96)
Dim constraint164 As Constraint
Set                constraint164                =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference173)
constraint164.Mode = catCstModeDrivingDimension
Dim reference174 As Reference
Set                reference174                =
part1.CreateReferenceFromObject(point2D67)
Dim constraint165 As Constraint
Set                constraint165                =
constraints2.AddBiEltCst(catCstTypeDistance, reference143,
reference174)
constraint165.Mode = catCstModeDrivingDimension
Dim reference175 As Reference
Set                reference175                =
part1.CreateReferenceFromObject(point2D68)
Dim constraint166 As Constraint
Set                constraint166                =
constraints2.AddBiEltCst(catCstTypeDistance, reference143,
reference175)
constraint166.Mode = catCstModeDrivingDimension
Dim reference176 As Reference
Set                reference176                =
part1.CreateReferenceFromObject(point2D69)
Dim constraint167 As Constraint
Set                constraint167                =
constraints2.AddBiEltCst(catCstTypeDistance, reference143,
reference176)
constraint167.Mode = catCstModeDrivingDimension
Dim reference177 As Reference
Set                reference177                =
part1.CreateReferenceFromObject(point2D70)
Dim constraint168 As Constraint
Set                constraint168                =
constraints2.AddBiEltCst(catCstTypeDistance, reference143,
reference177)
constraint168.Mode = catCstModeDrivingDimension
Dim reference178 As Reference
Set                reference178                =
part1.CreateReferenceFromObject(point2D71)
Dim constraint169 As Constraint
Set                constraint169                =
constraints2.AddBiEltCst(catCstTypeDistance, reference143,
reference178)
constraint169.Mode = catCstModeDrivingDimension
Dim reference179 As Reference
Set                reference179                =
part1.CreateReferenceFromObject(point2D72)

```

```

Dim constraint170 As Constraint
Set constraint170 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference179)
constraint170.Mode = catCstModeDrivingDimension
Dim reference180 As Reference
Set reference180 = part1.CreateReferenceFromObject(point2D73)
Dim constraint171 As Constraint
Set constraint171 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference180)
constraint171.Mode = catCstModeDrivingDimension
Dim reference181 As Reference
Set reference181 = part1.CreateReferenceFromObject(point2D74)
Dim constraint172 As Constraint
Set constraint172 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference181)
constraint172.Mode = catCstModeDrivingDimension
Dim reference182 As Reference
Set reference182 = part1.CreateReferenceFromObject(point2D75)
Dim constraint173 As Constraint
Set constraint173 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference182)
constraint173.Mode = catCstModeDrivingDimension
Dim reference183 As Reference
Set reference183 = part1.CreateReferenceFromObject(point2D76)
Dim constraint174 As Constraint
Set constraint174 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference183)
constraint174.Mode = catCstModeDrivingDimension
Dim reference184 As Reference
Set reference184 = part1.CreateReferenceFromObject(point2D77)
Dim constraint175 As Constraint
Set constraint175 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference184)
constraint175.Mode = catCstModeDrivingDimension
Dim reference185 As Reference
Set reference185 = part1.CreateReferenceFromObject(point2D78)
Dim constraint176 As Constraint
Set constraint176 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference185)
constraint176.Mode = catCstModeDrivingDimension
Dim reference186 As Reference
Set reference186 = part1.CreateReferenceFromObject(point2D79)
Dim constraint177 As Constraint
Set constraint177 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference186)
constraint177.Mode = catCstModeDrivingDimension
Dim reference187 As Reference
Set reference187 = part1.CreateReferenceFromObject(point2D80)
Dim constraint178 As Constraint
Set constraint178 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference187)
constraint178.Mode = catCstModeDrivingDimension
Dim reference188 As Reference
Set reference188 = part1.CreateReferenceFromObject(point2D81)
Dim constraint179 As Constraint
Set constraint179 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference188)
constraint179.Mode = catCstModeDrivingDimension
Dim reference189 As Reference
Set reference189 = part1.CreateReferenceFromObject(point2D82)
Dim constraint180 As Constraint
Set constraint180 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference189)
constraint180.Mode = catCstModeDrivingDimension
Dim reference190 As Reference
Set reference190 = part1.CreateReferenceFromObject(point2D83)
Dim constraint181 As Constraint
Set constraint181 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference190)
constraint181.Mode = catCstModeDrivingDimension
Dim reference191 As Reference
Set reference191 = part1.CreateReferenceFromObject(point2D84)
Dim constraint182 As Constraint
Set constraint182 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference191)
constraint182.Mode = catCstModeDrivingDimension

```

constraint182.Mode = catCstModeDrivingDimension
 Dim reference192 As Reference
 Set reference192 =
 part1.CreateReferenceFromObject(point2D85)
 Dim constraint183 As Constraint
 Set constraint183 =
 constraints2.AddBiEltCst(catCstTypeDistance, reference143,
 reference192)
 constraint183.Mode = catCstModeDrivingDimension
 Dim reference193 As Reference
 Set reference193 =
 part1.CreateReferenceFromObject(point2D86)
 Dim constraint184 As Constraint
 Set constraint184 =
 constraints2.AddBiEltCst(catCstTypeDistance, reference143,
 reference193)
 constraint184.Mode = catCstModeDrivingDimension
 Dim reference194 As Reference
 Set reference194 =
 part1.CreateReferenceFromObject(point2D87)
 Dim constraint185 As Constraint
 Set constraint185 =
 constraints2.AddBiEltCst(catCstTypeDistance, reference143,
 reference194)
 constraint185.Mode = catCstModeDrivingDimension
 Dim reference195 As Reference
 Set reference195 =
 part1.CreateReferenceFromObject(point2D88)
 Dim constraint186 As Constraint
 Set constraint186 =
 constraints2.AddBiEltCst(catCstTypeDistance, reference143,
 reference195)
 constraint186.Mode = catCstModeDrivingDimension
 Dim reference196 As Reference
 Set reference196 =
 part1.CreateReferenceFromObject(point2D89)
 Dim constraint187 As Constraint
 Set constraint187 =
 constraints2.AddBiEltCst(catCstTypeDistance, reference143,
 reference196)
 constraint187.Mode = catCstModeDrivingDimension
 Dim reference197 As Reference
 Set reference197 =
 part1.CreateReferenceFromObject(point2D90)
 Dim constraint188 As Constraint
 Set constraint188 =
 constraints2.AddBiEltCst(catCstTypeDistance, reference143,
 reference197)
 constraint188.Mode = catCstModeDrivingDimension
 Dim reference198 As Reference

Set reference198 =
 part1.CreateReferenceFromObject(point2D91)
 Dim constraint189 As Constraint
 Set constraint189 =
 constraints2.AddBiEltCst(catCstTypeDistance, reference143,
 reference198)
 constraint189.Mode = catCstModeDrivingDimension
 Dim reference199 As Reference
 Set reference199 =
 part1.CreateReferenceFromObject(point2D92)
 Dim constraint190 As Constraint
 Set constraint190 =
 constraints2.AddBiEltCst(catCstTypeDistance, reference143,
 reference199)
 constraint190.Mode = catCstModeDrivingDimension
 Dim reference200 As Reference
 Set reference200 =
 part1.CreateReferenceFromObject(point2D93)
 Dim constraint191 As Constraint
 Set constraint191 =
 constraints2.AddBiEltCst(catCstTypeDistance, reference143,
 reference200)
 constraint191.Mode = catCstModeDrivingDimension
 Dim reference201 As Reference
 Set reference201 =
 part1.CreateReferenceFromObject(point2D94)
 Dim constraint192 As Constraint
 Set constraint192 =
 constraints2.AddBiEltCst(catCstTypeDistance, reference143,
 reference201)
 constraint192.Mode = catCstModeDrivingDimension
 Dim reference202 As Reference
 Set reference202 =
 part1.CreateReferenceFromObject(point2D95)
 Dim constraint193 As Constraint
 Set constraint193 =
 constraints2.AddBiEltCst(catCstTypeDistance, reference143,
 reference202)
 constraint193.Mode = catCstModeDrivingDimension
 Dim reference203 As Reference
 Set reference203 =
 part1.CreateReferenceFromObject(point2D96)
 Dim constraint194 As Constraint
 Set constraint194 =
 constraints2.AddBiEltCst(catCstTypeDistance, reference143,
 reference203)
 constraint194.Mode = catCstModeDrivingDimension

' En tercer lugar vamos a acotar los puntos del Intradós por su
 distancia al eje vertical '

```

.....
Dim reference204 As Reference
Set          reference204          =
part1.CreateReferenceFromObject(point2D97)
Dim constraint195 As Constraint
Set          constraint195        =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference204)
constraint195.Mode = catCstModeDrivingDimension
Dim reference205 As Reference
Set          reference205        =
part1.CreateReferenceFromObject(point2D98)
Dim constraint196 As Constraint
Set          constraint196        =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference205)
constraint196.Mode = catCstModeDrivingDimension
Dim reference206 As Reference
Set          reference206        =
part1.CreateReferenceFromObject(point2D99)
Dim constraint197 As Constraint
Set          constraint197        =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference206)
constraint197.Mode = catCstModeDrivingDimension
Dim reference207 As Reference
Set          reference207        =
part1.CreateReferenceFromObject(point2D100)
Dim constraint198 As Constraint
Set          constraint198        =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference207)
constraint198.Mode = catCstModeDrivingDimension
Dim reference208 As Reference
Set          reference208        =
part1.CreateReferenceFromObject(point2D101)
Dim constraint199 As Constraint
Set          constraint199        =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference208)
constraint199.Mode = catCstModeDrivingDimension
Dim reference209 As Reference
Set          reference209        =
part1.CreateReferenceFromObject(point2D102)
Dim constraint200 As Constraint
Set          constraint200        =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference209)
constraint200.Mode = catCstModeDrivingDimension
Dim reference210 As Reference
Set          reference210        =
part1.CreateReferenceFromObject(point2D103)
Dim constraint201 As Constraint
Set          constraint201        =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference210)
constraint201.Mode = catCstModeDrivingDimension
Dim reference211 As Reference
Set          reference211        =
part1.CreateReferenceFromObject(point2D104)
Dim constraint202 As Constraint
Set          constraint202        =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference211)
constraint202.Mode = catCstModeDrivingDimension
Dim reference212 As Reference
Set          reference212        =
part1.CreateReferenceFromObject(point2D105)
Dim constraint203 As Constraint
Set          constraint203        =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference212)
constraint203.Mode = catCstModeDrivingDimension
Dim reference213 As Reference
Set          reference213        =
part1.CreateReferenceFromObject(point2D106)
Dim constraint204 As Constraint
Set          constraint204        =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference213)
constraint204.Mode = catCstModeDrivingDimension
Dim reference214 As Reference
Set          reference214        =
part1.CreateReferenceFromObject(point2D107)
Dim constraint205 As Constraint
Set          constraint205        =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference214)
constraint205.Mode = catCstModeDrivingDimension
Dim reference215 As Reference
Set          reference215        =
part1.CreateReferenceFromObject(point2D108)
Dim constraint206 As Constraint
Set          constraint206        =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference215)
constraint206.Mode = catCstModeDrivingDimension
Dim reference216 As Reference
Set          reference216        =
part1.CreateReferenceFromObject(point2D109)

```

```

Dim constraint207 As Constraint
Set constraint207 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference216)
constraint207.Mode = catCstModeDrivingDimension
Dim reference217 As Reference
Set reference217 =
part1.CreateReferenceFromObject(point2D110)
Dim constraint208 As Constraint
Set constraint208 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference217)
constraint208.Mode = catCstModeDrivingDimension
Dim reference218 As Reference
Set reference218 =
part1.CreateReferenceFromObject(point2D111)
Dim constraint209 As Constraint
Set constraint209 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference218)
constraint209.Mode = catCstModeDrivingDimension
Dim reference219 As Reference
Set reference219 =
part1.CreateReferenceFromObject(point2D112)
Dim constraint210 As Constraint
Set constraint210 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference219)
constraint210.Mode = catCstModeDrivingDimension
Dim reference220 As Reference
Set reference220 =
part1.CreateReferenceFromObject(point2D113)
Dim constraint211 As Constraint
Set constraint211 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference220)
constraint211.Mode = catCstModeDrivingDimension
Dim reference221 As Reference
Set reference221 =
part1.CreateReferenceFromObject(point2D114)
Dim constraint212 As Constraint
Set constraint212 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference221)
constraint212.Mode = catCstModeDrivingDimension
Dim reference222 As Reference
Set reference222 =
part1.CreateReferenceFromObject(point2D115)
Dim constraint213 As Constraint
Set constraint213 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference222)
constraint213.Mode = catCstModeDrivingDimension
Dim reference223 As Reference
Set reference223 =
part1.CreateReferenceFromObject(point2D116)
Dim constraint214 As Constraint
Set constraint214 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference223)
constraint214.Mode = catCstModeDrivingDimension
Dim reference224 As Reference
Set reference224 =
part1.CreateReferenceFromObject(point2D117)
Dim constraint215 As Constraint
Set constraint215 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference224)
constraint215.Mode = catCstModeDrivingDimension
Dim reference225 As Reference
Set reference225 =
part1.CreateReferenceFromObject(point2D118)
Dim constraint216 As Constraint
Set constraint216 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference225)
constraint216.Mode = catCstModeDrivingDimension
Dim reference226 As Reference
Set reference226 =
part1.CreateReferenceFromObject(point2D119)
Dim constraint217 As Constraint
Set constraint217 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference226)
constraint217.Mode = catCstModeDrivingDimension
Dim reference227 As Reference
Set reference227 =
part1.CreateReferenceFromObject(point2D120)
Dim constraint218 As Constraint
Set constraint218 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference227)
constraint218.Mode = catCstModeDrivingDimension
Dim reference228 As Reference
Set reference228 =
part1.CreateReferenceFromObject(point2D121)
Dim constraint219 As Constraint
Set constraint219 =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference228)

```

