

Trabajo Fin de Grado

Ingeniería Aeroespacial

Generación de las mediciones y el presupuesto en un entorno de CAD 3D

Autor: José Luis Lucas Curado

Tutor: Francisco Hernández Rodríguez

**Dpto. Ingeniería de la Construcción y Proyectos
de Ingeniería**

Escuela Técnica Superior de Ingeniería

Sevilla, 2019



Trabajo Fin de Grado
Ingeniería Aeroespacial

Generación de las mediciones y el presupuesto en un entorno de CAD 3D

Autor:

José Luis Lucas Curado

Tutor:

Francisco Hernández Rodríguez

Profesor titular

Dpto. de Ingeniería de la Construcción y Proyectos de Ingeniería

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2019

Trabajo Fin de Grado: Generación de las mediciones y el presupuesto en un entorno de CAD 3D

Autor: José Luis Lucas Curado

Tutor: Francisco Hernández Rodríguez

El tribunal nombrado para juzgar el Trabajo Fin de Grado arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2019

El Secretario del Tribunal

A mis padres

A mis hermanas

Agradecimientos

Quisiera agradecer a todas las personas que de una manera u otra me han ayudado a superar esta difícil, pero fructífera etapa de mi vida. A los compañeros y compañeras con los que he compartido tantísimas horas de trabajos y estudios. A mi familia y amigos, que me han animado y apoyado en todo este camino. Y, por supuesto, a mi pareja, la persona que ha sabido alegrarse por mí en las buenas y soportarme en las malas.

También me gustaría agradecer a Don Francisco Hernández Rodríguez por su inestimable colaboración en el desarrollo de este proyecto.

José Luis Lucas Curado

Sevilla, 2019

Resumen

En este trabajo fin de grado se desarrolla un código de programación que sirve de ayuda para la generación de las mediciones y el presupuesto de un proyecto.

Para poder usar esta aplicación es necesaria la definición del producto en el entorno de CAD 3D (CATIA) y la disposición de una base de datos relacional (Access) en la que se incluyen los precios unitarios de los distintos elementos que componen el producto. A partir de aquí, se genera un archivo en formato FIEBDC que es directamente importable por programas comerciales de mediciones y presupuesto, como por ejemplo Presto o Arquímedes, en los que se elabora la presentación del presupuesto correspondiente.

Abstract

In this final degree project, a programming code is developed that helps to generate the measurements and the budget of a project.

In order to use this application, it is necessary to define the product in the 3D CAD environment (CATIA) and the provision of a relational database (Access) that includes the unit prices of the different elements that make up the product. From here, an FIEBDC format file is generated that is directly imported by commercial measurement and budget programs, such as Presto or Archimedes, in which the corresponding budget presentation is prepared.

Índice

Agradecimientos	9
Resumen	11
Abstract	13
Índice	15
Índice de Tablas	17
Índice de Figuras	19
1 Introducción.	21
1.1 <i>Objetivo.</i>	23
1.2 <i>Arquitectura del Sistema.</i>	23
1.3 <i>Organización de la documentación.</i>	24
2 Implicaciones en el entorno de CAD 3D.	25
2.1 <i>Propiedades de los elementos.</i>	25
2.2 <i>Definición de propiedades.</i>	25
2.3 <i>Elementos geométricos.</i>	26
3 Implicaciones en el entorno de BDR.	29
4 Formato FIEBDC. Generación del archivo BC3.	32
4.1 <i>Constitución del formato FIEBDC.</i>	32
4.2 <i>Estructura del formato FIEBDC.</i>	32
4.3 <i>Tipos de registros.</i>	33
5 Aplicación VBA.	35
5.1 <i>Visual Basic en CATIA.</i>	35
5.2 <i>Aplicación desarrollada en VBA</i>	38
6 Ejemplo de ensamblaje: Balancín.	45
6.1 <i>Elementos del ensamblaje: Medición.</i>	45
6.1.1 <i>Subconjunto.</i>	46
6.1.2 <i>Subconjunto 2.</i>	51
6.1.3 <i>Eje.</i>	53
6.1.4 <i>Mango.</i>	54
6.1.5 <i>Remache.</i>	54
6.2 <i>Organización del ensamblaje en capítulos.</i>	55
6.2.1 <i>Capítulo 1: Estructura principal.</i>	55
6.2.2 <i>Capítulo 2: Uniones.</i>	55
6.2.3 <i>Capítulo 3: Elementos de seguridad.</i>	55
6.3 <i>Tablas de presupuesto de comprobación.</i>	56
6.4 <i>Archivo bc3 generado.</i>	56
6.5 <i>Importación del archivo bc3 desde Arquímedes.</i>	58
Anexo Código	62
Bibliografía	80

ÍNDICE DE TABLAS

Tabla 3-1: Base de datos propia	30
Tabla 3-2: Base de Costes de la Construcción de Andalucía	31
Tabla 5-1: Ordenación de códigos	40
Tabla 5-2: Ordenación de capítulos	41
Tabla 5-3: Unidades en Sistema Internacional	41
Tabla 6-1: Mediciones y presupuesto tras modificaciones	56
Tabla 6-2: Presupuesto rendimientos tras modificación	56

ÍNDICE DE FIGURAS

Figura 1-1. Esquema del sistema empleado	23
Figura 2-1. Propiedades de un cuerpo.	26
Figura 2-2. Define other properties.	26
Figura 2-3. Definición de la longitud de un eje por puntos.	27
Figura 3-6. Plancha de anclaje.	28
Figura 5-1. Ventana para acceder a las macros.	35
Figura 5-2. Ventana para la creación de un directorio.	36
Figura 5-3. Interfaz superior-izquierda de Visual Basic.	36
Figura 5-4. Ventana de reporte de error en VBA.	37
Figura 5-5. Línea de error del programa.	37
Figura 5-6. Ventana inmediata Visual Basic.	37
Figura 5-7. Diagrama de flujo de la aplicación.	39
Figura 5-8. Geometría abierta	42
Figura 5-9. Geometría cerrada	42
Figura 6-1. Ensamblaje balancín.	45
Figura 6-2. Estructura árbol ensamblaje	46
Figura 6-3. Subconjunto.	47
Figura 6-4. Cuadro de diálogo. Profundización en elemento no medible.	47
Figura 6-5. Soporte.	48
Figura 6-6. Cuadro de diálogo. Volumen soporte.	48
Figura 6-7. Cuadro de diálogo. Masa soporte.	49
Figura 6-8. Base.	49
Figura 6-9. Cuadro de diálogo. Área mayor de la base.	50
Figura 7-10. Plancha de anclaje.	50
Figura 7-11. Cuadro de diálogo. Área superior plancha de anclaje.	50
Figura 7-12. Estructura árbol plancha de anclaje.	51
Figura 6-13. Subconjunto 2.	52
Figura 6-14. Balancín.	52
Figura 6-15. Cuadro de diálogo. Longitud balancín.	52
Figura 6-16. Asiento.	53
Figura 6-17. Amortiguador.	53
Figura 6-18. Cuadro de diálogo. Longitud eje.	54
Figura 6-19. Mango.	54
Figura 6-20. Remache.	55
Figura 6-21. Disposición del ensamblaje en Arquímedes.	58

Figura 6-22. Medición capítulo 1.	59
Figura 6-23. Medición capítulo 2.	59
Figura 6-24. Medición capítulo 3.	59
Figura 6-25. Presupuesto capítulo 1.	60
Figura 6-20. Presupuesto capítulo 2.	60
Figura 6-20. Presupuesto capítulo 3.	60
Figura 6-20. Presupuesto ejecución material total.	61

1 INTRODUCCIÓN.

Un proyecto de ingeniería está formado por un conjunto de escritos, cálculos y dibujos que tienen como objetivo dar idea de cómo tiene que ser y cuánto tiene que costar una obra de ingeniería.

La norma UNE 157001:2014 define un proyecto de ingeniería como “conjunto de documentos, modelos o maquetas, en soporte físico, lógico o cualquier otro, que tiene como objeto la definición y la valoración de las características de un producto, obra, edificio, instalación y servicio, y que se requieren en función de su fin o destino”.

Los documentos que componen un proyecto de ingeniería acorde a la norma UNE 157001: 2014 son los siguientes:

- 1) Índice
- 2) Memoria
- 3) Anexo
- 4) Planos
- 5) Pliego de condiciones
- 6) Mediciones
- 7) Presupuesto

Cada documento tiene una información diferente que aportar al proyecto de ingeniería, la cual se resume a continuación.

1) Índice:

Tiene la función de cualquier índice de un documento, que no es otra que la de facilitar al lector la búsqueda de la información específica del documento que le interese.

2) Memoria:

Tiene como objetivo describir y justificar las soluciones tomadas.

En la memoria se responde a 7 preguntas esenciales:

- Qué proyecto se va a realizar.
- Por qué se va a realizar. Es decir, se describe el problema que motiva la creación del proyecto.
- Para qué se va a realizar. Dicho de otra manera, en qué mejorará el problema planteado.
- Dónde se va a realizar, en qué lugar.
- Cómo se va a proceder a la ejecución del proyecto, qué forma de resolución se va a adoptar.
- Cuándo se va a realizar, en qué fechas se va a desarrollar cada parte del proyecto.
- Cuánto va a costar la realización del proyecto.

3) Anexo:

Recoge documentación que ha sido necesaria para la ejecución del proyecto pero que no forma parte del objetivo del proyecto como tal. Otra forma de verlo es que en el anexo no se responde a ninguna de las preguntas planteadas anteriormente en la memoria.

4) Planos:

Su objetivo es describir gráficamente el objeto del proyecto. Se diferencian los siguientes tipos de planos:

- Planos generales:
 - Planos de situación: muestran un entorno amplio que engloba municipios y vías de comunicación.
 - Planos de localización: muestran el entorno próximo al lugar de realización del proyecto, con edificios y parcelas colindantes.
 - Planos de replanteo: permiten situar la obra en el terreno, indicando los ejes y las cotas del terreno de referencia respecto a un punto invariable y no removible (origen).
 - Planos de distribución general: detallan los accesos al edificio y su estructuración interna.
- Planos de detalle: deben permitir la fabricación y/o construcción de los elementos descritos.

5) Pliego de condiciones:

Establece la forma de ejecución del proyecto atendiendo a condicionantes de tipo técnico, legal, económico y administrativo.

6) Mediciones:

Define y cuantifica las unidades de obra. Contiene el cuadro de mediciones.

Una primera parte del contenido de este documento consta de una lista de unidades de obra necesarias para la ejecución del proyecto.

La segunda del parte trata de determinar la cantidad de cada una de las unidades de obra.

Tradicionalmente la medición se realizaba a partir de planos en formato papel, lo cual es una labor tediosa y lenta, además de propensa a errores, ya que interviene el hombre.

Posteriormente se comenzó a realizar la medición sobre planos digitales 2D. La medición se hacía directamente desde el plano realizado a ordenador, de manera que se mejoraba la exactitud de las mediciones. Sin embargo, el factor humano seguía interviniendo considerablemente en el dibujo y estructuración de esos planos a ordenador, lo cual era una tarea que seguía requiriendo un esfuerzo considerable.

Finalmente, la tecnología nos ha permitido realizar modelos en 3D, con herramientas como CATIA, que facilita considerablemente el modelaje del objeto del proyecto en cuestión. Además, da la posibilidad de automatizar la medición a través de herramientas como Visual Basic conectadas con CATIA. Esta posibilidad es la que se ha explorado en el desarrollo del trabajo.

7) Presupuesto:

Tiene como objetivo informar sobre el coste total de todas las unidades de obras, siendo el resultado del producto del cuadro de mediciones por el cuadro de precios, que es donde aparecen los costes unitarios de cada unidad de obra.

Según la definición de la Real Academia Española (RAE), un presupuesto es el “cómputo anticipado del coste de una obra o de los gastos y rentas de una corporación”. El objetivo de este trabajo está más orientado a la primera parte de la definición, ya que se busca cuantificar el coste de las unidades de obras empleadas en un proyecto.

Por tanto, una definición más certera sería que un presupuesto de obra es la “previsión total de los costes involucrados en una obra de construcción incrementados con el margen de beneficio que se tenga previsto”.

Todo presupuesto debe contener la siguiente información:

- Unidades de obras que intervengan en el proyecto.
- Mediciones de cada unidad de obra.
- Precio unitario de cada unidad de obra.
- Resultado del precio unitario de cada unidad por su respectiva medición.

En definitiva, el presupuesto se basa en las mediciones realizadas, por lo que, si somos capaces de automatizar las mediciones, no hay más que añadir el factor proporcionado por el cuadro de precios para automatizar igualmente los costes totales obtenidos, es decir, el presupuesto.

1.1 Objetivo.

A la vista de lo comentado anteriormente, este trabajo plantea una metodología para generar las mediciones y el presupuesto de un producto a partir del modelo 3D del mismo.

1.2 Arquitectura del Sistema.

El Sistema desarrollado requiere de distintas fuentes para obtener el resultado deseado. Así, para una concepción más clara sobre las conexiones realizadas entre las distintas fuentes se presenta la siguiente figura:

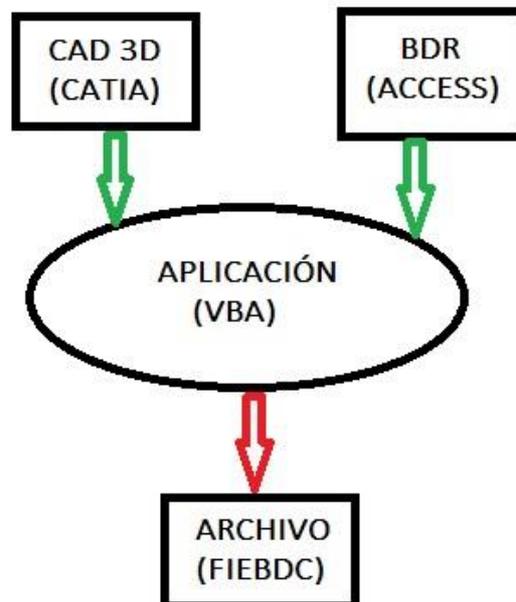


Figura 1-1. Esquema del sistema empleado

El producto que se pretende medir y presupuestar se encuentra definido en 3D en el entorno de CAD 3D, en este caso CATIA V5 Dassault Systèmes. En este entorno el producto se describe en términos de sus componentes.

En la Base de Datos Relacional (BDR) se encuentran registrados los precios unitarios de los distintos elementos del producto que se pretende medir. Por ejemplo, el precio del kilogramo de acero, el precio del metro cuadrado de pintura, el precio del metro de un cable, el precio de la colocación de una puerta, etc.

Con objeto de relacionar los elementos del modelo 3D con los precios unitarios de sus mediciones, se le asigna un código a cada registro de la base de datos.

La aplicación desarrollada recorre la estructura árbol del modelo 3D recuperando el código de cada uno de los elementos a medir. A partir de dicho código, accede a la base de datos y recupera el registro correspondiente al propio elemento (magnitud a medir, precio unitario, nombre del elemento, descripción del elemento, etc).

Una vez conocida la magnitud a medir, la aplicación realiza su medición y calcula su precio correspondiente.