```

constraint219.Mode = catCstModeDrivingDimension
Dim reference229 As Reference
Set          reference229          =
part1.CreateReferenceFromObject(point2D122)
Dim constraint220 As Constraint
Set          constraint220         =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference229)
constraint220.Mode = catCstModeDrivingDimension
Dim reference230 As Reference
Set          reference230          =
part1.CreateReferenceFromObject(point2D123)
Dim constraint221 As Constraint
Set          constraint221         =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference230)
constraint221.Mode = catCstModeDrivingDimension
Dim reference231 As Reference
Set          reference231          =
part1.CreateReferenceFromObject(point2D124)
Dim constraint222 As Constraint
Set          constraint222         =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference231)
constraint222.Mode = catCstModeDrivingDimension
Dim reference232 As Reference
Set          reference232          =
part1.CreateReferenceFromObject(point2D125)
Dim constraint223 As Constraint
Set          constraint223         =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference232)
constraint223.Mode = catCstModeDrivingDimension
Dim reference233 As Reference
Set          reference233          =
part1.CreateReferenceFromObject(point2D126)
Dim constraint224 As Constraint
Set          constraint224         =
constraints2.AddBiEltCst(catCstTypeDistance, reference142,
reference233)
constraint224.Mode = catCstModeDrivingDimension
.....
' En último lugar vamos a acotar los puntos del Intradós por su
distancia al eje horizontal '
.....
Dim reference234 As Reference
Set          reference234          =
part1.CreateReferenceFromObject(point2D97)
Dim constraint225 As Constraint
Set          constraint225         =
constraints2.AddBiEltCst(catCstTypeDistance, reference143,
reference234)
constraint225.Mode = catCstModeDrivingDimension
Dim reference235 As Reference
Set          reference235          =
part1.CreateReferenceFromObject(point2D98)
Dim constraint226 As Constraint
Set          constraint226         =
constraints2.AddBiEltCst(catCstTypeDistance, reference143,
reference235)
constraint226.Mode = catCstModeDrivingDimension
Dim reference236 As Reference
Set          reference236          =
part1.CreateReferenceFromObject(point2D99)
Dim constraint227 As Constraint
Set          constraint227         =
constraints2.AddBiEltCst(catCstTypeDistance, reference143,
reference236)
constraint227.Mode = catCstModeDrivingDimension
Dim reference237 As Reference
Set          reference237          =
part1.CreateReferenceFromObject(point2D100)
Dim constraint228 As Constraint
Set          constraint228         =
constraints2.AddBiEltCst(catCstTypeDistance, reference143,
reference237)
constraint228.Mode = catCstModeDrivingDimension
Dim reference238 As Reference
Set          reference238          =
part1.CreateReferenceFromObject(point2D101)
Dim constraint229 As Constraint
Set          constraint229         =
constraints2.AddBiEltCst(catCstTypeDistance, reference143,
reference238)
constraint229.Mode = catCstModeDrivingDimension
Dim reference239 As Reference
Set          reference239          =
part1.CreateReferenceFromObject(point2D102)
Dim constraint230 As Constraint
Set          constraint230         =
constraints2.AddBiEltCst(catCstTypeDistance, reference143,
reference239)
constraint230.Mode = catCstModeDrivingDimension
Dim reference240 As Reference
Set          reference240          =
part1.CreateReferenceFromObject(point2D103)
Dim constraint231 As Constraint
Set          constraint231         =
constraints2.AddBiEltCst(catCstTypeDistance, reference143,
reference240)

```

constraint231.Mode = catCstModeDrivingDimension		Set	reference247	=	part1.CreateReferenceFromObject(point2D110)
Dim reference241 As Reference					Dim constraint238 As Constraint
Set	reference241	=			Set
part1.CreateReferenceFromObject(point2D104)			constraint238	=	constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference247)
Dim constraint232 As Constraint					constraint238.Mode = catCstModeDrivingDimension
Set	constraint232	=			Dim reference248 As Reference
constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference241)					Set
constraint232.Mode = catCstModeDrivingDimension			reference248	=	part1.CreateReferenceFromObject(point2D111)
Dim reference242 As Reference					Dim constraint239 As Constraint
Set	reference242	=			Set
part1.CreateReferenceFromObject(point2D105)			constraint239	=	constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference248)
Dim constraint233 As Constraint					constraint239.Mode = catCstModeDrivingDimension
Set	constraint233	=			Dim reference249 As Reference
constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference242)					Set
constraint233.Mode = catCstModeDrivingDimension			reference249	=	part1.CreateReferenceFromObject(point2D112)
Dim reference243 As Reference					Dim constraint240 As Constraint
Set	reference243	=			Set
part1.CreateReferenceFromObject(point2D106)			constraint240	=	constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference249)
Dim constraint234 As Constraint					constraint240.Mode = catCstModeDrivingDimension
Set	constraint234	=			Dim reference250 As Reference
constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference243)					Set
constraint234.Mode = catCstModeDrivingDimension			reference250	=	part1.CreateReferenceFromObject(point2D113)
Dim reference244 As Reference					Dim constraint241 As Constraint
Set	reference244	=			Set
part1.CreateReferenceFromObject(point2D107)			constraint241	=	constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference250)
Dim constraint235 As Constraint					constraint241.Mode = catCstModeDrivingDimension
Set	constraint235	=			Dim reference251 As Reference
constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference244)					Set
constraint235.Mode = catCstModeDrivingDimension			reference251	=	part1.CreateReferenceFromObject(point2D114)
Dim reference245 As Reference					Dim constraint242 As Constraint
Set	reference245	=			Set
part1.CreateReferenceFromObject(point2D108)			constraint242	=	constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference251)
Dim constraint236 As Constraint					constraint242.Mode = catCstModeDrivingDimension
Set	constraint236	=			Dim reference252 As Reference
constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference245)					Set
constraint236.Mode = catCstModeDrivingDimension			reference252	=	part1.CreateReferenceFromObject(point2D115)
Dim reference246 As Reference					Dim constraint243 As Constraint
Set	reference246	=			Set
part1.CreateReferenceFromObject(point2D109)			constraint243	=	constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference252)
Dim constraint237 As Constraint					constraint243.Mode = catCstModeDrivingDimension
Set	constraint237	=			Dim reference253 As Reference
constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference246)					Set
constraint237.Mode = catCstModeDrivingDimension			reference253	=	part1.CreateReferenceFromObject(point2D116)
Dim reference247 As Reference					


```

Dim constraint244 As Constraint
Set constraint244 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference253)
constraint244.Mode = catCstModeDrivingDimension
Dim reference254 As Reference
Set reference254 = part1.CreateReferenceFromObject(point2D117)
Dim constraint245 As Constraint
Set constraint245 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference254)
constraint245.Mode = catCstModeDrivingDimension
Dim reference255 As Reference
Set reference255 = part1.CreateReferenceFromObject(point2D118)
Dim constraint246 As Constraint
Set constraint246 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference255)
constraint246.Mode = catCstModeDrivingDimension
Dim reference256 As Reference
Set reference256 = part1.CreateReferenceFromObject(point2D119)
Dim constraint247 As Constraint
Set constraint247 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference256)
constraint247.Mode = catCstModeDrivingDimension
Dim reference257 As Reference
Set reference257 = part1.CreateReferenceFromObject(point2D120)
Dim constraint248 As Constraint
Set constraint248 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference257)
constraint248.Mode = catCstModeDrivingDimension
Dim reference258 As Reference
Set reference258 = part1.CreateReferenceFromObject(point2D121)
Dim constraint249 As Constraint
Set constraint249 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference258)
constraint249.Mode = catCstModeDrivingDimension
Dim reference259 As Reference
Set reference259 = part1.CreateReferenceFromObject(point2D122)
Dim constraint250 As Constraint
Set constraint250 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference259)
constraint250.Mode = catCstModeDrivingDimension
Dim reference260 As Reference
Set reference260 = part1.CreateReferenceFromObject(point2D123)
Dim constraint251 As Constraint
Set constraint251 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference260)
constraint251.Mode = catCstModeDrivingDimension
Dim reference261 As Reference
Set reference261 = part1.CreateReferenceFromObject(point2D124)
Dim constraint252 As Constraint
Set constraint252 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference261)
constraint252.Mode = catCstModeDrivingDimension
Dim reference262 As Reference
Set reference262 = part1.CreateReferenceFromObject(point2D125)
Dim constraint253 As Constraint
Set constraint253 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference262)
constraint253.Mode = catCstModeDrivingDimension
Dim reference263 As Reference
Set reference263 = part1.CreateReferenceFromObject(point2D126)
Dim constraint254 As Constraint
Set constraint254 = constraints2.AddBiEltCst(catCstTypeDistance, reference143, reference263)
constraint254.Mode = catCstModeDrivingDimension
' Acotamos el control point del borde de ataque en el cual creamos una fórmula para que este varíe al variar el perfil '
Dim reference264 As Reference
Set reference264 = part1.CreateReferenceFromObject(point2D127)
Dim constraint255 As Constraint
Set constraint255 = constraints2.AddBiEltCst(catCstTypeDistance, reference142, reference264)
constraint255.Mode = catCstModeDrivingDimension
Dim length255 As Length
Set length255 = constraint255.Dimension
length255.Value = 0

```

```

Dim reference265 As Reference
Set          reference265          =
part1.CreateReferenceFromObject(point2D127)
Dim constraint256 As Constraint
Set          constraint256        =
constraints2.AddBiEltCst(catCstTypeDistance, reference143,
reference265)
constraint256.Mode = catCstModeDrivingDimension
Dim length256 As Length
Set length256 = constraint256.Dimension
Dim formula11 As Formula
Set formula11 = relations.CreateFormula("Punto control borde
de ataque raiz", "Borde de ataque", length256,
"Semienvergadura*tan(Diedro) + Semienvergadura")
' Una vez creado todos los puntos del perfil, realizamos un
spline alrededor de todos los puntos.
' En dicho spline hemos unido todos los puntos exceptuando
los situados en el borde de salida. Ya que si unimos
' estos dos puntos en el mismo spline este se distorsiona.
'
Dim arrayOfObject3(60)
Set arrayOfObject3(0) = point2D96
Set arrayOfObject3(1) = point2D95
Set arrayOfObject3(2) = point2D94
Set arrayOfObject3(3) = point2D93
Set arrayOfObject3(4) = point2D92
Set arrayOfObject3(5) = point2D91
Set arrayOfObject3(6) = point2D90
Set arrayOfObject3(7) = point2D89
Set arrayOfObject3(8) = point2D88
Set arrayOfObject3(9) = point2D87
Set arrayOfObject3(10) = point2D86
Set arrayOfObject3(11) = point2D85
Set arrayOfObject3(12) = point2D84
Set arrayOfObject3(13) = point2D83
Set arrayOfObject3(14) = point2D82
Set arrayOfObject3(15) = point2D81
Set arrayOfObject3(16) = point2D80
Set arrayOfObject3(17) = point2D79
Set arrayOfObject3(18) = point2D78
Set arrayOfObject3(19) = point2D77
Set arrayOfObject3(20) = point2D76
Set arrayOfObject3(21) = point2D75
Set arrayOfObject3(22) = point2D74
Set arrayOfObject3(23) = point2D73
Set arrayOfObject3(24) = point2D72
Set arrayOfObject3(25) = point2D71
Set arrayOfObject3(26) = point2D70
Set arrayOfObject3(27) = point2D69
Set arrayOfObject3(28) = point2D68
Set arrayOfObject3(29) = point2D67
Set arrayOfObject3(30) = point2D127
Set arrayOfObject3(31) = point2D97
Set arrayOfObject3(32) = point2D98
Set arrayOfObject3(33) = point2D99
Set arrayOfObject3(34) = point2D100
Set arrayOfObject3(35) = point2D101
Set arrayOfObject3(36) = point2D102
Set arrayOfObject3(37) = point2D103
Set arrayOfObject3(38) = point2D104
Set arrayOfObject3(39) = point2D105
Set arrayOfObject3(40) = point2D106
Set arrayOfObject3(41) = point2D107
Set arrayOfObject3(42) = point2D108
Set arrayOfObject3(43) = point2D109
Set arrayOfObject3(44) = point2D110
Set arrayOfObject3(45) = point2D111
Set arrayOfObject3(46) = point2D112
Set arrayOfObject3(47) = point2D113
Set arrayOfObject3(48) = point2D114
Set arrayOfObject3(49) = point2D115
Set arrayOfObject3(50) = point2D116
Set arrayOfObject3(51) = point2D117
Set arrayOfObject3(52) = point2D118
Set arrayOfObject3(53) = point2D119
Set arrayOfObject3(54) = point2D120
Set arrayOfObject3(55) = point2D121
Set arrayOfObject3(56) = point2D122
Set arrayOfObject3(57) = point2D123
Set arrayOfObject3(58) = point2D124
Set arrayOfObject3(59) = point2D125
Set arrayOfObject3(60) = point2D126
Dim spline2D3 As Spline2D
Set factory2D2temp = factory2D2
Set          spline2D3          =
factory2D2temp.CreateSpline(arrayOfObject3)
spline2D1.ReportName = 141

```

```

' Unimos los dos puntos del borde de salida mediante otro
spline el cual creará una línea recta entre estos dos puntos '
' aunque lo ideal sería unirlos mediante un arco de
circunferencia el cual aun no sabemos hacer.
'

Dim arrayOfObject4(1)
Set arrayOfObject4(0) = point2D96
Set arrayOfObject4(1) = point2D126

Dim spline2D4 As Spline2D
Set factory2D2temp = factory2D2

Set spline2D4 =
factory2D2temp.CreateSpline(arrayOfObject4)

spline2D1.ReportName = 142

' Cerramos el sketch
sketch2.CloseEdition

" Creamos las Regla a través de las cuales tendremos
parametrizados nuestro perfil en la raíz.
"
" Vamos a crear una regla para cada punto del perfil.
"
" Posteriormente modificamos la Regla e introducimos las
líneas de códigos obtenidas con MATLAB.
"
" Al ser líneas de códigos muy largas, estas no caben en una
sola línea, por lo cual habrá que poner al final de cada línea
"
" " & _ para poder continuar debajo y cuando queramos que
cambie de línea & vbCrLf & (para cambiar de línea dentro del
mismo) "

Dim rule61 As Rule

Set rule61 = relations.CreateProgram("rule61", "", "")

rule61.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _

"if (20/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.135\Offset = (20/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(20/1000))-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA2
T/10)**2)*((NACA2T/10)-(20/1000)))) else
PartBody\Sketch.2\Offset.135\Offset = (20/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(20/1000))-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(20/1000))))" & vbCrLf &
_

"if (20/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.165\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(20/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
20/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(20/1000))-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA2
T/10)**2)*((NACA2T/10)-(20/1000)))) else
PartBody\Sketch.2\Offset.165\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
20/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+20/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(20/1000))-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(20/1000))))"

Dim rule62 As Rule

Set rule62 = relations.CreateProgram("rule62", "", "")

rule62.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _

"if (40/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.136\Offset = (40/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(40/1000))-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA2
T/10)**2)*((NACA2T/10)-(40/1000)))) else
PartBody\Sketch.2\Offset.136\Offset = (40/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(40/1000))-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(40/1000))))" & vbCrLf &
_

"if (40/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.166\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(40/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
40/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(40/1000))-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA2
T/10)**2)*((NACA2T/10)-(40/1000)))) else
PartBody\Sketch.2\Offset.166\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
40/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+40/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(40/1000))-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(40/1000))))"

```

```
0.1015*(40/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(40/1000))))"
```

Dim rule63 As Rule

```
Set rule63 = relations.CreateProgram("rule63", "", "")
```

```
rule63.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (60/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.137\Offset = (60/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA2
T/10)**2)*((NACA2T/10)-(60/1000))))" else
PartBody\Sketch.2\Offset.137\Offset = (60/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(60/1000))))" & vbCrLf
& _
```

```
"if (60/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.167\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(60/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
60/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA2
T/10)**2)*((NACA2T/10)-(60/1000))))" else
PartBody\Sketch.2\Offset.167\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
60/1000*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+60/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(60/1000))))"
```

Dim rule64 As Rule

```
Set rule64 = relations.CreateProgram("rule64", "", "")
```

```
rule64.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (80/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.138\Offset = (80/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA2
T/10)**2)*((NACA2T/10)-(80/1000))))" else
PartBody\Sketch.2\Offset.138\Offset = (80/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(80/1000)-
```

```
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(80/1000))))" & vbCrLf
& _
```

```
"if (80/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.168\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(80/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
80/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA2
T/10)**2)*((NACA2T/10)-(80/1000))))" else
PartBody\Sketch.2\Offset.168\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
80/1000*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+80/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(80/1000))))"
```

Dim rule65 As Rule

```
Set rule65 = relations.CreateProgram("rule65", "", "")
```

```
rule65.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (100/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.139\Offset = (100/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(100/1000))))" else
PartBody\Sketch.2\Offset.139\Offset = (100/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(100/1000))))" &
vbCrLf & _
```

```
"if (100/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.169\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(100/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
100/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(100/1000))))" else
PartBody\Sketch.2\Offset.169\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
```

```
100/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+100/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(100/1000))))))"
```

Dim rule66 As Rule

Set rule66 = relations.CreateProgram("rule66", "", "")

rule66.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (136/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.140\Offset = (136/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(136/1000)))))) else
PartBody\Sketch.2\Offset.140\Offset = (136/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(136/1000))))))" &
vbCrLf & _
```

```
"if (136/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.170\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(136/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
136/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(136/1000)))))) else
PartBody\Sketch.2\Offset.170\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
136/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+136/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(136/1000))))))"
```

Dim rule67 As Rule

Set rule67 = relations.CreateProgram("rule67", "", "")

rule67.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (172/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.141\Offset = (172/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
```

```
0.1015*(172/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(172/1000)))))) else
PartBody\Sketch.2\Offset.141\Offset = (172/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(172/1000))))))" &
vbCrLf & _
```

```
"if (172/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.171\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(172/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
172/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(172/1000)))))) else
PartBody\Sketch.2\Offset.171\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
172/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+172/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(172/1000))))))"
```

Dim rule68 As Rule

Set rule68 = relations.CreateProgram("rule68", "", "")

rule68.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (208/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.142\Offset = (208/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(208/1000)))))) else
PartBody\Sketch.2\Offset.142\Offset = (208/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(208/1000))))))" &
vbCrLf & _
```

```
"if (208/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.172\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(208/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
208/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
```

```

2T/10)**2*((NACA2T/10)-(208/1000))))    else
PartBody\Sketch.2\Offset.172\Offset      =
(Semienvergadura*tan(Diedro))+(Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
208/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+208/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(208/1000))))))"

Dim rule69 As Rule
Set rule69 = relations.CreateProgram("rule69", "", "")

rule69.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _

"if (244/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.143\Offset = (244/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(244/1000))))))    else
PartBody\Sketch.2\Offset.143\Offset = (244/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(244/1000))))))" &
vbCrLf & _

"if (244/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.173\Offset      =
(Semienvergadura*tan(Diedro))+(Semienvergadura
+((NACA1T/100)*(244/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
244/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(244/1000))))))    else
PartBody\Sketch.2\Offset.173\Offset      =
(Semienvergadura*tan(Diedro))+(Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
244/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+244/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(244/1000))))))"

Dim rule70 As Rule
Set rule70 = relations.CreateProgram("rule70", "", "")

rule70.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _

"if (280/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.144\Offset = (280/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(280/1000))))))    else
PartBody\Sketch.2\Offset.144\Offset = (280/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(280/1000))))))" &
vbCrLf & _

"if (280/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.174\Offset      =
(Semienvergadura*tan(Diedro))+(Semienvergadura
+((NACA1T/100)*(280/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
280/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(280/1000))))))    else
PartBody\Sketch.2\Offset.174\Offset      =
(Semienvergadura*tan(Diedro))+(Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
280/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+280/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(280/1000))))))"

Dim rule71 As Rule
Set rule71 = relations.CreateProgram("rule71", "", "")

rule71.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _

"if (316/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.145\Offset = (316/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(316/1000))))))    else
PartBody\Sketch.2\Offset.145\Offset = (316/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(316/1000))))))" &
vbCrLf & _

"if (316/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.175\Offset      =
(Semienvergadura*tan(Diedro))+(Semienvergadura
+((NACA1T/100)*(316/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
316/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(316/1000))))))"

```

```
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
316/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*cos(atan(2*(NACA1T/100)/(NACA
2T/10)**2)*((NACA2T/10)-(316/1000))))))" else
PartBody\Sketch.2\Offset.175\Offset
=
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
316/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+316/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(316/1000))))))"
```

Dim rule72 As Rule

Set rule72 = relations.CreateProgram("rule72", "", "")

```
rule72.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (352/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.146\Offset = (352/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*sin(atan(2*(NACA1T/100)/(NACA
2T/10)**2)*((NACA2T/10)-(352/1000))))" else
PartBody\Sketch.2\Offset.146\Offset = (352/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(352/1000))))" &
vbCrLf & _
```

```
"if (352/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.176\Offset
=
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(352/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
352/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*cos(atan(2*(NACA1T/100)/(NACA
2T/10)**2)*((NACA2T/10)-(352/1000))))" else
PartBody\Sketch.2\Offset.176\Offset
=
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
352/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+352/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(352/1000))))"
```

Dim rule73 As Rule

Set rule73 = relations.CreateProgram("rule73", "", "")

```
rule73.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (388/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.147\Offset = (388/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*sin(atan(2*(NACA1T/100)/(NACA
2T/10)**2)*((NACA2T/10)-(388/1000))))" else
PartBody\Sketch.2\Offset.147\Offset = (388/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(388/1000))))" &
vbCrLf & _
```

```
"if (388/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.177\Offset
=
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(388/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
388/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*cos(atan(2*(NACA1T/100)/(NACA
2T/10)**2)*((NACA2T/10)-(388/1000))))" else
PartBody\Sketch.2\Offset.177\Offset
=
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
388/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+388/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(388/1000))))"
```

Dim rule74 As Rule

Set rule74 = relations.CreateProgram("rule60", "", "")

```
rule74.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (424/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.148\Offset = (424/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4)*sin(atan(2*(NACA1T/100)/(NACA
2T/10)**2)*((NACA2T/10)-(424/1000))))" else
PartBody\Sketch.2\Offset.148\Offset = (424/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(424/1000))))" &
vbCrLf & _
```

```
"if (424/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
```

```

Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.178\Offset
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(424/1000)*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
424/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4*cos(atan(2*(NACA1T/100)/(NACA
2T/10)**2*(NACA2T/10)-(424/1000))))"
PartBody\Sketch.2\Offset.178\Offset
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
424/1000)*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+424/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2*(NACA2T/10)-(424/1000))))"
Dim rule75 As Rule
Set rule75 = relations.CreateProgram("rule60", "", "")
rule75.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (460/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.149\Offset = (460/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(460/1000)-
0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4)*sin(atan(2*(NACA1T/100)/(NACA
2T/10)**2*(NACA2T/10)-(460/1000))))"
PartBody\Sketch.2\Offset.149\Offset = (460/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(460/1000)-
0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2*(NACA2T/10)-(460/1000))))"
PartBody\Sketch.2\Offset.180\Offset
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(496/1000)*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
496/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(496/1000)-
0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4*cos(atan(2*(NACA1T/100)/(NACA
2T/10)**2*(NACA2T/10)-(496/1000))))"
PartBody\Sketch.2\Offset.180\Offset
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
496/1000)*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+496/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(496/1000)-
0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2*(NACA2T/10)-(496/1000))))"
Dim rule77 As Rule
Set rule77 = relations.CreateProgram("rule77", "", "")
rule77.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (532/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.151\Offset = (532/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*sin(atan(2*(NACA1T/100)/(NACA
2T/10)**2*(NACA2T/10)-(532/1000))))"
PartBody\Sketch.2\Offset.151\Offset = (532/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2*(NACA2T/10)-(532/1000))))"

```



```

0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(532/1000))))" &
vbCrLf & _

"if (532/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.181\Offset =
(Semienvergadura*tan(Diedro))+(Semienvergadura
+((NACA1T/100)*(532/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
532/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(532/1000))))" else
PartBody\Sketch.2\Offset.181\Offset =
(Semienvergadura*tan(Diedro))+(Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
532/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+532/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(532/1000))))"

Dim rule78 As Rule

Set rule78 = relations.CreateProgram("rule78", "", "")

rule78.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _

"if (568/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.152\Offset = (568/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(568/1000))))" else
PartBody\Sketch.2\Offset.152\Offset = (568/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(568/1000))))" &
vbCrLf & _

"if (568/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.182\Offset =
(Semienvergadura*tan(Diedro))+(Semienvergadura+((NACA
1T/100)*(568/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
568/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(568/1000))))" else
PartBody\Sketch.2\Offset.182\Offset =
(Semienvergadura*tan(Diedro))+(Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
568/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+568/1000+1)/(1-

```

```

NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(568/1000))))"

Dim rule79 As Rule

Set rule79 = relations.CreateProgram("rule79", "", "")

rule79.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _

"if (604/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.153\Offset = (604/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(604/1000))))" else
PartBody\Sketch.2\Offset.153\Offset = (604/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(604/1000))))" &
vbCrLf & _

"if (604/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.183\Offset =
(Semienvergadura*tan(Diedro))+(Semienvergadura
+((NACA1T/100)*(604/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
604/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(604/1000))))" else
PartBody\Sketch.2\Offset.183\Offset =
(Semienvergadura*tan(Diedro))+(Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
604/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+604/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(604/1000))))"

Dim rule80 As Rule

Set rule80 = relations.CreateProgram("rule80", "", "")

rule80.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _

"if (640/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.154\Offset = (640/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(640/1000))))" else

```

```

PartBody\Sketch.2\Offset.154\Offset = (640/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(640/1000))))" &
vbCrLf & _
"if (640/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.184\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(640/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
640/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(640/1000))))" else
PartBody\Sketch.2\Offset.184\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+((NACA
1T/100)*((100-Estrechamiento)/100)*Cuerda-
640/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+640/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(640/1000))))"
Dim rule81 As Rule
Set rule81 = relations.CreateProgram("rule81", "", "")
rule81.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (676/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.155\Offset = (676/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(676/1000)-
0.1260*(676/1000)-
0.3516*(676/1000)**2+0.2843*(676/1000)**3-
0.1015*(676/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(676/1000))))" else
PartBody\Sketch.2\Offset.155\Offset = (676/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(676/1000)-
0.1260*(676/1000)-
0.3516*(676/1000)**2+0.2843*(676/1000)**3-
0.1015*(676/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(676/1000))))" &
vbCrLf & _
"if (676/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.185\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(676/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
676/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(676/1000)-
0.1260*(676/1000)-
0.3516*(676/1000)**2+0.2843*(676/1000)**3-
0.1015*(676/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(676/1000))))" else
PartBody\Sketch.2\Offset.185\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
712/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+712/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(712/1000))))"
Dim rule82 As Rule
Set rule82 = relations.CreateProgram("rule82", "", "")
rule82.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (712/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.156\Offset = (712/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(712/1000))))" else
PartBody\Sketch.2\Offset.156\Offset = (712/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(712/1000))))" &
vbCrLf & _
"if (712/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.186\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(712/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
712/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(712/1000))))" else
PartBody\Sketch.2\Offset.186\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
712/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+712/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(712/1000))))"
Dim rule83 As Rule
Set rule83 = relations.CreateProgram("rule83", "", "")
rule83.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (748/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.157\Offset = (748/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(748/1000))))"

```

```
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(748/1000))))))" else
PartBody\Sketch.2\Offset.157\Offset = (748/1000)*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(748/1000))))))" &
vbCrLf & _
```

```
"if (748/1000)*((100-Estrechamiento)/100)*Cuerda <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.187\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*((748/1000)*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
748/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(748/1000))))))" else
PartBody\Sketch.2\Offset.187\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
748/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+748/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(748/1000))))))"
```

Dim rule84 As Rule

Set rule84 = relations.CreateProgram("rule84", "", "")

rule84.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (784/1000)*((100-Estrechamiento)/100)*Cuerda <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.158\Offset = (784/1000)*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(784/1000))))))" else
PartBody\Sketch.2\Offset.158\Offset = (784/1000)*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(784/1000))))))" &
vbCrLf & _
```

```
"if (784/1000)*((100-Estrechamiento)/100)*Cuerda <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.188\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*((784/1000)*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
784/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
```

```
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(784/1000))))))" else
PartBody\Sketch.2\Offset.188\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
784/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+784/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(784/1000))))))"
```

Dim rule85 As Rule

Set rule85 = relations.CreateProgram("rule85", "", "")

rule85.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (820/1000)*((100-Estrechamiento)/100)*Cuerda <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.159\Offset = (820/1000)*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(820/1000))))))" else
PartBody\Sketch.2\Offset.159\Offset = (820/1000)*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(820/1000))))))" &
vbCrLf & _
```

```
"if (820/1000)*((100-Estrechamiento)/100)*Cuerda <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.189\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*((820/1000)*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
820/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(820/1000))))))" else
PartBody\Sketch.2\Offset.189\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
820/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+820/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(820/1000))))))"
```

Dim rule86 As Rule

Set rule86 = relations.CreateProgram("rule86", "", "")

rule86.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (856/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.160\Offset = (856/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(856/1000))))" else
PartBody\Sketch.2\Offset.160\Offset = (856/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(856/1000))))" &
vbCrLf & _
```

```
"if (856/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.190\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(856/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
856/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(856/1000))))" else
PartBody\Sketch.2\Offset.190\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
856/1000*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+856/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(856/1000))))" &
```

Dim rule87 As Rule

```
Set rule87 = relations.CreateProgram("rule87", "", "")
```

```
rule87.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (892/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.161\Offset = (892/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(892/1000))))" else
PartBody\Sketch.2\Offset.161\Offset = (892/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(892/1000))))" &
vbCrLf & _
```

```
"if (892/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.191\Offset =
```

```
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(892/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
892/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(892/1000))))" else
PartBody\Sketch.2\Offset.191\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
892/1000*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+892/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(892/1000))))" &
```

Dim rule88 As Rule

```
Set rule88 = relations.CreateProgram("rule88", "", "")
```

```
rule88.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (928/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.162\Offset = (928/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(928/1000))))" else
PartBody\Sketch.2\Offset.162\Offset = (928/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(928/1000))))" &
vbCrLf & _
```

```
"if (928/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.192\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(928/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
928/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(928/1000))))" else
PartBody\Sketch.2\Offset.192\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
928/1000*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+928/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(928/1000))))" &
```

Dim rule89 As Rule

```
Set rule89 = relations.CreateProgram("rule89", "", "")
rule89.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _
```

```
"if (964/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.163\Offset = (964/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(964/1000))))" else
PartBody\Sketch.2\Offset.163\Offset = (964/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(964/1000))))" &
vbCrLf & _
```

```
"if (964/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.193\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(964/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
964/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(964/1000))))" else
PartBody\Sketch.2\Offset.193\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
964/1000*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+964/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(964/1000))))" &
```

Dim rule90 As Rule

```
Set rule90 = relations.CreateProgram("rule90", "", "")
rule90.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _
```

```
"if (1000/1000*((100-Estrechamiento)/100)*Cuerda) <=
(((100-Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.164\Offset = (1000/1000*((100 -
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100 -
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*sin(atan(2*(NACA1T/100)/((NAC
A2T/10)**2)*((NACA2T/10)-(1000/1000))))" else
PartBody\Sketch.2\Offset.164\Offset = (1000/1000*((100-
Estrechamiento)/100)*Cuerda - NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(1000/1000))))" &
```

vbCrLf & _

```
"if (1000/1000*((100-Estrechamiento)/100)*Cuerda) <=
(((100-Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.194\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(1000/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
1000/1000)/(NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*cos(atan(2*(NACA1T/100)/((NAC
A2T/10)**2)*((NACA2T/10)-(1000/1000))))" else
PartBody\Sketch.2\Offset.194\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
1000/1000*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+1000/1000+1)/(1-
NACA2T/10)**2)+NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(1000/1000))))" &
```

Dim rule91 As Rule

```
Set rule91 = relations.CreateProgram("rule91", "", "")
rule91.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _
```

```
"if (20/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.195\Offset = (20/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA2
T/10)**2)*((NACA2T/10)-(20/1000))))" else
PartBody\Sketch.2\Offset.195\Offset = (20/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(20/1000))))" & vbCrLf & _
```

```
"if (20/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.225\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+((NACA
1T/100)*(20/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
20/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA2
T/10)**2)*((NACA2T/10)-(20/1000))))" else
PartBody\Sketch.2\Offset.225\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
20/1000*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+20/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(20/1000)-
0.1260*(20/1000)-
```

```
0.3516*(20/1000)**2+0.2843*(20/1000)**3-
0.1015*(20/1000)**4*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(20/1000))))"
```

Dim rule92 As Rule

```
Set rule92 = relations.CreateProgram("rule92", "", "")
```

```
rule92.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (40/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.196\Offset = (40/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(40/1000)-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4*sin(atan(2*(NACA1T/100)/((NACA2
T/10)**2)*((NACA2T/10)-(40/1000))))" else
PartBody\Sketch.2\Offset.196\Offset = (40/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(40/1000)-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(40/1000))))" & vbCrLf
& _
```