Tras recorrer todos los elementos del modelo 3D, genera un archivo de texto con formato FIEBDC (Formato de Intercambio Estándar de Bases de Datos de la Construcción), que puede importarse por un programa comercial desde el que generar el documento final de mediciones y presupuesto.

1.3 Organización de la documentación.

En base a la arquitectura propuesta, en primer lugar se describirán las implicaciones que ésta tiene en el entorno de CAD 3D.

En segundo lugar, se comentarán los campos existentes en la base de datos, así como la utilidad de cada uno.

Posteriormente, se describirán el origen y las características del formato FIEBDC, comentando sus diferentes tipos de registros y ejemplificándolos.

A continuación, se realizará un acercamiento a la herramienta Visual Basic, a través de la cual se han desarrollado las líneas de código expuestas en el anexo. Tras esta primera toma de contacto con Visual Basic se abordará la lógica del funcionamiento de la aplicación propuesta.

Por último, se aplicará el código desarrollado a un caso concreto, comprobando así su correcto funcionamiento. Además, se mostrará el resultado obtenido de importar el archivo de texto generado con extensión bc3 en un programa de mediciones y presupuesto.

2 IMPLICACIONES EN EL ENTORNO DE CAD 3D.

En primer lugar, antes de realizar una medición, se necesita saber qué objeto físico se va a medir. Es decir, se requiere identificar a los elementos medibles.

Y, en segundo lugar, se debe introducir en CATIA V5 una serie de parámetros que servirán de guiado a la aplicación. Dichos parámetros indicarán por ejemplo qué magnitud física se debe medir. Esto es imprescindible, puesto que no sirve de nada que se obtenga el valor exacto de una magnitud física si no es esa la que queremos medir.

2.1 Propiedades de los elementos.

Código

Para identificar cada elemento se le va a proporcionar un código único a cada uno de ellos.

Este código deberá estar estructurado de manera que las 2 primeras cifras indiquen el capítulo al que pertenecen. Con esto conseguimos una mejor organización posterior a la hora de generar el presupuesto. Por tanto, con esta estructuración del código se contempla un máximo de 99 capítulos en un la base de datos.

El código identificativo de cada elemento se definirá en CATIA como una propiedad del elemento de tipo "String", y también estará registrado en una base de datos en Access.

Podrá tomarse una base de datos que pertenezca a algún sector en concreto, como puede ser la Base de Costes de la Construcción de Andalucía, o puede ser cualquiera propia que siga las características expuestas en el apartado 3.

En la estructura árbol del diseño en CATIA del proyecto, habrá elementos que no sean directamente medibles a los que se les definirá un código nulo. Literalmente es necesario definir esta propiedad de manera que la propiedad se llame "CÓDIGO", en mayúsculas y con tilde, y el valor que se le de sea "NULO", en mayúsculas también.

Cuando la aplicación obtenga este código nulo, profundizará en el elemento y realizará una medición de los componentes que lo formen. Así sucesivamente hasta dar con un elemento medible.

Densidad

Será necesaria añadir esta propiedad al elemento en cuestión cuando la magnitud física que se espere medir de él sea la masa. Para el resto de magnitudes la densidad no interviene, por lo que no es necesario definirla.

La densidad se añadirá como una propiedad de tipo "Density", y se definirá como "DENSIDAD", dándole el valor numérico correspondiente en cada caso.

2.2 Definición de propiedades.

Seguidamente se va a explicar el procedimiento que hay que seguir para añadir dichas propiedades y el tipo de cada una.

En primer lugar, se clica con el botón derecho sobre el nombre del cuerpo y se selecciona "Properties".

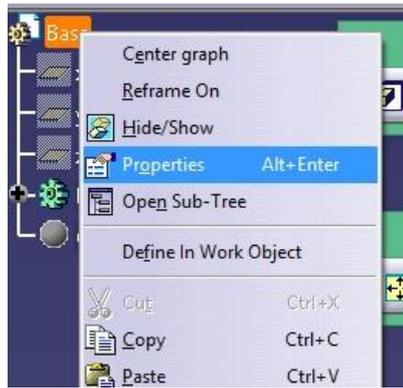


Figura 2-1. Propiedades de un cuerpo

En segundo lugar, en la parte baja de la pestaña “Product” de la ventana emergente, se selecciona “Define other properties”.

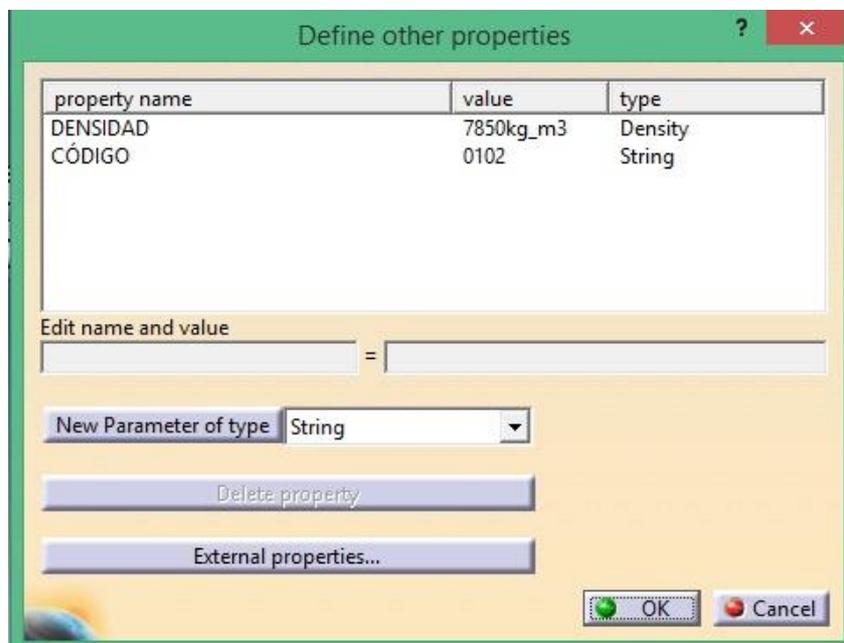


Figura 2-2. Define other properties

Se puede observar en la figura 2-2 la opción de añadir el tipo de parámetro. Si se despliega la pestaña aparecen multitud de tipos, entre los que destacan “String” y “Density” en los parámetros ya definidos para este ejemplo.

Por último, se clicca sobre “New parameter of type”, se introduce el nombre del parámetro a la izquierda del igual, el valor del parámetro a la derecha.

2.3 Elementos geométricos.

Determinados cuerpos requieren del uso de algunos elementos geométricos que nos ayude a realizar la medición. Por ejemplo, si queremos medir la longitud de un cuerpo, necesitamos saber a qué longitud nos referimos.

A continuación, se expone el caso de un eje. Para conocer su longitud se van a definir un par de puntos, uno en cada extremo del eje, permitiéndonos calcular la distancia entre ambos puntos que resultará ser la longitud del eje.

La estructura del nombre de los puntos que definan una longitud estará formada por una primera parte que permanece fija, “PuntoLong”. Una segunda parte que será el nombre del elemento, por ejemplo “Eje”. Y, una tercera que acabará indicando el número del punto, “.x”. Siendo “x” un número entero positivo tal que 1, 2, 3, ...

En la *figura 2-3* se puede apreciar cómo se han definido un par de puntos, uno en cada extremo del eje. Están marcados con un aspa blanca y resaltados por flechas rojas para mejorar su visualización.