```
"if (40/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.226\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+(NACA
1T/100)*(40/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
40/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(40/1000)-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4*cos(atan(2*(NACA1T/100)/((NACA2
T/10)**2)*((NACA2T/10)-(40/1000))))" else
PartBody\Sketch.2\Offset.226\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+(NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
40/1000*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+40/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(40/1000)-
0.1260*(40/1000)-
0.3516*(40/1000)**2+0.2843*(40/1000)**3-
0.1015*(40/1000)**4*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(40/1000))))"
```

Dim rule93 As Rule

```
Set rule93 = relations.CreateProgram("rule93", "", "")
```

```
rule93.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (60/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.197\Offset = (60/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4*sin(atan(2*(NACA1T/100)/((NACA2
T/10)**2)*((NACA2T/10)-(60/1000))))" else
PartBody\Sketch.2\Offset.197\Offset = (60/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
```

```
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(60/1000))))" & vbCrLf
& _
```

```
"if (60/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.227\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+(NACA
1T/100)*(60/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
60/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4*cos(atan(2*(NACA1T/100)/((NACA2
T/10)**2)*((NACA2T/10)-(60/1000))))" else
PartBody\Sketch.2\Offset.227\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+(NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
60/1000*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+60/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(60/1000)-
0.1260*(60/1000)-
0.3516*(60/1000)**2+0.2843*(60/1000)**3-
0.1015*(60/1000)**4*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(60/1000))))"
```

Dim rule94 As Rule

```
Set rule94 = relations.CreateProgram("rule94", "", "")
```

```
rule94.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (80/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.198\Offset = (80/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4*sin(atan(2*(NACA1T/100)/((NACA2
T/10)**2)*((NACA2T/10)-(80/1000))))" else
PartBody\Sketch.2\Offset.198\Offset = (80/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(80/1000))))" & vbCrLf
& _
```

```
"if (80/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.228\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+(NACA
1T/100)*(80/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
80/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4*cos(atan(2*(NACA1T/100)/((NACA2
T/10)**2)*((NACA2T/10)-(80/1000))))" else
PartBody\Sketch.2\Offset.228\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
```

```

+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
80/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+80/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(80/1000)-
0.1260*(80/1000)-
0.3516*(80/1000)**2+0.2843*(80/1000)**3-
0.1015*(80/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(80/1000))))"

```

Dim rule95 As Rule

Set rule95 = relations.CreateProgram("rule90", "", "")

rule95.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```

"if (100/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.199\Offset = (100/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(100/1000)))) else
PartBody\Sketch.2\Offset.199\Offset = (100/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(100/1000))))" &
vbCrLf & _

```

```

"if (100/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.229\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+((NACA
1T/100)*(100/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
100/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(100/1000)))) else
PartBody\Sketch.2\Offset.229\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
100/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+100/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(100/1000)-
0.1260*(100/1000)-
0.3516*(100/1000)**2+0.2843*(100/1000)**3-
0.1015*(100/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(100/1000))))"

```

Dim rule96 As Rule

Set rule96 = relations.CreateProgram("rule96", "", "")

rule96.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```

"if (136/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.200\Offset = (136/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(136/1000)))) else
PartBody\Sketch.2\Offset.200\Offset = (136/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(136/1000))))" &
vbCrLf & _

```

```

0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(136/1000)))) else
PartBody\Sketch.2\Offset.200\Offset = (136/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(136/1000))))" &
vbCrLf & _

```

```

"if (136/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.230\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+((NACA
1T/100)*(136/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
136/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(136/1000)))) else
PartBody\Sketch.2\Offset.230\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
136/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+136/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(136/1000)-
0.1260*(136/1000)-
0.3516*(136/1000)**2+0.2843*(136/1000)**3-
0.1015*(136/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(136/1000))))"

```

Dim rule97 As Rule

Set rule97 = relations.CreateProgram("rule97", "", "")

rule97.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```

"if (172/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.201\Offset = (172/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(172/1000)))) else
PartBody\Sketch.2\Offset.201\Offset = (172/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(172/1000))))" &
vbCrLf & _

```

```

"if (172/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.231\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+((NACA
1T/100)*(172/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
172/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(172/1000))))" &
vbCrLf & _

```

```

0.1015*(172/1000)**4*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(172/1000))))           else
PartBody\Sketch.2\Offset.231\Offset                =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
172/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+172/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(172/1000)-
0.1260*(172/1000)-
0.3516*(172/1000)**2+0.2843*(172/1000)**3-
0.1015*(172/1000)**4*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(172/1000))))))"

```

Dim rule98 As Rule

Set rule98 = relations.CreateProgram("rule98", "", "")

rule98.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```

"if (208/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.202\Offset = (208/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(208/1000))))           else
PartBody\Sketch.2\Offset.202\Offset = (208/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(208/1000))))))" &
vbCrLf & _

```

```

"if (208/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.232\Offset                =
(Semienvergadura*tan(Diedro))+Semienvergadura+((NACA
1T/100)*(208/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
208/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(208/1000))))           else
PartBody\Sketch.2\Offset.232\Offset                =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
208/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+208/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(208/1000)-
0.1260*(208/1000)-
0.3516*(208/1000)**2+0.2843*(208/1000)**3-
0.1015*(208/1000)**4*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(208/1000))))))"

```

Dim rule99 As Rule

Set rule99 = relations.CreateProgram("rule99", "", "")

rule99.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```

"if (244/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-

```

```

Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.203\Offset = (244/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(244/1000))))           else
PartBody\Sketch.2\Offset.203\Offset = (244/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(244/1000))))))" &
vbCrLf & _

```

```

"if (244/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.233\Offset                =
(Semienvergadura*tan(Diedro))+Semienvergadura+((NACA
1T/100)*(244/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
244/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(244/1000))))           else
PartBody\Sketch.2\Offset.233\Offset                =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
244/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+244/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(244/1000)-
0.1260*(244/1000)-
0.3516*(244/1000)**2+0.2843*(244/1000)**3-
0.1015*(244/1000)**4*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(244/1000))))))"

```

Dim rule100 As Rule

Set rule100 = relations.CreateProgram("rule100", "", "")

rule100.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```

"if (280/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.204\Offset = (280/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(280/1000))))           else
PartBody\Sketch.2\Offset.204\Offset = (280/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(280/1000))))))" &
vbCrLf & _

```

```

"if (280/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.234\Offset                =
(Semienvergadura*tan(Diedro))+Semienvergadura+((NACA

```



```
1T/100)*(280/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
280/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*cos(atan(2*(NACA1T/100)/(NACA
2T/10)**2*((NACA2T/10)-(280/1000)))))) else
PartBody\Sketch.2\Offset.234\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
280/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+280/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(280/1000)-
0.1260*(280/1000)-
0.3516*(280/1000)**2+0.2843*(280/1000)**3-
0.1015*(280/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2*((NACA2T/10)-(280/1000))))))"
```

Dim rule101 As Rule

```
Set rule101 = relations.CreateProgram("rule101", "", "")
```

```
rule101.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (316/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.205\Offset = (316/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*sin(atan(2*(NACA1T/100)/(NACA
2T/10)**2*((NACA2T/10)-(316/1000)))))) else
PartBody\Sketch.2\Offset.205\Offset = (316/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2*((NACA2T/10)-(316/1000))))))" &
vbCrLf & _
```

```
"if (316/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.235\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura+(NACA
1T/100)*(316/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
316/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*cos(atan(2*(NACA1T/100)/(NACA
2T/10)**2*((NACA2T/10)-(316/1000)))))) else
PartBody\Sketch.2\Offset.235\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
316/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+316/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(316/1000)-
0.1260*(316/1000)-
0.3516*(316/1000)**2+0.2843*(316/1000)**3-
0.1015*(316/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2*((NACA2T/10)-(316/1000))))))"
```

Dim rule102 As Rule

```
Set rule102 = relations.CreateProgram("rule102", "", "")
```

```
rule102.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (352/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.206\Offset = (352/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*sin(atan(2*(NACA1T/100)/(NACA
2T/10)**2*((NACA2T/10)-(352/1000)))))) else
PartBody\Sketch.2\Offset.206\Offset = (352/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2*((NACA2T/10)-(352/1000))))))" &
vbCrLf & _
```

```
"if (352/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.236\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura+(NACA
1T/100)*(352/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
352/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*cos(atan(2*(NACA1T/100)/(NACA
2T/10)**2*((NACA2T/10)-(352/1000)))))) else
PartBody\Sketch.2\Offset.236\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
352/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+352/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(352/1000)-
0.1260*(352/1000)-
0.3516*(352/1000)**2+0.2843*(352/1000)**3-
0.1015*(352/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2*((NACA2T/10)-(352/1000))))))"
```

Dim rule103 As Rule

```
Set rule103 = relations.CreateProgram("rule103", "", "")
```

```
rule103.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (388/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.207\Offset = (388/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*sin(atan(2*(NACA1T/100)/(NACA
2T/10)**2*((NACA2T/10)-(388/1000)))))) else
PartBody\Sketch.2\Offset.207\Offset = (388/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2*((NACA2T/10)-(388/1000))))))" &
vbCrLf & _
```

```
"if (388/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.237\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura+((NACA
1T/100)*(388/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
388/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(388/1000))))" else
PartBody\Sketch.2\Offset.237\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
388/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+388/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(388/1000)-
0.1260*(388/1000)-
0.3516*(388/1000)**2+0.2843*(388/1000)**3-
0.1015*(388/1000)**4*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(388/1000))))"
```

Dim rule104 As Rule

Set rule104 = relations.CreateProgram("rule104", "", "")

rule104.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (424/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.208\Offset = (424/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(424/1000))))" else
PartBody\Sketch.2\Offset.208\Offset = (424/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(424/1000))))" &
vbCrLf & _
```

```
"if (424/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.238\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura+((NACA
1T/100)*(424/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
424/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
0.1015*(424/1000)**4*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(424/1000))))" else
PartBody\Sketch.2\Offset.238\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
424/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+424/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(424/1000)-
0.1260*(424/1000)-
0.3516*(424/1000)**2+0.2843*(424/1000)**3-
```

```
0.1015*(424/1000)**4*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(424/1000))))"
```

Dim rule105 As Rule

Set rule105 = relations.CreateProgram("rule105", "", "")

rule105.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (460/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.209\Offset = (460/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(460/1000)-
0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(460/1000))))" else
PartBody\Sketch.2\Offset.209\Offset = (460/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(460/1000)-
0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(460/1000))))" &
vbCrLf & _
```

```
"if (460/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.239\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura+((NACA
1T/100)*(460/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
460/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(460/1000)-
0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(460/1000))))" else
PartBody\Sketch.2\Offset.239\Offset =
(Semienvergadura*tan(Diedro))+Semienvergadura
+((NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
460/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+460/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(460/1000)-
0.1260*(460/1000)-
0.3516*(460/1000)**2+0.2843*(460/1000)**3-
0.1015*(460/1000)**4*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(460/1000))))"
```

Dim rule106 As Rule

Set rule106 = relations.CreateProgram("rule106", "", "")

rule106.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (496/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.210\Offset = (496/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(496/1000)-
0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(496/1000))))" else
PartBody\Sketch.2\Offset.210\Offset = (496/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(496/1000)-
```

```

0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/100)**2)*(NACA2T/10)-(496/1000))))" &
vbCrLf & _

"if (496/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.240\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+(NACA
1T/100)*(496/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
496/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(496/1000)-
0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4)*cos(atan(2*(NACA1T/100)/(NACA
2T/10)**2)*(NACA2T/10)-(496/1000))))" else
PartBody\Sketch.2\Offset.240\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+(NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
496/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+496/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(496/1000)-
0.1260*(496/1000)-
0.3516*(496/1000)**2+0.2843*(496/1000)**3-
0.1015*(496/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/100)**2)*(NACA2T/10)-(496/1000))))"

Dim rule107 As Rule

Set rule107 = relations.CreateProgram("rule107", "", "")

rule107.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _

"if (532/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.211\Offset = (532/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*sin(atan(2*(NACA1T/100)/(NACA
2T/10)**2)*(NACA2T/10)-(532/1000))))" else
PartBody\Sketch.2\Offset.211\Offset = (532/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/100)**2)*(NACA2T/10)-(532/1000))))" &
vbCrLf & _

"if (532/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.241\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+(NACA
1T/100)*(532/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
532/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*cos(atan(2*(NACA1T/100)/(NACA
2T/10)**2)*(NACA2T/10)-(532/1000))))" else
PartBody\Sketch.2\Offset.241\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+(NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
532/1000*((100-Estrechamiento)/100)*Cuerda)*(-

```

```

2*(NACA2T/10)+532/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(532/1000)-
0.1260*(532/1000)-
0.3516*(532/1000)**2+0.2843*(532/1000)**3-
0.1015*(532/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/100)**2)*(NACA2T/10)-(532/1000))))"

Dim rule108 As Rule

Set rule108 = relations.CreateProgram("rule108", "", "")

rule108.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _

"if (568/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.212\Offset = (568/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*sin(atan(2*(NACA1T/100)/(NACA
2T/10)**2)*(NACA2T/10)-(568/1000))))" else
PartBody\Sketch.2\Offset.212\Offset = (568/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/100)**2)*(NACA2T/10)-(568/1000))))" &
vbCrLf & _

"if (568/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.242\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+(NACA
1T/100)*(568/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
568/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*cos(atan(2*(NACA1T/100)/(NACA
2T/10)**2)*(NACA2T/10)-(568/1000))))" else
PartBody\Sketch.2\Offset.242\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+(NACA1T/100)*(((100-Estrechamiento)/100)*Cuerda-
568/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+568/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(568/1000)-
0.1260*(568/1000)-
0.3516*(568/1000)**2+0.2843*(568/1000)**3-
0.1015*(568/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/100)**2)*(NACA2T/10)-(568/1000))))"

Dim rule109 As Rule

Set rule109 = relations.CreateProgram("rule109", "", "")

rule109.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _

"if (604/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.213\Offset = (604/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*sin(atan(2*(NACA1T/100)/(NACA

```

```

2T/10)**2*((NACA2T/10)-(604/1000))))    else
PartBody\Sketch.2\Offset.213\Offset = (604/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(604/1000))))" &
vbCrLf & _
"if (604/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.243\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+((NACA
1T/100)*(604/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
604/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(604/1000))))" else
PartBody\Sketch.2\Offset.243\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
604/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+604/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(604/1000)-
0.1260*(604/1000)-
0.3516*(604/1000)**2+0.2843*(604/1000)**3-
0.1015*(604/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(604/1000))))"
Dim rule110 As Rule
Set rule110 = relations.CreateProgram("rule110", "", "")
rule110.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (640/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.214\Offset = (640/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(640/1000))))" else
PartBody\Sketch.2\Offset.214\Offset = (640/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(640/1000))))" &
vbCrLf & _
"if (640/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.244\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+((NACA
1T/100)*(640/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
640/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(640/1000))))" else
2T/10)**2*((NACA2T/10)-(640/1000))))    else
PartBody\Sketch.2\Offset.244\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+((NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
640/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+640/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(640/1000)-
0.1260*(640/1000)-
0.3516*(640/1000)**2+0.2843*(640/1000)**3-
0.1015*(640/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(640/1000))))"
Dim rule111 As Rule
Set rule111 = relations.CreateProgram("rule111", "", "")
rule111.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (676/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.215\Offset = (676/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(676/1000)-
0.1260*(676/1000)-
0.3516*(676/1000)**2+0.2843*(676/1000)**3-
0.1015*(676/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(676/1000))))" else
PartBody\Sketch.2\Offset.215\Offset = (676/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(676/1000)-
0.1260*(676/1000)-
0.3516*(676/1000)**2+0.2843*(676/1000)**3-
0.1015*(676/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(676/1000))))" &
vbCrLf & _
"if (676/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.245\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+((NACA
1T/100)*(676/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
676/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(676/1000)-
0.1260*(676/1000)-
0.3516*(676/1000)**2+0.2843*(676/1000)**3-
0.1015*(676/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(676/1000))))" else
PartBody\Sketch.2\Offset.245\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+((NACA
1T/100)*((100-Estrechamiento)/100)*Cuerda-
676/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+676/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(676/1000)-
0.1260*(676/1000)-
0.3516*(676/1000)**2+0.2843*(676/1000)**3-
0.1015*(676/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(676/1000))))"
Dim rule112 As Rule
Set rule112 = relations.CreateProgram("rule112", "", "")
rule112.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
"if (712/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))

```

```
PartBody\Sketch.2\Offset.216\Offset = (712/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(712/1000))))))" else
PartBody\Sketch.2\Offset.216\Offset = (712/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(712/1000))))))" &
vbCrLf & _
```

```
"if (712/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.246\Offset =
(Semienvergadura*tan(Diedro))+((NACA
1T/100)*(712/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
712/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(712/1000))))" else
PartBody\Sketch.2\Offset.246\Offset =
(Semienvergadura*tan(Diedro))+((Semienvergadura+(NACA
1T/100)*((100-Estrechamiento)/100)*Cuerda-
712/1000*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+712/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(712/1000)-
0.1260*(712/1000)-
0.3516*(712/1000)**2+0.2843*(712/1000)**3-
0.1015*(712/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(712/1000)))))"
```

Dim rule113 As Rule

Set rule113 = relations.CreateProgram("rule113", "", "")

```
rule113.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (748/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.217\Offset = (748/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(748/1000))))" else
PartBody\Sketch.2\Offset.217\Offset = (748/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(748/1000))))" &
vbCrLf & _
```

```
"if (748/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.247\Offset =
(Semienvergadura*tan(Diedro))+((Semienvergadura+(NACA
1T/100)*(748/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
```

```
748/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(748/1000))))" else
PartBody\Sketch.2\Offset.247\Offset =
(Semienvergadura*tan(Diedro))+((Semienvergadura
+((NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
748/1000*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+748/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(748/1000)-
0.1260*(748/1000)-
0.3516*(748/1000)**2+0.2843*(748/1000)**3-
0.1015*(748/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(748/1000))))" &
```

Dim rule114 As Rule

Set rule114 = relations.CreateProgram("rule114", "", "")

```
rule114.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*/" & vbCrLf & _
```

```
"if (784/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.218\Offset = (784/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(784/1000))))" else
PartBody\Sketch.2\Offset.218\Offset = (784/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(784/1000))))" &
vbCrLf & _
```

```
"if (784/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.248\Offset =
(Semienvergadura*tan(Diedro))+((Semienvergadura+(NACA
1T/100)*(784/1000*((100-
Estrechamiento)/100)*Cuerda)*(2*(NACA2T/10)-
784/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(784/1000))))" else
PartBody\Sketch.2\Offset.248\Offset =
(Semienvergadura*tan(Diedro))+((Semienvergadura
+((NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
784/1000*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+784/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(784/1000)-
0.1260*(784/1000)-
0.3516*(784/1000)**2+0.2843*(784/1000)**3-
0.1015*(784/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(784/1000))))" &
```

Dim rule115 As Rule

Set rule115 = relations.CreateProgram("rule115", "", "")

```
rule115.Modify "/*Rule created by Francisco Javier Jiménez
```

```

Calvo*" & vbCrLf & _
"if (820/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.219\Offset = (820/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(820/1000))))" else
PartBody\Sketch.2\Offset.219\Offset = (820/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(820/1000))))" &
vbCrLf & _

"if (820/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.249\Offset =
(Semienvergadura*tan(Diedro))+((Semienvergadura+((NACA
1T/100)*(820/1000)*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
820/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(820/1000))))" else
PartBody\Sketch.2\Offset.249\Offset =
(Semienvergadura*tan(Diedro))+((Semienvergadura
+((NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
820/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+820/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(820/1000)-
0.1260*(820/1000)-
0.3516*(820/1000)**2+0.2843*(820/1000)**3-
0.1015*(820/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(820/1000))))"

Dim rule116 As Rule
Set rule116 = relations.CreateProgram("rule116", "", "")
rule116.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*" & vbCrLf & _

"if (856/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.220\Offset = (856/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(856/1000))))" else
PartBody\Sketch.2\Offset.220\Offset = (856/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(856/1000))))" &
vbCrLf & _

"if (856/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.250\Offset =
(Semienvergadura*tan(Diedro))+((Semienvergadura+((NACA
1T/100)*(856/1000)*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
856/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(856/1000))))" else
PartBody\Sketch.2\Offset.250\Offset =
(Semienvergadura*tan(Diedro))+((Semienvergadura
+((NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
856/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+856/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(856/1000)-
0.1260*(856/1000)-
0.3516*(856/1000)**2+0.2843*(856/1000)**3-
0.1015*(856/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(856/1000))))"

Dim rule117 As Rule
Set rule117 = relations.CreateProgram("rule117", "", "")
rule117.Modify "/*Rule created by Francisco Javier Jiménez
Calvo*" & vbCrLf & _

"if (892/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.221\Offset = (892/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(892/1000))))" else
PartBody\Sketch.2\Offset.221\Offset = (892/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(892/1000))))" &
vbCrLf & _

"if (892/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.251\Offset =
(Semienvergadura*tan(Diedro))+((Semienvergadura+((NACA
1T/100)*(892/1000)*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
892/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(892/1000))))" else
PartBody\Sketch.2\Offset.251\Offset =
(Semienvergadura*tan(Diedro))+((Semienvergadura
+((NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
892/1000*((100-Estrechamiento)/100)*Cuerda)*(-
2*(NACA2T/10)+892/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(892/1000)-
0.1260*(892/1000)-
0.3516*(892/1000)**2+0.2843*(892/1000)**3-
0.1015*(892/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(892/1000))))"

```

Dim rule118 As Rule

Set rule118 = relations.CreateProgram("rule118", "", "")

rule118.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (928/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.222\Offset = (928/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(928/1000))))" else
PartBody\Sketch.2\Offset.222\Offset = (928/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(928/1000))))" &
vbCrLf & _
```

```
"if (928/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.252\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+(NACA
1T/100)*(928/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
928/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(928/1000))))" else
PartBody\Sketch.2\Offset.252\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+(NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
928/1000*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+928/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(928/1000)-
0.1260*(928/1000)-
0.3516*(928/1000)**2+0.2843*(928/1000)**3-
0.1015*(928/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(928/1000))))" &
```

Dim rule119 As Rule

Set rule119 = relations.CreateProgram("rule119", "", "")

rule119.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (964/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.223\Offset = (964/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*sin(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(964/1000))))" else
PartBody\Sketch.2\Offset.223\Offset = (964/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(964/1000))))" &
```

vbCrLf & _

```
"if (964/1000*((100-Estrechamiento)/100)*Cuerda) <= (((100-
Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.253\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+(NACA
1T/100)*(964/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
964/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*cos(atan(2*(NACA1T/100)/((NACA
2T/10)**2)*((NACA2T/10)-(964/1000))))" else
PartBody\Sketch.2\Offset.253\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+(NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
964/1000*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+964/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(964/1000)-
0.1260*(964/1000)-
0.3516*(964/1000)**2+0.2843*(964/1000)**3-
0.1015*(964/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(964/1000))))" &
```

Dim rule120 As Rule

Set rule120 = relations.CreateProgram("rule120", "", "")

rule120.Modify "/*Rule created by Francisco Javier Jiménez Calvo*/" & vbCrLf & _

```
"if (1000/1000*((100-Estrechamiento)/100)*Cuerda) <=
(((100-Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.224\Offset = (1000/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*sin(atan(2*(NACA1T/100)/((NAC
A2T/10)**2)*((NACA2T/10)-(1000/1000))))" else
PartBody\Sketch.2\Offset.224\Offset = (1000/1000*((100-
Estrechamiento)/100)*Cuerda + NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*sin(atan(2*(NACA1T/100)/((1-
(NACA2T/10)**2)*((NACA2T/10)-(1000/1000))))" &
vbCrLf & _
```

```
"if (1000/1000*((100-Estrechamiento)/100)*Cuerda) <=
(((100-Estrechamiento)/100)*Cuerda*(NACA2T/10))
PartBody\Sketch.2\Offset.254\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura+(NACA
1T/100)*(1000/1000*((100-
Estrechamiento)/100)*Cuerda*(2*(NACA2T/10)-
1000/1000)/(NACA2T/10)**2)-NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*cos(atan(2*(NACA1T/100)/((NAC
A2T/10)**2)*((NACA2T/10)-(1000/1000))))" else
PartBody\Sketch.2\Offset.254\Offset =
(Semienvergadura*tan(Diedro))+ (Semienvergadura
+(NACA1T/100)*((100-Estrechamiento)/100)*Cuerda-
1000/1000*((100-Estrechamiento)/100)*Cuerda*(-
2*(NACA2T/10)+1000/1000+1)/(1-NACA2T/10)**2)-
NACA34T/(100*0.2)*((100-
Estrechamiento)/100)*Cuerda*(0.2969*sqrt(1000/1000)-
0.1260*(1000/1000)-
```

```

0.3516*(1000/1000)**2+0.2843*(1000/1000)**3-
0.1015*(1000/1000)**4)*cos(atan(2*(NACA1T/100)/((1-
(NACA2T/10))**2)*((NACA2T/10)-(1000/1000))))"
.....
" LINEA PUNTOS BORDE DE ATAQUE "
.....
'Creamos la linea que unirá los bordes de ataque de ambos
perfiles
Dim reference266 As Reference
Set reference266 =
part1.CreateReferenceFromBRepName("BorderFVertex:(BE
dge:(Brp:(Sketch.1;64);None:(Limits1:();Limits2:();+1);Cf11:()
);WithPermanentBody;WithoutBuildError;WithSelectingFeatu

```

```

reSupport;MFBRepVersion_CXR15)", sketch1)
Dim reference267 As Reference
Set reference267 =
part1.CreateReferenceFromBRepName("BorderFVertex:(BE
dge:(Brp:(Sketch.2;134);None:(Limits1:();Limits2:();+1);Cf11:()
);WithPermanentBody;WithoutBuildError;WithSelectingFeatu
reSupport;MFBRepVersion_CXR15)", sketch2)
Dim hybridShapeLinePtPt1 As HybridShapeLinePtPt
Set hybridShapeLinePtPt1 =
hybridShapeFactory1.AddNewLinePtPt(reference266,
reference267)
body1.InsertHybridShape hybridShapeLinePtPt1
part1.InWorkObject = hybridShapeLinePtPt1

```

ANEXO 4. CÓDIGO ALA IZQUIERDA SUPERFICIE

```

.....
" EXTRUSIÓN DEL ALA (SUPERFICIE) "
.....
Dim hybridShapeLoft1 As HybridShapeLoft
Set hybridShapeLoft1 = hybridShapeFactory1.AddNewLoft()
hybridShapeLoft1.SectionCoupling = 1
hybridShapeLoft1.Relimitation = 1
hybridShapeLoft1.CanonicalDetection = 2
Dim reference268 As Reference
Set reference268 =
part1.CreateReferenceFromObject(hybridShapeLinePtPt1)
hybridShapeLoft1.AddGuide reference268
Dim reference269 As Reference
Set reference269 =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;142);None:(Limits1:();Limits2:());Cf11:());WithP
ermanentBody;WithoutBuildError;WithSelectingFeatureSupp
ort;MFBRepVersion_CXR15)", sketch1)
hybridShapeLoft1.AddSectionToLoft reference269, 1,
Nothing
Dim reference270 As Reference
Set reference270 =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.2;1);None:(Limits1:();Limits2:());Cf11:());WithP
ermanentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch2)
hybridShapeLoft1.AddSectionToLoft reference270, 1,
Nothing

```

```

body1.InsertHybridShape hybridShapeLoft1
part1.InWorkObject = hybridShapeLoft1
part1.Update
.....
Dim hybridShapeLoft2 As HybridShapeLoft
Set hybridShapeLoft2 = hybridShapeFactory1.AddNewLoft()
hybridShapeLoft2.SectionCoupling = 1
hybridShapeLoft2.Relimitation = 1
hybridShapeLoft2.CanonicalDetection = 2
Dim reference271 As Reference
Set reference271 =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;1);None:(Limits1:();Limits2:());Cf11:());WithP
ermanentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch1)
hybridShapeLoft2.AddSectionToLoft reference271, 1,
Nothing
Dim reference272 As Reference
Set reference272 =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.2;2);None:(Limits1:();Limits2:());Cf11:());WithP
ermanentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch2)
hybridShapeLoft2.AddSectionToLoft reference272, 1,
Nothing
body1.InsertHybridShape hybridShapeLoft2
part1.InWorkObject = hybridShapeLoft2
part1.Update

```



```

CATIA.ActiveDocument.Part.Update
.....
" FILL DE LOS EXTREMOS Y JOIN (SUPERFICIE) "
.....

Dim hybridShapeFill1 As HybridShapeFill
Set hybridShapeFill1 = hybridShapeFactory1.AddNewFill()
Dim reference273 As Reference
Set reference273 = part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;1);None:(Limits1:();Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRRepVersion_CXR15)", sketch1)
hybridShapeFill1.AddBound reference273
Dim reference274 As Reference
Set reference274 = part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;142);None:(Limits1:();Limits2:());Cf11:());WithP
ermanentBody;WithoutBuildError;WithSelectingFeatureSupp
ort;MFBRRepVersion_CXR15)", sketch1)
hybridShapeFill1.AddBound reference274
hybridShapeFill1.Continuity = 0
body1.InsertHybridShape hybridShapeFill1
part1.InWorkObject = hybridShapeFill1
part1.Update
.....

Dim hybridShapeFill2 As HybridShapeFill
Set hybridShapeFill2 = hybridShapeFactory1.AddNewFill()
hybridShapeFill2.AddBound reference270
hybridShapeFill2.AddBound reference272
hybridShapeFill2.Continuity = 0
body1.InsertHybridShape hybridShapeFill2
part1.InWorkObject = hybridShapeFill2
part1.Update
.....

Dim reference275 As Reference
Set reference275 = part1.CreateReferenceFromObject(hybridShapeLoft1)
Dim reference276 As Reference
Set reference276 = part1.CreateReferenceFromObject(hybridShapeLoft2)
Dim hybridShapeAssemble1 As HybridShapeAssemble
Set hybridShapeAssemble1 = hybridShapeFactory1.AddNewJoin(reference275,
reference276)

Dim reference277 As Reference
Set reference277 = part1.CreateReferenceFromObject(hybridShapeFill1)
hybridShapeAssemble1.AddElement reference277
Dim reference278 As Reference
Set reference278 = part1.CreateReferenceFromObject(hybridShapeFill2)
hybridShapeAssemble1.AddElement reference278
hybridShapeAssemble1.SetConnex 1
hybridShapeAssemble1.SetManifold 0
hybridShapeAssemble1.SetSimplify 0
hybridShapeAssemble1.SetSuppressMode 0
hybridShapeAssemble1.SetDeviation 0.001
hybridShapeAssemble1.SetAngularToleranceMode 0
hybridShapeAssemble1.SetAngularTolerance 0.5
hybridShapeAssemble1.SetFederationPropagation 0
body1.InsertHybridShape hybridShapeAssemble1
part1.InWorkObject = hybridShapeAssemble1
part1.Update
.....

" HIDE SKETCHES & LINES "
.....

Dim selection1 As Selection
Set selection1 = partDocument1.Selection
Dim visPropertySet1 As VisPropertySet
Set visPropertySet1 = selection1.VisProperties
Set sketches1 = sketch1.Parent
Dim bSTR1 As String
bSTR1 = sketch1.Name
selection1.Add sketch1
Set visPropertySet1 = visPropertySet1.Parent
Dim bSTR2 As String
bSTR2 = visPropertySet1.Name
Dim bSTR3 As String
bSTR3 = visPropertySet1.Name
visPropertySet1.SetShow 1
selection1.Clear
Dim selection2 As Selection
Set selection2 = partDocument1.Selection
Dim visPropertySet2 As VisPropertySet
Set visPropertySet2 = selection2.VisProperties
Set hybridShapes1 = hybridShapePlaneOffset1.Parent