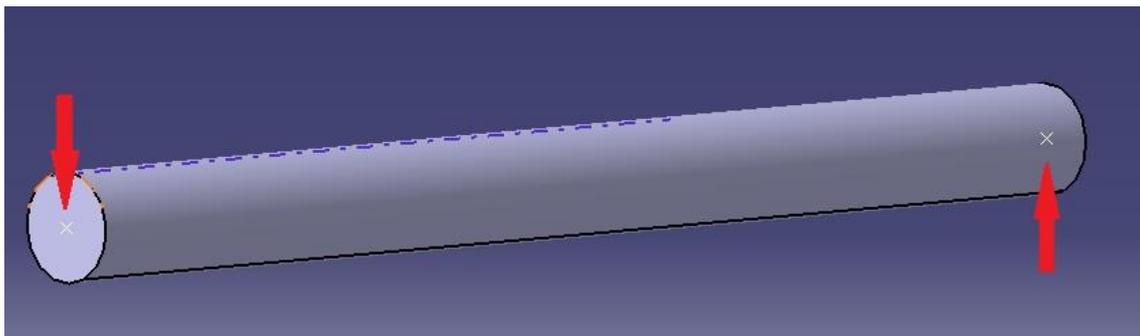


Figura 2-3. Definición de la longitud de un eje por puntos

Por tanto, la nomenclatura correcta para estos puntos sería: “PuntoLongEje.1” y “PuntoLongEje.2”.

Por otro lado, también hay que indicar si la longitud que queremos medir es abierta o cerrada, ya que el resultado variará de en función del tipo de longitud que se indique.

Esto sucede porque la medida de la longitud se obtiene a partir de la distancia entre las coordenadas de los puntos creados. Así, si se tienen únicamente dos puntos, bastará con calcular la distancia que hay del primero al segundo. Sin embargo, si se tienen tres puntos, se pueden dar dos casos: que los puntos definan una geometría abierta o cerrada.

Al igual que pasa con la medición de una longitud, un elemento tiene varias superficies o caras sujetas de ser medidas. Por tanto, se necesitará indicar mediante la introducción de nuevos parámetros qué superficie queremos medir.

Se han contemplado dos posibilidades:

- I) Medir el área de la cara de mayor superficie del cuerpo.
- II) Medir una cara cualquiera

La primera opción es muy interesante para objetos en los que una cara sea muy superior al resto, como pueda ser una chapa de metálica. Para este supuesto no es necesario añadir ningún parámetro adicional, ya que se ha implementado unas líneas de código donde se calculan las áreas de todas las caras y da como resultado la mayor.

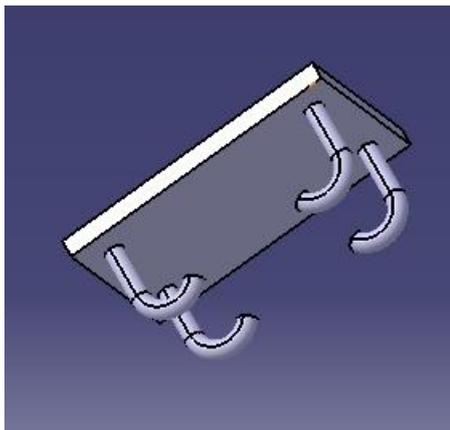


Figura 2-4. Plancha de anclaje

Como ejemplo representativo del objetivo de esta primera opción tenemos la plancha de anclaje. La superficie de mayor interés es la que ofrece la toma de contacto con la infraestructura que se va a posicionar sobre ella.

En definitiva, aplicando la opción de medida de área I), conseguimos obviar las áreas de las patas de anclaje y bordes de la plancha, para obtener la que nos interesa.

En contraposición, para la segunda opción sí se vuelve a necesitar de la definición de unos puntos que delimiten nuestra área a calcular.

La nomenclatura de los puntos que se definen para esta opción debe seguir siempre la misma estructura. Una primera parte formada por “PuntoArea”. Sucedida por una segunda parte con el nombre del elemento que se está midiendo, “PlanchaAnclaje”. Y, finalizando con el número del punto que se ha definido.

Es decir, si se definen varios puntos para la plancha de anclaje, los puntos se nombrarían como: “PuntoAreaPlanchaAnclaje.1”, “PuntoAreaPlanchaAnclaje.2”, etc.

Hay que prestar especial atención al nombre que se le da al elemento, ya que, si nombramos al elemento con espacio, el nombre del punto también deberá respetar ese espacio. Por ejemplo, si el elemento se llama “Plancha Anclaje”, los puntos deberán llamarse “PuntoAreaPlancha Anclaje.1”, “PuntoAreaPlancha Anclaje.2”, etc.

Una puntualización importante es que con la primera opción tenemos la posibilidad de medir cualquier tipo de área, ya sea plana o curva. Sin embargo, para el área definida por puntos, solo podemos calcular áreas planas.

Si se definen muchos puntos muy cerca unos de otros, se conseguiría una aproximación muy buena a la superficie con curvatura.

Esta teoría es conocida como la teoría del cálculo infinitesimal. Sin embargo, el estudio de esta teoría no es objetivo de este trabajo.

3 IMPLICACIONES EN EL ENTORNO DE BDR.

La base de datos relacional (BDR) es una tabla donde se recogen todas las unidades de obra empleadas en un proyecto, acompañada con información imprescindible como el coste unitario, el código, el nombre del elemento o una pequeña descripción.

Dicha base de datos puede ser por ejemplo la Base de Costes de la Construcción de la Junta de Andalucía (BCCA). Sin embargo, también se puede crear una Base de Datos propia, como se ha hecho para el ejemplo expuesto en el apartado 6.

En la base de datos mostrada en la *tabla 3-1* se observan seis campos distintos estructurados en columnas.

CÓDIGO	UNIDAD	PRECIO_UNITARIO	DESCRIPCION	RESUMEN	FECHA
0101	m2	250,00 €	Base de hormigón que sirve para dotar de estabilidad a la estructu Base		18/07/2019
0102	kg	10,00 €	Soporte de acero unido a la base y sobre el que descansa el mec	Soporte	18/07/2019
0103	m	500,00 €	Barra de acero que se somete a momentos de balanceo	Balancin	18/07/2019
0104	u	150,00 €	Asiento de acero soldado a la barra de acero del balancin	Asiento	18/07/2019
0201	m	100,00 €	Eje de acero que dota de libertad de balanceo a la estructura	Eje	18/07/2019
0202	u	0,10 €	Remache	Remache	18/07/2019
0301	u	50,00 €	Mango	Mango	18/07/2019
0203	m2	200,00 €	Plancha de acero para el anclaje del soporte a la base	Plancha anclaje	18/07/2019
01				Estructura principal	31/07/2019
02				Uniones	31/07/2019
03				Elementos de seguridad	31/07/2019
0302	u	20,00 €	Apoyo de goma de seguridad para reducir la fuerza del impacto c	Apoyo de goma	31/07/2019

Tabla 3-1: Base de datos propia

Si miramos de izquierda a derecha, el primer campo se trata del “CÓDIGO”. Se ha ideado un código de cuatro cifras en el que las dos de la izquierda indican el capítulo en el que se enmarca dicha unidad de obra. Las dos cifras de la derecha indican el lugar de la unidad de obra dentro del capítulo al que pertenecen.

Los códigos que están compuestos solo por dos cifras hacen referencia al nombre de los capítulos. De ahí que no dispongan de un valor para “UNIDAD”, “PRECIO_UNITARIO” o “DESCRIPCION”.

El segundo campo es el de “UNIDAD”. Aquí aparecen los valores “u”, “m”, “m2”, “m3” y puede aparecer también “kg”:

“u”: indica que se pretende obtener la cantidad de unidades empleada de dicho elemento en la construcción.

“m”: indica que se pretende obtener la longitud del elemento en cuestión.

“m2”: indica que el objetivo es conocer el área del elemento.