```

```

Dim bSTR4 As String
bSTR4 = hybridShapePlaneOffset1.Name
selection2.Add hybridShapePlaneOffset1
Set visPropertySet2 = visPropertySet2.Parent
Dim bSTR5 As String
bSTR5 = visPropertySet2.Name
Dim bSTR6 As String
bSTR6 = visPropertySet2.Name
visPropertySet2.SetShow 1
selection2.Clear
Dim selection3 As Selection
Set selection3 = partDocument1.Selection
Dim visPropertySet3 As VisPropertySet
Set visPropertySet3 = selection3.VisProperties
Set sketches1 = sketch2.Parent
Dim bSTR7 As String
bSTR7 = sketch2.Name
selection3.Add sketch2
Set visPropertySet3 = visPropertySet3.Parent
Dim bSTR8 As String
bSTR8 = visPropertySet3.Name
Dim bSTR9 As String
bSTR9 = visPropertySet3.Name
visPropertySet3.SetShow 1
selection3.Clear
Dim selection4 As Selection
Set selection4 = partDocument1.Selection
Dim visPropertySet4 As VisPropertySet
Set visPropertySet4 = selection4.VisProperties
Set hybridShapes1 = hybridShapeLinePtPt1.Parent
Dim bSTR10 As String
bSTR10 = hybridShapeLinePtPt1.Name
selection4.Add hybridShapeLinePtPt1
Set visPropertySet4 = visPropertySet4.Parent
Dim bSTR11 As String
bSTR11 = visPropertySet4.Name
Dim bSTR12 As String
bSTR12 = visPropertySet4.Name
visPropertySet4.SetShow 1
selection4.Clear
.....
" ACTUALIZAMOS Y CERRAMOS EL DOCUMENTO"
.....
' Se actualiza el documento y se cierra la interface
CATIA.ActiveDocument.Part.Update
Unload Me

```

ANEXO 5. CÓDIGO ALA DERECHA SUPERFICIE

```

.....
" EXTRUSIÓN DEL ALA (SUPERFICIE) "
.....
Dim hybridShapeLoft1 As HybridShapeLoft
Set hybridShapeLoft1 = hybridShapeFactory1.AddNewLoft()
hybridShapeLoft1.SectionCoupling = 1
hybridShapeLoft1.Relimitation = 1
hybridShapeLoft1.CanonicalDetection = 2
Dim reference268 As Reference
Set reference268 = part1.CreateReferenceFromObject(hybridShapeLinePtPt1)
hybridShapeLoft1.AddGuide reference268
Dim reference269 As Reference
Set reference269 = part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;142);None:(Limits1:());Limits2:());Cf11:());WithP
ermanentBody;WithoutBuildError;WithSelectingFeatureSupp
ort;MFBRPVersion_CXR15)", sketch1)
hybridShapeLoft1.AddSectionToLoft reference269, 1,
Nothing
Dim reference270 As Reference
Set reference270 = part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.2;1);None:(Limits1:());Limits2:());Cf11:());WithP
ermanentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRPVersion_CXR15)", sketch2)
hybridShapeLoft1.AddSectionToLoft reference270, 1,
Nothing
body1.InsertHybridShape hybridShapeLoft1
part1.InWorkObject = hybridShapeLoft1

```

```

part1.Update
.....
.....
Dim hybridShapeLoft2 As HybridShapeLoft
Set hybridShapeLoft2 = hybridShapeFactory1.AddNewLoft()
hybridShapeLoft2.SectionCoupling = 1
hybridShapeLoft2.Relimitation = 1
hybridShapeLoft2.CanonicalDetection = 2
Dim reference271 As Reference
Set reference271 = part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;1);None:(Limits1:();Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch1)
hybridShapeLoft2.AddSectionToLoft reference271, 1,
Nothing
Dim reference272 As Reference
Set reference272 = part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.2;2);None:(Limits1:();Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch2)
hybridShapeLoft2.AddSectionToLoft reference272, 1,
Nothing
body1.InsertHybridShape hybridShapeLoft2
part1.InWorkObject = hybridShapeLoft2
part1.Update
CATIA.ActiveDocument.Part.Update
.....
.....
" FILL DE LOS EXTREMOS Y JOIN (SUPERFICIE) "
.....
.....
Dim hybridShapeFill1 As HybridShapeFill
Set hybridShapeFill1 = hybridShapeFactory1.AddNewFill()
Dim reference273 As Reference
Set reference273 = part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;1);None:(Limits1:();Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch1)
hybridShapeFill1.AddBound reference273
Dim reference274 As Reference
Set reference274 = part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;142);None:(Limits1:();Limits2:());Cf11:());WithP
ermanentBody;WithoutBuildError;WithSelectingFeatureSupp
ort;MFBRepVersion_CXR15)", sketch1)
hybridShapeFill1.AddBound reference274
hybridShapeFill1.Continuity = 0

```

```

body1.InsertHybridShape hybridShapeFill1
part1.InWorkObject = hybridShapeFill1
part1.Update
.....
.....
Dim hybridShapeFill2 As HybridShapeFill
Set hybridShapeFill2 = hybridShapeFactory1.AddNewFill()
hybridShapeFill2.AddBound reference270
hybridShapeFill2.AddBound reference272
hybridShapeFill2.Continuity = 0
body1.InsertHybridShape hybridShapeFill2
part1.InWorkObject = hybridShapeFill2
part1.Update
.....
.....
Dim reference275 As Reference
Set reference275 = part1.CreateReferenceFromObject(hybridShapeLoft1)
Dim reference276 As Reference
Set reference276 = part1.CreateReferenceFromObject(hybridShapeLoft2)
Dim hybridShapeAssemble1 As HybridShapeAssemble
Set hybridShapeAssemble1 = hybridShapeFactory1.AddNewJoin(reference275,
reference276)
Dim reference277 As Reference
Set reference277 = part1.CreateReferenceFromObject(hybridShapeFill1)
hybridShapeAssemble1.AddElement reference277
Dim reference278 As Reference
Set reference278 = part1.CreateReferenceFromObject(hybridShapeFill2)
hybridShapeAssemble1.AddElement reference278
hybridShapeAssemble1.SetConnex 1
hybridShapeAssemble1.SetManifold 0
hybridShapeAssemble1.SetSimplify 0
hybridShapeAssemble1.SetSuppressMode 0
hybridShapeAssemble1.SetDeviation 0.001
hybridShapeAssemble1.SetAngularToleranceMode 0
hybridShapeAssemble1.SetAngularTolerance 0.5
hybridShapeAssemble1.SetFederationPropagation 0
body1.InsertHybridShape hybridShapeAssemble1
part1.InWorkObject = hybridShapeAssemble1
part1.Update
.....
.....

```

```

"          HIDE SKETCHES & LINES          "
.....
.....

Dim selection1 As Selection
Set selection1 = partDocument1.Selection
Dim visPropertySet1 As VisPropertySet
Set visPropertySet1 = selection1.VisProperties
Set sketches1 = sketch1.Parent
Dim bSTR1 As String
bSTR1 = sketch1.Name
selection1.Add sketch1
Set visPropertySet1 = visPropertySet1.Parent
Dim bSTR2 As String
bSTR2 = visPropertySet1.Name
Dim bSTR3 As String
bSTR3 = visPropertySet1.Name
visPropertySet1.SetShow 1
selection1.Clear
Dim selection2 As Selection
Set selection2 = partDocument1.Selection
Dim visPropertySet2 As VisPropertySet
Set visPropertySet2 = selection2.VisProperties
Set hybridShapes1 = hybridShapePlaneOffset1.Parent
Dim bSTR4 As String
bSTR4 = hybridShapePlaneOffset1.Name
selection2.Add hybridShapePlaneOffset1
Set visPropertySet2 = visPropertySet2.Parent
Dim bSTR5 As String
bSTR5 = visPropertySet2.Name
Dim bSTR6 As String
bSTR6 = visPropertySet2.Name
visPropertySet2.SetShow 1
selection2.Clear
Dim selection3 As Selection
Set selection3 = partDocument1.Selection
Dim visPropertySet3 As VisPropertySet
Set visPropertySet3 = selection3.VisProperties
Set sketches1 = sketch2.Parent
Dim bSTR7 As String
bSTR7 = sketch2.Name
selection3.Add sketch2
Set visPropertySet3 = visPropertySet3.Parent
Dim bSTR8 As String
bSTR8 = visPropertySet3.Name
Dim bSTR9 As String
bSTR9 = visPropertySet3.Name
visPropertySet3.SetShow 1
selection3.Clear
Dim selection4 As Selection
Set selection4 = partDocument1.Selection
Dim visPropertySet4 As VisPropertySet
Set visPropertySet4 = selection4.VisProperties
Set hybridShapes1 = hybridShapeLinePtPt1.Parent
Dim bSTR10 As String
bSTR10 = hybridShapeLinePtPt1.Name
selection4.Add hybridShapeLinePtPt1
Set visPropertySet4 = visPropertySet4.Parent
Dim bSTR11 As String
bSTR11 = visPropertySet4.Name
Dim bSTR12 As String
bSTR12 = visPropertySet4.Name
visPropertySet4.SetShow 1
selection4.Clear
.....
.....
"          SIMETRIA SUPERCIE          "
.....
.....

Dim reference279 As Reference
Set          reference279          =
part1.CreateReferenceFromObject(hybridShapeAssemble1)
Dim hybridShapePlaneExplicit2 As HybridShapePlaneExplicit
Set hybridShapePlaneExplicit2 = originElements1.PlaneYZ
Dim reference280 As Reference
Set          reference280          =
part1.CreateReferenceFromObject(hybridShapePlaneExplicit2
)
Dim hybridShapeSymmetry1 As HybridShapeSymmetry
Set          hybridShapeSymmetry1          =
hybridShapeFactory1.AddNewSymmetry(reference279,
reference280)
hybridShapeSymmetry1.VolumeResult = False
body1.InsertHybridShape hybridShapeSymmetry1
part1.InWorkObject = hybridShapeSymmetry1
.....
.....

Dim selection5 As Selection
Set selection5 = partDocument1.Selection
Dim visPropertySet5 As VisPropertySet

```

```
Set visPropertySet5 = selection5.VisProperties
Set hybridShapes1 = hybridShapeAssemble1.Parent
Dim bSTR13 As String
bSTR13 = hybridShapeAssemble1.Name
selection5.Add hybridShapeAssemble1
Set visPropertySet5 = visPropertySet5.Parent
Dim bSTR14 As String
bSTR14 = visPropertySet5.Name
Dim bSTR15 As String
bSTR15 = visPropertySet5.Name
```

```
visPropertySet5.SetShow 1
selection5.Clear
.....
" ACTUALIZAMOS Y CERRAMOS EL DOCUMENTO "
.....
'Se actualiza el documento y se cierra la interface
CATIA.ActiveDocument.Part.Update
Unload Me
```

ANEXO 6. CÓDIGO ALAS SUPERFICIES

```
.....
" EXTRUSIÓN DEL ALA (SUPERFICIE) "
.....
Dim hybridShapeLoft1 As HybridShapeLoft
Set hybridShapeLoft1 = hybridShapeFactory1.AddNewLoft()
hybridShapeLoft1.SectionCoupling = 1
hybridShapeLoft1.Relimitation = 1
hybridShapeLoft1.CanonicalDetection = 2
Dim reference268 As Reference
Set reference268 = part1.CreateReferenceFromObject(hybridShapeLinePt1)
hybridShapeLoft1.AddGuide reference268
Dim reference269 As Reference
Set reference269 = part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;142);None:(Limits1:());Limits2:());Cf11:());WithP
ermanentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch1)
hybridShapeLoft1.AddSectionToLoft reference269, 1,
Nothing
Dim reference270 As Reference
Set reference270 = part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.2;1);None:(Limits1:());Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch2)
hybridShapeLoft1.AddSectionToLoft reference270, 1,
Nothing
body1.InsertHybridShape hybridShapeLoft1
part1.InWorkObject = hybridShapeLoft1
```

```
part1.Update
.....
Dim hybridShapeLoft2 As HybridShapeLoft
Set hybridShapeLoft2 = hybridShapeFactory1.AddNewLoft()
hybridShapeLoft2.SectionCoupling = 1
hybridShapeLoft2.Relimitation = 1
hybridShapeLoft2.CanonicalDetection = 2
Dim reference271 As Reference
Set reference271 = part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;1);None:(Limits1:());Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch1)
hybridShapeLoft2.AddSectionToLoft reference271, 1,
Nothing
Dim reference272 As Reference
Set reference272 = part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.2;2);None:(Limits1:());Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch2)
hybridShapeLoft2.AddSectionToLoft reference272, 1,
Nothing
body1.InsertHybridShape hybridShapeLoft2
part1.InWorkObject = hybridShapeLoft2
part1.Update
CATIA.ActiveDocument.Part.Update
.....
" FILL DE LOS EXTREMOS Y JOIN (SUPERFICIE) "
.....
```

```

.....
Dim hybridShapeFill1 As HybridShapeFill
Set hybridShapeFill1 = hybridShapeFactory1.AddNewFill()
Dim reference273 As Reference
Set reference273 =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;1);None:(Limits1:();Limits2:());Cf11:());WithP
ermanentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRRepVersion_CXR15)", sketch1)
hybridShapeFill1.AddBound reference273
Dim reference274 As Reference
Set reference274 =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;142);None:(Limits1:();Limits2:());Cf11:());WithP
ermanentBody;WithoutBuildError;WithSelectingFeatureSupp
ort;MFBRRepVersion_CXR15)", sketch1)
hybridShapeFill1.AddBound reference274
hybridShapeFill1.Continuity = 0
body1.InsertHybridShape hybridShapeFill1
part1.InWorkObject = hybridShapeFill1
part1.Update
.....
.....
Dim hybridShapeFill2 As HybridShapeFill
Set hybridShapeFill2 = hybridShapeFactory1.AddNewFill()
hybridShapeFill2.AddBound reference270
hybridShapeFill2.AddBound reference272
hybridShapeFill2.Continuity = 0
body1.InsertHybridShape hybridShapeFill2
part1.InWorkObject = hybridShapeFill2
part1.Update
.....
.....
Dim reference275 As Reference
Set reference275 =
part1.CreateReferenceFromObject(hybridShapeLoft1)
Dim reference276 As Reference
Set reference276 =
part1.CreateReferenceFromObject(hybridShapeLoft2)
Dim hybridShapeAssemble1 As HybridShapeAssemble
Set hybridShapeAssemble1 =
hybridShapeFactory1.AddNewJoin(reference275,
reference276)
Dim reference277 As Reference
Set reference277 =
part1.CreateReferenceFromObject(hybridShapeFill1)
hybridShapeAssemble1.AddElement reference277
Dim reference278 As Reference
Set reference278 =
part1.CreateReferenceFromObject(hybridShapeFill2)
hybridShapeAssemble1.AddElement reference278
hybridShapeAssemble1.SetConnex 1
hybridShapeAssemble1.SetManifold 0
hybridShapeAssemble1.SetSimplify 0
hybridShapeAssemble1.SetSuppressMode 0
hybridShapeAssemble1.SetDeviation 0.001
hybridShapeAssemble1.SetAngularToleranceMode 0
hybridShapeAssemble1.SetAngularTolerance 0.5
hybridShapeAssemble1.SetFederationPropagation 0
body1.InsertHybridShape hybridShapeAssemble1
part1.InWorkObject = hybridShapeAssemble1
part1.Update
.....
.....
" HIDE SKETCHES & LINES "
.....
.....
Dim selection1 As Selection
Set selection1 = partDocument1.Selection
Dim visPropertySet1 As VisPropertySet
Set visPropertySet1 = selection1.VisProperties
Set sketches1 = sketch1.Parent
Dim bSTR1 As String
bSTR1 = sketch1.Name
selection1.Add sketch1
Set visPropertySet1 = visPropertySet1.Parent
Dim bSTR2 As String
bSTR2 = visPropertySet1.Name
Dim bSTR3 As String
bSTR3 = visPropertySet1.Name
visPropertySet1.SetShow 1
selection1.Clear
Dim selection2 As Selection
Set selection2 = partDocument1.Selection
Dim visPropertySet2 As VisPropertySet
Set visPropertySet2 = selection2.VisProperties
Set hybridShapes1 = hybridShapePlaneOffset1.Parent
Dim bSTR4 As String
bSTR4 = hybridShapePlaneOffset1.Name
selection2.Add hybridShapePlaneOffset1
Set visPropertySet2 = visPropertySet2.Parent