“m3”: indica que el objetivo es obtener el volumen del elemento.

“kg”: indica que el objetivo es obtener la masa del elemento.

El tercer campo es el “PRECIO_UNITARIO”. Se trata del factor que relaciona la medición con el presupuesto, ya que cuantifica el coste económico del proyecto a partir de la cantidad de materiales empleados.

Cabe resaltar que los precios unitarios que aparecen en la base de datos no tienen ninguna veracidad,

y están expuestos únicamente a modo de ejemplo.

El siguiente campo es la “DESCRIPCIÓN”. Contiene un breve comentario sobre en qué consiste la unidad de obra que se expone. Añadiendo algunos detalles como el material o la función que desempeña en el proyecto.

Le sigue el campo “RESUMEN”. Consiste básicamente en el nombre dado a cada unidad de obra o elemento.

Por último, se muestra el campo “FECHA”. En este caso, al tratarse de una base de datos propia, la fecha hace referencia al día en que se añadió la unidad de obra pertinente al proyecto.

Para una mejor comprensión se plantean un par de ejemplos sencillos. Para los siguientes códigos se deduce que:

0103
└───┘
Capítulo 1 | Unidad de obra 3

0301
└───┘
Capítulo 3 | Unidad de obra 1

Esta numeración va a servir para ordenar los códigos y los capítulos de forma que cuando se genere el presupuesto se muestre un orden lógico de capítulos y unidades de obra.

Si nos encontráramos ante un proyecto de construcción podría utilizarse una base de datos existente sin necesidad de crear una base de datos específica. Por ejemplo, si se ejecutara un proyecto de edificación en Andalucía, se podría recurrir a la Base de Costes de la Construcción de Andalucía (BCCA). Como se puede comprobar en la figura mostrada a continuación, el formato de la base de datos es similar a la creada para nuestro ejemplo.

Código	U	Resumen	Fecha	PrPr	Texto1
07		CUBIERTAS	05/08/2002	0	
07H		HORIZONTALES	05/08/2002	0	
07HA		Ajardinadas	05/08/2002	0	
07HAE00001	m	ENC.FALDON SOBRE TABIQUILLOS CON PARAMEN	05/08/2002	22.88	DE ENCUENTRO DE FALDON PARA AJARDINAR CON PARAMENTOS INCLUSO FORMACION Y RELLENO DE ROZAS DE 5X5 cm, REFUERZOS
07HAE00002	u	ENCUNTRO FALDON PARA AJARDINAR CON SUMI	05/08/2002	53.18	DE ENCUENTRO DE FALDON PARA AJARDINAR CON SUMIDERO, INCLUSO CAJA PARA RECIBIR LA CAZOLETA FORMADA POR LADRILLO HUECO
07HAF00001	m2	FALDON DE AZOTEA PARA AJARDINAR	05/08/2002	35.63	DE FALDON DE AZOTEA PARA AJARDINAR FORMADO POR BARRERA DE VAPOR DE BASE ASFALTICA, CAPA DE HORMIGON ALIGERADO DE
07HAW00001	m	JUNTA DE DILATACION EN FALDON PARA AJARDIN	05/08/2002	29.43	DE JUNTA DE DILATACION EN FALDON PARA AJARDINAR FORMADA POR MAESTRAS DE TABICON DE LADRILLO HUECO, PLANCHA DE PLO
07HN		No transitables	05/08/2002	0	
07HNE00001	m	ENC.FALDON SOBRE TABIQUILLOS CON PARAMEN	05/08/2002	16.58	DE ENCUENTRO DE FALDON SOBRE TABIQUILLOS CON PARAMENTOS INCLUSO ZOCALO DE LADRILLO Y REFUERZO CON MEMBRANA DE
07HNE00002	m	ENC.FALDON S/HORMIG.ALIG. CON PARAMENTO,S	05/08/2002	8.76	DE ENCUENTRO DE FALDON SOBRE HORMIGON ALIGERADO CON PARAMENTOS, INCLUSO JUNTA ELASTICA, FORMACION Y RELLENO DE
07HNE00003	u	ENC.FALDON DE GRAVILLA CON SUMIDERO,REF. N	05/08/2002	10.9	DE ENCUENTRO DE FALDON CON PROTECCION DE GRAVILLA CON SUMIDERO, INCLUSO MAESTRA DE TABICON DE LADRILLO HUECO Y R
07HNE00012	m	ENC.FALDON S/HORMIG.ALIG. CON PARAMENTO,S	05/08/2002	9.46	DE ENCUENTRO DE FALDON SOBRE HORMIGON ALIGERADO CON PARAMENTOS, INCLUSO JUNTA ELASTICA, FORMACION Y RELLENO DE
07HNE00013	u	ENC.FALDON DE GRAVILLA CON SUMIDERO,REF. U	05/08/2002	13.31	DE ENCUENTRO DE FALDON CON PROTECCION DE GRAVILLA CON SUMIDERO, INCLUSO MAESTRA DE TABICON DE LADRILLO HUECO Y R
07HNF00001	m2	FALDON DE AZOTEA NO TRANSITABLE,SOBR TABIQ	05/08/2002	33.22	DE FALDON DE AZOTEA NO TRANSITABLE FORMADO POR: TABIQUILLOS SEPARADOS 50 cm, CAPA DE AISLAMIENTO CON POLIESTIREN
07HNF00002	m2	FALDON AZ.NO TRANS.S/HORMIG.10CM,SUP.GRA	05/08/2002	26.53	DE FALDON DE AZOTEA NO TRANSITABLE FORMADO POR: BARRERA DE VAPOR DE BASE ASFALTICA, CAPA DE HORMIGON ALIGERADO D
07HNF00003	m2	FALDON AZ.NO TRANS.S/HORMIG.10CM,SUP.ALU	05/08/2002	26.36	DE FALDON DE AZOTEA NO TRANSITABLE, FORMADO POR: BARRERA DE VAPOR DE BASE ASFALTICA, CAPA DE HORMIGON ALIGERADO C
07HNF00004	m2	FALDON AZ.NO TRANS.S/HORMIG.10CM,SUP.GRA	05/08/2002	36.04	DE FALDON DE AZOTEA NO TRANSITABLE, FORMADO POR: BARRERA DE VAPOR DE BASE ASFALTICA, CAPA DE HORMIGON ALIGERADO C
07HNF00005	m2	FALDON AZ.NO TRANS.S/HORMIG.10CM,SUP.ALU	05/08/2002	24.54	DE FALDON DE AZOTEA NO TRANSITABLE, FORMADO POR: BARRERA DE VAPOR DE BASE ASFALTICA, CAPA DE HORMIGON ALIGERADO C
07HNF00012	m2	FALDON AZ.NO TRANS.S/HORMIG.15CM,SUP.GRA	05/08/2002	29.22	DE FALDON DE AZOTEA NO TRANSITABLE FORMADO POR: BARRERA DE VAPOR DE BASE ASFALTICA, CAPA DE HORMIGON ALIGERADO D
07HNF00013	m2	FALDON AZ.NO TRANS.S/HORMIG.15CM,SUP.ALU	05/08/2002	28.16	DE FALDON DE AZOTEA NO TRANSITABLE, FORMADO POR: BARRERA DE VAPOR DE BASE ASFALTICA, CAPA DE HORMIGON ALIGERADO C
07HNF00014	m2	FALDON AZ.NO TRANS.S/HORMIG.15CM,SUP.GRA	05/08/2002	38.61	DE FALDON DE AZOTEA NO TRANSITABLE, FORMADO POR: BARRERA DE VAPOR DE BASE ASFALTICA, CAPA DE HORMIGON ALIGERADO C
07HNF00015	m2	FALDON AZ.NO TRANS.S/HORMIG.15CM,SUP.ALU	05/08/2002	26.97	DE FALDON DE AZOTEA NO TRANSITABLE, FORMADO POR: BARRERA DE VAPOR DE BASE ASFALTICA, CAPA DE HORMIGON ALIGERADO C
07HNF00021	m2	FALDON AZ.NO TRANS. INVERTIDA	05/08/2002	28.63	DE FALDON DE AZOTEA INVERTIDA NO TRANSITABLE CONSTITUIDA POR: FORMACION DE PENDIENTE SUAVE CON MORTERO M-2 (1:8);
07HNF00051	m2	FALDON AZ.NO TRANS.S/HORMIG.15CM,SUP.PINT	05/08/2002	31.01	DE FALDON DE AZOTEA NO TRANSITABLE, FORMADO POR: BARRERA DE VAPOR DE BASE ASFALTICA, CAPA DE HORMIGON ALIGERADO C
07HNF00061	m2	FALDON AZ.NO TRANS.S/MORTERO M-20,SUP.ARI	05/08/2002	35.19	DE FALDON AZOTEA NO TRANSITABLE, FORMADO POR: FORMACION DE PENDIENTE CON MORTERO M-20 (1:8),FILTRO GEOTEXTIL SEE
07HNF00071	m2	FALDON AZ.NO TRANS.S/HORMIG M-20,AGLOM.	05/08/2002	36.08	DE FALDON DE AZOTEA NO TRANSITABLE, FORMADO POR: FORMACION PENDIENTE DE MORTERO M-20 (1:8), PANEL AISLANTE DE AGU
07HNW00001	m	BORDE LIBRE EN FALDON SOBRE TABIQUILLOS,CIN	05/08/2002	37.93	DE BORDE LIBRE EN FALDON SOBRE TABIQUILLOS, INCLUSO MAESTRA DE TABICON DE LADRILLO HUECO, RESALTO DE MORTERO DE CEI
07HNW00002	m	BORDE LIBRE EN FALDON SOBRE HORMIGON ALIG	05/08/2002	19.45	DE BORDE LIBRE EN FALDON SOBRE HORMIGON ALIGERADO INCLUSO MAESTRA DE TABICON DE LADRILLO HUECO, RESALTO DE MORTE
07HNW00003	m	JUNTA DE DILATACION FALDONES CON PROTECCI	05/08/2002	38.8	DE JUNTA DE DILATACION DE FALDONES CON PROTECCION DE GRAVILLA, INCLUSO MAESTRA DE TABICON DE LADRILLO HUECO, PLANCI
07HNW00010	m	REMATE DE FALDON DE AZOTEA INVERTIDA NO TR	05/08/2002	12.45	DE REMATE DE FALDON DE AZOTEA INVERTIDA NO TRANSITABLE FORMADO POR BLOQUE DE HORMIGON ARMADO EN FORMA DE L. D
07HT		Transitables	05/08/2002	0	
07HTE00001	m	ENC.FALDON S/HORMIG.ALIG. CON PARAMENTO,C	05/08/2002	13.6	DE ENCUENTRO DE FALDON SOBRE HORMIGON ALIGERADO CON PARAMENTOS, INCLUSO JUNTA ELASTICA, FORMACION Y RELLENO DE
07HTE00002	m	ENC.FALDON S/TABIQUILLOS CON PARAMENTO,C	05/08/2002	21.26	DE ENCUENTRO DE FALDON SOBRE TABIQUILLOS CON PARAMENTOS, INCLUSO ZOCALO DE LADRILLO, MEMBRANA DE BETUN MODIFIC
07HTE00003	u	ENC.FALDON CON CAZOLETA,REFUERZO MEMBRAI	05/08/2002	6.16	DE ENCUENTRO DE FALDON CON CAZOLETA, INCLUSO CAJA PARA RECIBIR LA CAZOLETA FORMADA CON LADRILLO HUECO Y REFUERZO I
07HTE00011	m	ENC.FALDON S/HORMIG.ALIG. CON PARAMENTO,F	05/08/2002	15.22	DE ENCUENTRO DE FALDON SOBRE HORMIGON ALIGERADO CON PARAMENTOS, INCLUSO JUNTA ELASTICA, FORMACION Y RELLENO DE
07HTE00013	u	ENC.FALDON CON CAZOLETA,REFUERZO LAMINA D	05/08/2002	12.66	DE ENCUENTRO DE FALDON CON CAZOLETA, INCLUSO CAJA PARA RECIBIR LA CAZOLETA FORMADA CON LADRILLO HUECO Y REFUERZO I
07HTF00001	m2	FALDON AZ.TRANS.S/HORMIG.10CM. 1 MEMB.BET	05/08/2002	38.16	DE FALDON DE AZOTEA TRANSITABLE FORMADO POR: BARRERA DE VAPOR DE BASE ASFALTICA, CAPA DE HORMIGON ALIGERADO DE 10
07HTF00002	m2	FALDON AZ.TRANS.S/TABIQUILLOS	05/08/2002	53.58	DE FALDON DE AZOTEA TRANSITABLE FORMADO POR: BARRERA DE VAPOR DE BASE ASFALTICA, TABIQUILLOS A 0.5M. CAPA DE POLIEST
07HTF00003	m2	FALDON AZ.TRANS.S/HORMIG.10CM. 2 MEMB.BET	05/08/2002	42.81	DE FALDON DE AZOTEA TRANSITABLE FORMADO POR:BARRERA DE VAPOR DE BASE ASFALTICA, CAPA DE HORMIGON ALIGERADO DE 10
07HTF00011	m2	FALDON AZ.TRANS.S/HORMIG.15CM. 1 MEMB.BET	05/08/2002	40.73	DE FALDON DE AZOTEA TRANSITABLE FORMADO POR: BARRERA DE VAPOR DE BASE ASFALTICA, CAPA DE HORMIGON ALIGERADO DE 15

Tabla 3-2: Base de Costes de la Construcción de Andalucía

Se observa que los códigos que aparecen en la *tabla 3-2* tienen una cadena de caracteres ampliamente superior, dado que el número unidades de obra que se tienen que almacenar e identificar es muy superior. Por tanto, habría que modificar la propiedad código añadida al elemento en CATIA para que coincida con la del concepto asociado a esta base de costes.

Sin embargo, en el campo unidad si se definen los mismos supuestos, lo que hace que el programa desarrollado en este trabajo sea compatible en ese aspecto con esta base de costes.

Para el resto de campos solo se necesitaría una ligera modificación en la subrutina con la que conseguimos conectar con la base de datos. Indicándole ahora el nombre del campo tal cual aparece en esta base de costes.

4 FORMATO FIEBDC. GENERACIÓN DEL ARCHIVO BC3.

El formato FIEBDC es el formato usado para el archivo final que se genera con las mediciones y el presupuesto del proyecto. Especifica qué datos deben aparecer en el presupuesto, así como la estructuración que se presenta de los mismos.

La ventaja de este formato es que es importable por otros programas utilizados frecuentemente para el desarrollo de un presupuesto, como por ejemplo PRESTO.

4.1 Constitución del formato FIEBDC.

FIEBDC se define desde su página web www.fiebdc.es como una “Asociación constituida para definir y supervisar el Formato de Intercambio Estándar de Bases de Datos para la Construcción”.

La idea de construir un formato intercambiable tuvo lugar en Julio de 1996, y fueron 17 las empresas que formaron parte desde un principio de este proyecto, sabedores de la importancia de una buena transferencia de datos entre ellas. Así se reducen tiempos y se aumenta en la rigurosidad del formato. En definitiva, se consigue aumentar la eficiencia del proceso.