```

```

Dim bSTR5 As String
bSTR5 = visPropertySet2.Name
Dim bSTR6 As String
bSTR6 = visPropertySet2.Name
visPropertySet2.SetShow 1
selection2.Clear
Dim selection3 As Selection
Set selection3 = partDocument1.Selection
Dim visPropertySet3 As VisPropertySet
Set visPropertySet3 = selection3.VisProperties
Set sketches1 = sketch2.Parent
Dim bSTR7 As String
bSTR7 = sketch2.Name
selection3.Add sketch2
Set visPropertySet3 = visPropertySet3.Parent
Dim bSTR8 As String
bSTR8 = visPropertySet3.Name
Dim bSTR9 As String
bSTR9 = visPropertySet3.Name
visPropertySet3.SetShow 1
selection3.Clear
Dim selection4 As Selection
Set selection4 = partDocument1.Selection
Dim visPropertySet4 As VisPropertySet
Set visPropertySet4 = selection4.VisProperties
Set hybridShapes1 = hybridShapeLinePtPt1.Parent
Dim bSTR10 As String
bSTR10 = hybridShapeLinePtPt1.Name
selection4.Add hybridShapeLinePtPt1
Set visPropertySet4 = visPropertySet4.Parent
Dim bSTR11 As String
bSTR11 = visPropertySet4.Name
Dim bSTR12 As String
bSTR12 = visPropertySet4.Name
visPropertySet4.SetShow 1
selection4.Clear
.....
"          SIMETRIA SUPERCIE          "
.....
Dim reference279 As Reference
Set reference279 = part1.CreateReferenceFromObject(hybridShapeAssemble1)
Dim hybridShapePlaneExplicit2 As HybridShapePlaneExplicit
Set hybridShapePlaneExplicit2 = originElements1.Plane YZ
Dim reference280 As Reference
Set reference280 = part1.CreateReferenceFromObject(hybridShapePlaneExplicit2)
Dim hybridShapeSymmetry1 As HybridShapeSymmetry
Set hybridShapeSymmetry1 = hybridShapeFactory1.AddNewSymmetry(reference279, reference280)
hybridShapeSymmetry1.VolumeResult = False
body1.InsertHybridShape hybridShapeSymmetry1
part1.InWorkObject = hybridShapeSymmetry1
.....
Dim reference281 As Reference
Set reference281 = part1.CreateReferenceFromObject(hybridShapeAssemble1)
Dim reference282 As Reference
Set reference282 = part1.CreateReferenceFromObject(hybridShapeSymmetry1)
Dim hybridShapeAssemble2 As HybridShapeAssemble
Set hybridShapeAssemble2 = hybridShapeFactory1.AddNewJoin(reference281, reference282)
hybridShapeAssemble2.SetConnex 0
hybridShapeAssemble2.SetManifold 0
hybridShapeAssemble2.SetSimplify 0
hybridShapeAssemble2.SetSuppressMode 0
hybridShapeAssemble2.SetDeviation 0.001
hybridShapeAssemble2.SetAngularToleranceMode 0
hybridShapeAssemble2.SetAngularTolerance 0.5
hybridShapeAssemble2.SetFederationPropagation 0
body1.InsertHybridShape hybridShapeAssemble2
part1.InWorkObject = hybridShapeAssemble2
.....
" ACTUALIZAMOS Y CERRAMOS EL DOCUMENTO "
.....
'Se actualiza el documento y se cierra la interface
CATIA.ActiveDocument.Part.Update
Unload Me
Dim reference279 As Reference

```

ANEXO 7. CÓDIGO ALA IZQUIERDA SÓLIDA

```

.....
.....
"   EXTRUSIÓN DEL ALA (SUPERFICIE)   "
.....
.....

Dim hybridShapeLoft1 As HybridShapeLoft
Set hybridShapeLoft1 = hybridShapeFactory1.AddNewLoft()
hybridShapeLoft1.SectionCoupling = 1
hybridShapeLoft1.Relimitation = 1
hybridShapeLoft1.CanonicalDetection = 2
Dim reference268 As Reference
Set          reference268          =
part1.CreateReferenceFromObject(hybridShapeLinePtPt1)

hybridShapeLoft1.AddGuide reference268

Dim reference269 As Reference
Set          reference269          =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;142);None:(Limits1:();Limits2:());Cf11:());WithP
ermanentBody;WithoutBuildError;WithSelectingFeatureSupp
ort;MFBRepVersion_CXR15)", sketch1)
hybridShapeLoft1.AddSectionToLoft   reference269,   1,
Nothing

Dim reference270 As Reference
Set          reference270          =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.2;1);None:(Limits1:();Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch2)
hybridShapeLoft1.AddSectionToLoft   reference270,   1,
Nothing
body1.InsertHybridShape hybridShapeLoft1
part1.InWorkObject = hybridShapeLoft1
part1.Update
.....
.....

Dim hybridShapeLoft2 As HybridShapeLoft
Set hybridShapeLoft2 = hybridShapeFactory1.AddNewLoft()
hybridShapeLoft2.SectionCoupling = 1
hybridShapeLoft2.Relimitation = 1
hybridShapeLoft2.CanonicalDetection = 2

Dim reference271 As Reference
Set          reference271          =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;1);None:(Limits1:();Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch1)
hybridShapeLoft2.AddSectionToLoft   reference271,   1,
Nothing

Dim reference272 As Reference
Set          reference272          =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.2;2);None:(Limits1:();Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch2)
hybridShapeLoft2.AddSectionToLoft   reference272,   1,
Nothing
body1.InsertHybridShape hybridShapeLoft2
part1.InWorkObject = hybridShapeLoft2
part1.Update
CATIA.ActiveDocument.Part.Update
.....
.....
"   FILL DE LOS EXTREMOS Y JOIN (SUPERFICIE)   "
.....
.....

Dim hybridShapeFill1 As HybridShapeFill
Set hybridShapeFill1 = hybridShapeFactory1.AddNewFill()
Dim reference273 As Reference
Set          reference273          =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;1);None:(Limits1:();Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch1)
hybridShapeFill1.AddBound reference273

Dim reference274 As Reference
Set          reference274          =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;142);None:(Limits1:();Limits2:());Cf11:());WithP
ermanentBody;WithoutBuildError;WithSelectingFeatureSupp
ort;MFBRepVersion_CXR15)", sketch1)
hybridShapeFill1.AddBound reference274
hybridShapeFill1.Continuity = 0
body1.InsertHybridShape hybridShapeFill1
part1.InWorkObject = hybridShapeFill1

```



```

part1.Update
" HIDE SKETCHES & LINES "
.....
.....

Dim hybridShapeFill2 As HybridShapeFill
Dim selection1 As Selection
Set hybridShapeFill2 = hybridShapeFactory1.AddNewFill()
Set selection1 = partDocument1.Selection
hybridShapeFill2.AddBound reference270
Dim visPropertySet1 As VisPropertySet
hybridShapeFill2.AddBound reference272
Set visPropertySet1 = selection1.VisProperties
hybridShapeFill2.Continuity = 0
Set sketches1 = sketch1.Parent
body1.InsertHybridShape hybridShapeFill2
Dim bSTR1 As String
part1.InWorkObject = hybridShapeFill2
bSTR1 = sketch1.Name
part1.Update
selection1.Add sketch1
.....
Set visPropertySet1 = visPropertySet1.Parent
.....
Dim bSTR2 As String
bSTR2 = visPropertySet1.Name
Dim bSTR3 As String
bSTR3 = visPropertySet1.Name
Set reference275 As Reference
bSTR3 = visPropertySet1.Name
Set reference275 = part1.CreateReferenceFromObject(hybridShapeLoft1)
visPropertySet1.SetShow 1
Dim reference276 As Reference
selection1.Clear
Set reference276 = part1.CreateReferenceFromObject(hybridShapeLoft2)
Dim selection2 As Selection
Set selection2 = partDocument1.Selection
Dim hybridShapeAssemble1 As HybridShapeAssemble
Dim visPropertySet2 As VisPropertySet
Set hybridShapeAssemble1 = hybridShapeFactory1.AddNewJoin(reference275,
reference276)
Set visPropertySet2 = selection2.VisProperties
Set hybridShapes1 = hybridShapePlaneOffset1.Parent
Dim reference277 As Reference
Dim bSTR4 As String
Set reference277 = part1.CreateReferenceFromObject(hybridShapeFill1)
bSTR4 = hybridShapePlaneOffset1.Name
hybridShapeAssemble1.AddElement reference277
selection2.Add hybridShapePlaneOffset1
Set visPropertySet2 = visPropertySet2.Parent
Dim reference278 As Reference
Dim bSTR5 As String
Set reference278 = part1.CreateReferenceFromObject(hybridShapeFill2)
bSTR5 = visPropertySet2.Name
hybridShapeAssemble1.AddElement reference278
Dim bSTR6 As String
hybridShapeAssemble1.SetConnex 1
bSTR6 = visPropertySet2.Name
hybridShapeAssemble1.SetManifold 0
visPropertySet2.SetShow 1
hybridShapeAssemble1.SetSimplify 0
selection2.Clear
hybridShapeAssemble1.SetSuppressMode 0
Dim selection3 As Selection
hybridShapeAssemble1.SetDeviation 0.001
Set selection3 = partDocument1.Selection
hybridShapeAssemble1.SetAngularToleranceMode 0
Dim visPropertySet3 As VisPropertySet
hybridShapeAssemble1.SetAngularTolerance 0.5
Set visPropertySet3 = selection3.VisProperties
hybridShapeAssemble1.SetFederationPropagation 0
Set sketches1 = sketch2.Parent
body1.InsertHybridShape hybridShapeAssemble1
Dim bSTR7 As String
part1.InWorkObject = hybridShapeAssemble1
bSTR7 = sketch2.Name
part1.Update
selection3.Add sketch2
.....
Set visPropertySet3 = visPropertySet3.Parent
.....
Dim bSTR8 As String

```

```

bSTR8 = visPropertySet3.Name
Dim bSTR9 As String
bSTR9 = visPropertySet3.Name
visPropertySet3.SetShow 1
selection3.Clear
Dim selection4 As Selection
Set selection4 = partDocument1.Selection
Dim visPropertySet4 As VisPropertySet
Set visPropertySet4 = selection4.VisProperties
Set hybridShapes1 = hybridShapeLinePtPt1.Parent
Dim bSTR10 As String
bSTR10 = hybridShapeLinePtPt1.Name
selection4.Add hybridShapeLinePtPt1
Set visPropertySet4 = visPropertySet4.Parent
Dim bSTR11 As String
bSTR11 = visPropertySet4.Name
Dim bSTR12 As String
bSTR12 = visPropertySet4.Name
visPropertySet4.SetShow 1
selection4.Clear
.....
"          ALA SOLIDA          "
.....
Dim shapeFactory1 As ShapeFactory
Set shapeFactory1 = part1.ShapeFactory
Dim reference279 As Reference
Set reference279 = part1.CreateReferenceFromName("")
Dim closeSurface1 As CloseSurface
Set          closeSurface1          =
shapeFactory1.AddNewCloseSurface(reference279)

```

```

Dim reference280 As Reference
Set          reference280          =
part1.CreateReferenceFromObject(hybridShapeAssemble1)
closeSurface1.Surface = reference280
part1.UpdateObject closeSurface1
.....
Dim selection5 As Selection
Set selection5 = partDocument1.Selection
Dim visPropertySet5 As VisPropertySet
Set visPropertySet5 = selection5.VisProperties
Set hybridShapes1 = hybridShapeAssemble1.Parent
Dim bSTR13 As String
bSTR13 = hybridShapeAssemble1.Name
selection5.Add hybridShapeAssemble1
Set visPropertySet5 = visPropertySet5.Parent
Dim bSTR14 As String
bSTR14 = visPropertySet5.Name
Dim bSTR15 As String
bSTR15 = visPropertySet5.Name
visPropertySet5.SetShow 1
selection5.Clear
.....
"ACTUALIZAMOS Y CERRAMOS EL DOCUMENTO "
.....
'Se actualiza el documento y se cierra la interface
CATIA.ActiveDocument.Part.Update
Unload Me

```

ANEXO 8. CÓDIGO ALA DERECHA SÓLIDA

```

.....
"  EXTRUSIÓN DEL ALA (SUPERFICIE)  "
.....
Dim hybridShapeLoft1 As HybridShapeLoft
Set hybridShapeLoft1 = hybridShapeFactory1.AddNewLoft()

```

```

hybridShapeLoft1.SectionCoupling = 1
hybridShapeLoft1.Relimitation = 1
hybridShapeLoft1.CanonicalDetection = 2
Dim reference268 As Reference
Set          reference268          =
part1.CreateReferenceFromObject(hybridShapeLinePtPt1)
hybridShapeLoft1.AddGuide reference268

```

```

Dim reference269 As Reference

Set          reference269          =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;142);None:(Limits1:();Limits2:());Cf11:());WithP
ermanentBody;WithoutBuildError;WithSelectingFeatureSupp
ort;MFBRepVersion_CXR15)", sketch1)

hybridShapeLoft1.AddSectionToLoft  reference269,  1,
Nothing

Dim reference270 As Reference

Set          reference270          =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.2;1);None:(Limits1:();Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch2)

hybridShapeLoft1.AddSectionToLoft  reference270,  1,
Nothing

body1.InsertHybridShape hybridShapeLoft1

part1.InWorkObject = hybridShapeLoft1

part1.Update
.....
.....

Dim hybridShapeLoft2 As HybridShapeLoft

Set hybridShapeLoft2 = hybridShapeFactory1.AddNewLoft()

hybridShapeLoft2.SectionCoupling = 1

hybridShapeLoft2.Relimitation = 1

hybridShapeLoft2.CanonicalDetection = 2

Dim reference271 As Reference

Set          reference271          =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;1);None:(Limits1:();Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch1)

hybridShapeLoft2.AddSectionToLoft  reference271,  1,
Nothing

Dim reference272 As Reference

Set          reference272          =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.2;2);None:(Limits1:();Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch2)

hybridShapeLoft2.AddSectionToLoft  reference272,  1,
Nothing

body1.InsertHybridShape hybridShapeLoft2

part1.InWorkObject = hybridShapeLoft2

part1.Update

CATIA.ActiveDocument.Part.Update
.....
.....

"   FILL DE LOS EXTREMOS Y JOIN (SUPERFICIE)   "
.....