4.2 Estructura del formato FIEBDC.

El formato FIEBDC es un archivo de texto, y se puede identificar fácilmente porque su extensión es “.bc3”.

Cada archivo está compuesto por registros, que son fragmentos de texto encabezados por el signo “~”. Los archivos deben contener registros completos, es decir, que un archivo siempre debe comenzar junto con el comienzo de un registro. Por tanto, al comienzo de un archivo, el primer carácter debe ser “~”, que indica el comienzo de registro.

Por otra parte, cada registro está compuesto a su vez por campos. Estos campos se distinguen por estar comprendidos por una barra vertical “|”. El registro dispondrá de separadores de campo tanto para los campos con información como para los campos vacíos, para los cuales aparecerán dos barras verticales seguidas sin contenido de por medio: “||”.

Si continuamos adentrándonos desde lo general a lo particular, nos encontramos con que dentro de los campos aparecen los subcampos. Otro rango de información comprendido por “\”.

Como norma general, el primer campo de un registro se conoce como la cabecera del registro, y se trata de una letra mayúscula que indica el tipo de registro.

4.3 Tipos de registros.

Para el desarrollo de un archivo con formato FIEBDC orientado a la medición y presupuesto se van a utilizar seis tipos distintos de registros, los cuales se presentan a continuación:

I) Registro tipo propiedad y versión.

Sirve para documentar la procedencia y el formato de los archivos, haciendo las veces de encabezado.

La letra que da comienzo a este tipo de registro es la V. Siguiendo con las pautas indicadas anteriormente, para el ejemplo expuesto en el apartado 6, se ha obtenido el siguiente registro:

“ ~V|FIEBDC-3/2012|MEDyPRES||TFG| “

II) Registro tipo coeficientes.

Informa sobre el número de decimales que se tienen en cada campo con valor numérico.

La letra que encabeza este registro es la K. En el ejemplo práctico se ha obtenido:

“ ~K|3\3\3\2\2\2\2\EUR|0| ”

III) Registro tipo concepto.

Muestra la información básica que describe las unidades de obra o los capítulos. De modo que para el ejemplo práctica se ha usado este registro para contener la información dada en los campos de la base de datos. Así, el ejemplo queda:

“ ~C|01#||Estructura principal|2020.61|31/07/2019|0| “ ,

Donde se expresa la información relativa al capítulo 01, aunque en otra línea del archivo también se muestra información sobre sus cada uno de los componentes o unidades de obra:

“ ~C|0103|m|Balancin|500.00|18/07/2019|0| “

En ambos ejemplos aparece en primer lugar la letra que describe este tipo de registro, la letra C. Tras ella aparece el código, la unidad, el nombre, el precio y la fecha.

La diferencia es que para el balancín se muestra el precio unitario y para la estructura principal el coste total del capítulo con las cantidades de material usadas.

IV) Registro tipo descomposición.

Este registro contiene como su propio nombre indica la descomposición de un concepto en otros. Se desmenuza la composición del concepto padre y muestra los conceptos hijos que lo forman.

De este modo, un capítulo se puede descomponer en las unidades de obra que lo forman. Por ejemplo:

“~D|01#|0101\1\0.785\0102\1\0.016\0103\1\3.000\0104\1\2.000| “

Si vamos mirando de izquierda a derecha el ejemplo expuesto, se observa que en primer lugar aparece la letra D para dar comienzo al registro tipo descomposición. Después, se indica el código del capítulo que va a descomponer. Tras él aparece el código del primer elemento que lo compone, acompañado por su cantidad de material en su unidad de medida. Y sucesivamente pasa lo mismo con los demás elementos que forman parte del capítulo.

V) Registro tipo texto.

Este registro permite describir un concepto con palabras. Véase el siguiente ejemplo:

“~T|0101|Base de hormigón que sirve para dotar de estabilidad a la estructura frente a la irregularidad del terreno| “

Como se puede apreciar, aparece la letra T para dar comienzo a este tipo de registro, luego aparece el código de la unidad de obra, y por último la descripción aportada sobre dicha unidad de obra.

VI) Registro tipo mediciones.

Este registro contiene las cantidades empleadas de cada elemento en el proyecto, así como el capítulo al que pertenecen y el lugar que ocupan en dicho capítulo.

Esta vez la letra que abre este registro es la M. Todo lo dicho se puede corroborar con este ejemplo:

“~M|01#|0103|01\3\|3.000|| “

Se puede comprender fácilmente que 01 es el capítulo. 0103 es el código del elemento, descomponiéndose el código para indicar que es el tercer elemento del capítulo y, por último, la cantidad usada.

En este caso, al tratarse del elemento balancín empleado en el caso práctico del apartado 6, la cantidad hace referencia a 3 metros. Aunque para otros elementos puede hacer referencia a unidades, metros cuadrados, metros cúbicos o kilogramos.

5 APLICACIÓN VBA.

Microsoft Visual Basic es la herramienta que permite al usuario poder ejecutar el programa desarrollado y obtener como resultado el presupuesto buscado. Por tanto, se hace imprescindible describir algunos conceptos básicos sobre esta herramienta.

La definición más amplia y general es que se trata de un lenguaje de programación. Lo que lo hace tan útil es la conectividad que tiene con otros programas, permitiendo acceder y operar desde Visual Basic.

En el caso que nos atañe, Visual Basic permite que conectemos tanto con Catia, para acceder a la medición de la construcción que se presente, como con Access para recuperar la base de datos que se haya aportado.

Por otro lado, los códigos compilados en Visual Basic se almacenarán en librerías que tendrán la extensión “.catvba”. Dentro de una misma librería se podrán almacenar multitud de códigos de programación, también conocidos como macros.

5.1 Visual Basic en CATIA.

El procedimiento para acceder a Visual Basic desde la ventana de CATIA es muy sencillo, pero la primera vez que vayamos a grabar una macro necesitaremos crear una librería donde almacenarla.

Para ello hay que ir a la barra de herramientas de la ventana de CATIA y clicar en:

Tools → Macro → Macros

Entonces aparecerá la siguiente ventana emergente:

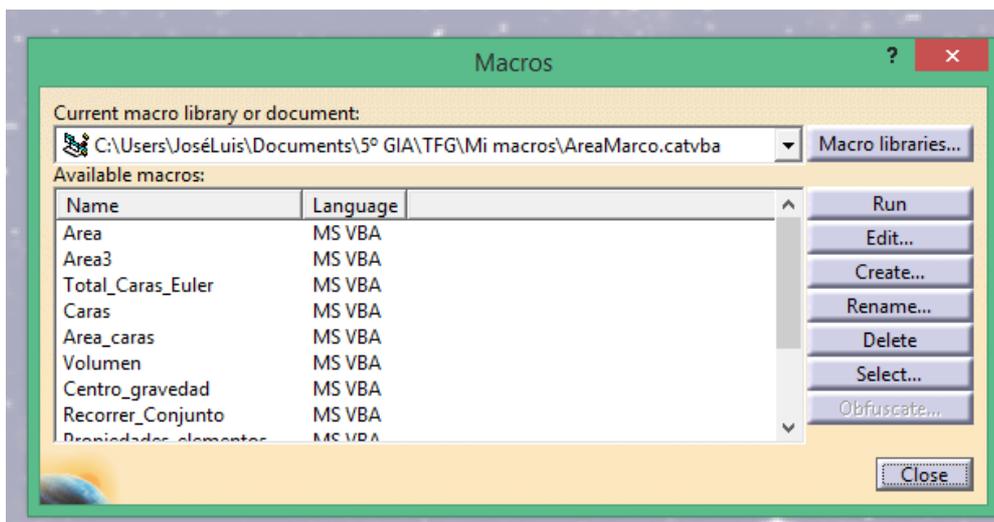


Figura 5-1. Ventana para acceder a las macros

A continuación, clicamos en “Macro libraries”. En la nueva ventana que aparecerá tenemos la opción de añadir nuevas macros o nuevos directorios.