```

```

.....

Dim hybridShapeFill1 As HybridShapeFill

Set hybridShapeFill1 = hybridShapeFactory1.AddNewFill()

Dim reference273 As Reference

Set          reference273          =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;1);None:(Limits1:();Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch1)

hybridShapeFill1.AddBound reference273

Dim reference274 As Reference

Set          reference274          =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;142);None:(Limits1:();Limits2:());Cf11:());WithP
ermanentBody;WithoutBuildError;WithSelectingFeatureSupp
ort;MFBRepVersion_CXR15)", sketch1)

hybridShapeFill1.AddBound reference274

hybridShapeFill1.Continuity = 0

body1.InsertHybridShape hybridShapeFill1

part1.InWorkObject = hybridShapeFill1

part1.Update
.....
.....

Dim hybridShapeFill2 As HybridShapeFill

Set hybridShapeFill2 = hybridShapeFactory1.AddNewFill()

hybridShapeFill2.AddBound reference270

hybridShapeFill2.AddBound reference272

hybridShapeFill2.Continuity = 0

body1.InsertHybridShape hybridShapeFill2

part1.InWorkObject = hybridShapeFill2

part1.Update
.....
.....

Dim reference275 As Reference

Set          reference275          =
part1.CreateReferenceFromObject(hybridShapeLoft1)

Dim reference276 As Reference

Set          reference276          =
part1.CreateReferenceFromObject(hybridShapeLoft2)

Dim hybridShapeAssemble1 As HybridShapeAssemble

Set          hybridShapeAssemble1  =
hybridShapeFactory1.AddNewJoin(reference275,
reference276)

Dim reference277 As Reference

Set          reference277          =
part1.CreateReferenceFromObject(hybridShapeFill1)

hybridShapeAssemble1.AddElement reference277

Dim reference278 As Reference

```

```

Set reference278 = Dim bSTR5 As String
part1.CreateReferenceFromObject(hybridShapeFill2) bSTR5 = visPropertySet2.Name
hybridShapeAssemble1.AddElement reference278 Dim bSTR6 As String
hybridShapeAssemble1.SetConnex 1 bSTR6 = visPropertySet2.Name
hybridShapeAssemble1.SetManifold 0 visPropertySet2.SetShow 1
hybridShapeAssemble1.SetSimplify 0 selection2.Clear
hybridShapeAssemble1.SetSuppressMode 0 Dim selection3 As Selection
hybridShapeAssemble1.SetDeviation 0.001 Set selection3 = partDocument1.Selection
hybridShapeAssemble1.SetAngularToleranceMode 0 Dim visPropertySet3 As VisPropertySet
hybridShapeAssemble1.SetAngularTolerance 0.5 Set visPropertySet3 = selection3.VisProperties
hybridShapeAssemble1.SetFederationPropagation 0 Set sketches1 = sketch2.Parent
body1.InsertHybridShape hybridShapeAssemble1 Dim bSTR7 As String
part1.InWorkObject = hybridShapeAssemble1 bSTR7 = sketch2.Name
part1.Update selection3.Add sketch2
..... Set visPropertySet3 = visPropertySet3.Parent
..... Dim bSTR8 As String
" HIDE SKETCHES & LINES " bSTR8 = visPropertySet3.Name
..... Dim bSTR9 As String
..... bSTR9 = visPropertySet3.Name
Dim selection1 As Selection visPropertySet3.SetShow 1
Set selection1 = partDocument1.Selection selection3.Clear
Dim visPropertySet1 As VisPropertySet Dim selection4 As Selection
Set visPropertySet1 = selection1.VisProperties Set selection4 = partDocument1.Selection
Set sketches1 = sketch1.Parent Dim visPropertySet4 As VisPropertySet
Dim bSTR1 As String Set visPropertySet4 = selection4.VisProperties
bSTR1 = sketch1.Name Set hybridShapes1 = hybridShapeLinePtPt1.Parent
selection1.Add sketch1 Dim bSTR10 As String
Set visPropertySet1 = visPropertySet1.Parent bSTR10 = hybridShapeLinePtPt1.Name
Dim bSTR2 As String selection4.Add hybridShapeLinePtPt1
bSTR2 = visPropertySet1.Name Set visPropertySet4 = visPropertySet4.Parent
Dim bSTR3 As String Dim bSTR11 As String
bSTR3 = visPropertySet1.Name bSTR11 = visPropertySet4.Name
visPropertySet1.SetShow 1 Dim bSTR12 As String
selection1.Clear bSTR12 = visPropertySet4.Name
Dim selection2 As Selection visPropertySet4.SetShow 1
Set selection2 = partDocument1.Selection selection4.Clear
Dim visPropertySet2 As VisPropertySet
.....
.....
Set visPropertySet2 = selection2.VisProperties
Set hybridShapes1 = hybridShapePlaneOffset1.Parent
Dim bSTR4 As String
bSTR4 = hybridShapePlaneOffset1.Name
selection2.Add hybridShapePlaneOffset1
Set visPropertySet2 = visPropertySet2.Parent
.....
.....
Dim shapeFactory1 As ShapeFactory
Set shapeFactory1 = part1.ShapeFactory

```

```

Dim reference279 As Reference
Set reference279 = part1.CreateReferenceFromName("")

Dim closeSurface1 As CloseSurface
Set closeSurface1 = shapeFactory1.AddNewCloseSurface(reference279)

Dim reference280 As Reference
Set reference280 = part1.CreateReferenceFromObject(hybridShapeAssemble1)
closeSurface1.Surface = reference280
part1.UpdateObject closeSurface1

Dim selection5 As Selection
Set selection5 = partDocument1.Selection

Dim visPropertySet5 As VisPropertySet
Set visPropertySet5 = selection5.VisProperties
Set hybridShapes1 = hybridShapeAssemble1.Parent

Dim bSTR13 As String
bSTR13 = hybridShapeAssemble1.Name
selection5.Add hybridShapeAssemble1
Set visPropertySet5 = visPropertySet5.Parent

Dim bSTR14 As String
bSTR14 = visPropertySet5.Name

Dim bSTR15 As String
bSTR15 = visPropertySet5.Name
visPropertySet5.SetShow 1

selection5.Clear

"          SIMETRÍA          "

Dim hybridShapePlaneExplicit2 As HybridShapePlaneExplicit
Set hybridShapePlaneExplicit2 = originElements1.Plane YZ

Dim reference281 As Reference
Set reference281 = part1.CreateReferenceFromObject(hybridShapePlaneExplicit2)

Dim symmetry1 As Symmetry
Set symmetry1 = shapeFactory1.AddNewSymmetry2(reference281)

Dim hybridShapeSymmetry1 As HybridShapeSymmetry
Set hybridShapeSymmetry1 = symmetry1.HybridShape

"          ACTUALIZAMOS Y CERRAMOS EL DOCUMENTO          "

'Se actualiza el documento y se cierra la interface
CATIA.ActiveDocument.Part.Update
Unload Me

```

ANEXO 9. CÓDIGO ALAS SÓLIDAS

```

"          EXTRUSIÓN DEL ALA (SUPERFICIE)          "

Dim hybridShapeLoft1 As HybridShapeLoft
Set hybridShapeLoft1 = hybridShapeFactory1.AddNewLoft()
hybridShapeLoft1.SectionCoupling = 1
hybridShapeLoft1.Relimitation = 1
hybridShapeLoft1.CanonicalDetection = 2

Dim reference268 As Reference
Set reference268 = part1.CreateReferenceFromObject(hybridShapeLinePt1)
hybridShapeLoft1.AddGuide reference268

Dim reference269 As Reference
Set reference269 = part1.CreateReferenceFromBRepName("WireREdge:(Wire:(BRep:(Sketch.1;142);None:(Limits1:();Limits2:());Cf11:());WithPermanentBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", sketch1)
hybridShapeLoft1.AddSectionToLoft reference269, 1, Nothing

Dim reference270 As Reference
Set reference270 = part1.CreateReferenceFromBRepName("WireREdge:(Wire:(BRep:(Sketch.2;1);None:(Limits1:();Limits2:());Cf11:());WithPermanentBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", sketch2)
hybridShapeLoft1.AddSectionToLoft reference270, 1, Nothing
body1.InsertHybridShape hybridShapeLoft1

```

```

part1.InWorkObject = hybridShapeLoft1
part1.Update
.....
.....

Dim hybridShapeLoft2 As HybridShapeLoft
Set hybridShapeLoft2 = hybridShapeFactory1.AddNewLoft()
hybridShapeLoft2.SectionCoupling = 1
hybridShapeLoft2.Relimitation = 1
hybridShapeLoft2.CanonicalDetection = 2

Dim reference271 As Reference

Set reference271 =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;1);None:(Limits1:();Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch1)

hybridShapeLoft2.AddSectionToLoft reference271, 1,
Nothing

Dim reference272 As Reference

Set reference272 =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.2;2);None:(Limits1:();Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch2)

hybridShapeLoft2.AddSectionToLoft reference272, 1,
Nothing

body1.InsertHybridShape hybridShapeLoft2
part1.InWorkObject = hybridShapeLoft2
part1.Update
CATIA.ActiveDocument.Part.Update
.....
.....

" FILL DE LOS EXTREMOS Y JOIN (SUPERFICIE) "
.....
.....

Dim hybridShapeFill1 As HybridShapeFill
Set hybridShapeFill1 = hybridShapeFactory1.AddNewFill()
Dim reference273 As Reference

Set reference273 =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;1);None:(Limits1:();Limits2:());Cf11:());WithPer
manentBody;WithoutBuildError;WithSelectingFeatureSupport
;MFBRepVersion_CXR15)", sketch1)

hybridShapeFill1.AddBound reference273

Dim reference274 As Reference

Set reference274 =
part1.CreateReferenceFromBRepName("WireREdge:(Wire:(B
rp:(Sketch.1;142);None:(Limits1:();Limits2:());Cf11:());WithP
ermanentBody;WithoutBuildError;WithSelectingFeatureSupp
ort;MFBRepVersion_CXR15)", sketch1)

hybridShapeFill1.AddBound reference274

hybridShapeFill1.Continuity = 0
body1.InsertHybridShape hybridShapeFill1
part1.InWorkObject = hybridShapeFill1
part1.Update
.....
.....

Dim hybridShapeFill2 As HybridShapeFill
Set hybridShapeFill2 = hybridShapeFactory1.AddNewFill()
hybridShapeFill2.AddBound reference270
hybridShapeFill2.AddBound reference272
hybridShapeFill2.Continuity = 0
body1.InsertHybridShape hybridShapeFill2
part1.InWorkObject = hybridShapeFill2
part1.Update
.....
.....

Dim reference275 As Reference
Set reference275 =
part1.CreateReferenceFromObject(hybridShapeLoft1)
Dim reference276 As Reference
Set reference276 =
part1.CreateReferenceFromObject(hybridShapeLoft2)
Dim hybridShapeAssemble1 As HybridShapeAssemble
Set hybridShapeAssemble1 =
hybridShapeFactory1.AddNewJoin(reference275,
reference276)
Dim reference277 As Reference
Set reference277 =
part1.CreateReferenceFromObject(hybridShapeFill1)
hybridShapeAssemble1.AddElement reference277
Dim reference278 As Reference
Set reference278 =
part1.CreateReferenceFromObject(hybridShapeFill2)
hybridShapeAssemble1.AddElement reference278
hybridShapeAssemble1.SetConnex 1
hybridShapeAssemble1.SetManifold 0
hybridShapeAssemble1.SetSimplify 0
hybridShapeAssemble1.SetSuppressMode 0
hybridShapeAssemble1.SetDeviation 0.001
hybridShapeAssemble1.SetAngularToleranceMode 0
hybridShapeAssemble1.SetAngularTolerance 0.5
hybridShapeAssemble1.SetFederationPropagation 0
body1.InsertHybridShape hybridShapeAssemble1
part1.InWorkObject = hybridShapeAssemble1
part1.Update

```

```

.....
.....
"      HIDE SKETCHES & LINES      "
.....
.....
Dim selection1 As Selection
Set selection1 = partDocument1.Selection
Dim visPropertySet1 As VisPropertySet
Set visPropertySet1 = selection1.VisProperties
Set sketches1 = sketch1.Parent
Dim bSTR1 As String
bSTR1 = sketch1.Name
selection1.Add sketch1
Set visPropertySet1 = visPropertySet1.Parent
Dim bSTR2 As String
bSTR2 = visPropertySet1.Name
Dim bSTR3 As String
bSTR3 = visPropertySet1.Name
visPropertySet1.SetShow 1
selection1.Clear
Dim selection2 As Selection
Set selection2 = partDocument1.Selection
Dim visPropertySet2 As VisPropertySet
Set visPropertySet2 = selection2.VisProperties
Set hybridShapes1 = hybridShapePlaneOffset1.Parent
Dim bSTR4 As String
bSTR4 = hybridShapePlaneOffset1.Name
selection2.Add hybridShapePlaneOffset1
Set visPropertySet2 = visPropertySet2.Parent
Dim bSTR5 As String
bSTR5 = visPropertySet2.Name
Dim bSTR6 As String
bSTR6 = visPropertySet2.Name
visPropertySet2.SetShow 1
selection2.Clear
Dim selection3 As Selection
Set selection3 = partDocument1.Selection
Dim visPropertySet3 As VisPropertySet
Set visPropertySet3 = selection3.VisProperties
Set sketches1 = sketch2.Parent
Dim bSTR7 As String
bSTR7 = sketch2.Name
selection3.Add sketch2

Set visPropertySet3 = visPropertySet3.Parent
Dim bSTR8 As String
bSTR8 = visPropertySet3.Name
Dim bSTR9 As String
bSTR9 = visPropertySet3.Name
visPropertySet3.SetShow 1
selection3.Clear
Dim selection4 As Selection
Set selection4 = partDocument1.Selection
Dim visPropertySet4 As VisPropertySet
Set visPropertySet4 = selection4.VisProperties
Set hybridShapes1 = hybridShapeLinePtPt1.Parent
Dim bSTR10 As String
bSTR10 = hybridShapeLinePtPt1.Name
selection4.Add hybridShapeLinePtPt1
Set visPropertySet4 = visPropertySet4.Parent
Dim bSTR11 As String
bSTR11 = visPropertySet4.Name
Dim bSTR12 As String
bSTR12 = visPropertySet4.Name
visPropertySet4.SetShow 1
selection4.Clear
.....
.....
"      ALA SOLIDA      "
.....
.....
Dim shapeFactory1 As ShapeFactory
Set shapeFactory1 = part1.ShapeFactory
Dim reference279 As Reference
Set reference279 = part1.CreateReferenceFromName("")
Dim closeSurface1 As CloseSurface
Set closeSurface1 = shapeFactory1.AddNewCloseSurface(reference279)
Dim reference280 As Reference
Set reference280 = part1.CreateReferenceFromObject(hybridShapeAssemble1)
closeSurface1.Surface = reference280
part1.UpdateObject closeSurface1
.....
.....
Dim selection5 As Selection
Set selection5 = partDocument1.Selection
Dim visPropertySet5 As VisPropertySet

```

```
Set visPropertySet5 = selection5.VisProperties
```

```
Set hybridShapes1 = hybridShapeAssemble1.Parent
```

```
Dim bSTR13 As String
```

```
bSTR13 = hybridShapeAssemble1.Name
```

```
selection5.Add hybridShapeAssemble1
```

```
Set visPropertySet5 = visPropertySet5.Parent
```

```
Dim bSTR14 As String
```

```
bSTR14 = visPropertySet5.Name
```

```
Dim bSTR15 As String
```

```
bSTR15 = visPropertySet5.Name
```

```
visPropertySet5.SetShow 1
```

```
selection5.Clear
```

```
.....
```

```
.....
```

```
" MIRROR "
```

```
.....
```

```
.....
```

```
Dim hybridShapePlaneExplicit2 As HybridShapePlaneExplicit
```

```
Set hybridShapePlaneExplicit2 = originElements1.PlaneYZ
```

```
Dim reference281 As Reference
```

```
Set reference281 = part1.CreateReferenceFromObject(hybridShapePlaneExplicit2)
```

```
Dim mirror1 As Mirror
```

```
Set mirror1 = shapeFactory1.AddNewMirror(reference281)
```

```
.....
```

```
.....
```

```
" ACTUALIZAMOS Y CERRAMOS EL DOCUMENTO "
```

```
.....
```

```
.....
```

```
' Se actualiza el documento y se cierra la interface
```

```
CATIA.ActiveDocument.Part.Update
```

```
Unload me
```