En primer lugar, es conveniente crear un directorio donde almacenar las librerías que guardemos posteriormente. Esto se consigue clicando en la pestaña de la parte superior y seleccionando directories. Cuando pulsemos sobre “Add new libraries”, se abrirá una nueva ventana donde nos pedirá que le demos un nombre y una ubicación al directorio. Bastará con copiar la ubicación deseada de nuestro equipo.

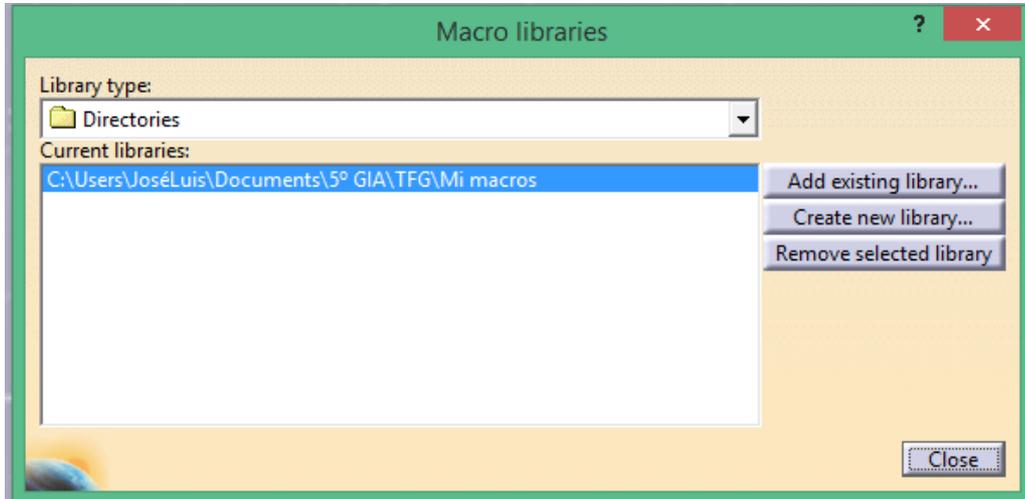


Figura 5-2. Ventana para la creación de un directorio

Una vez ya se disponga de un directorio y se almacenen en él las macros, aparecerá una lista con todas ellas, tal y como se observa en la figura 5-1, teniendo solo que seleccionar la macro que se quiera ejecutar y clicar en “Run”.

Si queremos ver el código, clicamos en “Edit”, y es entonces cuando se abrirá una nueva ventana completa donde se visualizará y se tendrá acceso a la interfaz de Visual Basic.

La interfaz de la ventana de Visual Basic es muy intuitiva. A continuación, se muestra un recorte de la parte superior izquierda de la ventana de Visual Basic.

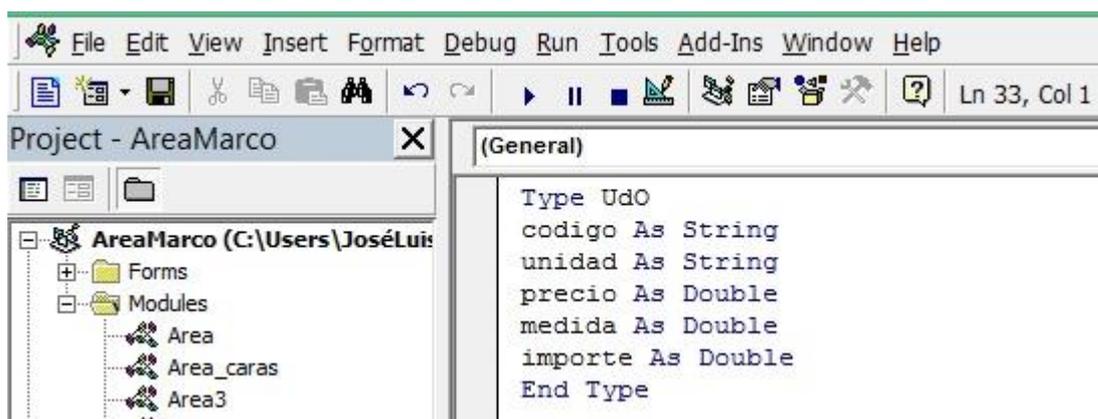


Figura 5-3. Interfaz superior-izquierda de Visual Basic

En el sector izquierdo de la figura 5-3 aparece la librería con las distintas macros almacenadas en ella. Haciendo doble click en cualquiera de ellas conseguimos abrir su código para poder trabajar con él y editarlo.

En la parte superior de la *figura 5-3*, se observa una barra de herramientas donde el símbolo de guardar mantiene la forma habitual con respecto a la interfaz de otros programas.

El botón de play sirve para ejecutar el programa, y el stop para detener la ejecución en caso de reportarse algún error y comenzar posteriormente a ejecutar el programa desde el principio.

Cuando nuestra macro tenga algún error en la ejecución emergerá una ventana como esta:

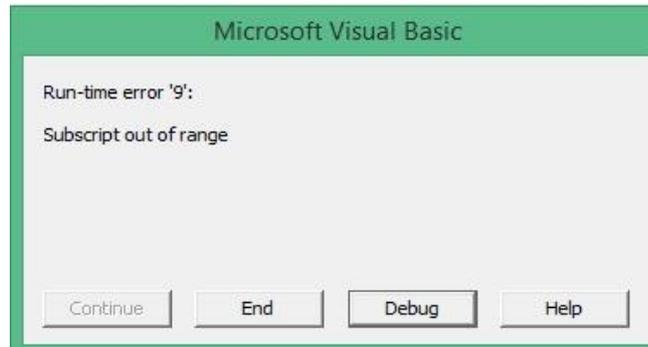


Figura 5-4. Ventana de reporte de error en VBA

Si pulsamos sobre “Debug”, nos indicará la línea del programa donde se encuentra el error resaltada con color amarillo. Tal y como aparece en la *figura 5-5*:

```

Debug.Print
Debug.Print "CÓDIGO"; Tab(25); "RENDIMIENTO"
Debug.Print "-----"
For m = 1 To UBound(ListaCapitulos)
    Debug.Print ListaCapitulos(m).codigo; Tab

```

Figura 5-5. Línea de error del programa

Como último comentario, las sentencias que aparecen en el código de la *figura 5-5* encabezadas por la función interna “Debug.Print”, sirve para escribir por la pantalla de ventana inmediata.

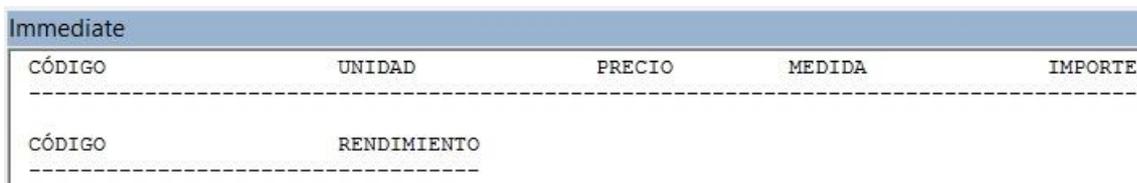


Figura 5-6. Ventana inmediata Visual Basic

Esta ventana puede estar visible o no. Si no lo está, basta con pulsar Ctrl + g en el teclado de su ordenador para que aparezca en la parte inferior de la pantalla. La ventana inmediata puede ampliarse o reducirse arrastrando su borde superior. Y si desea volver a ocultarla para ampliar su visión del código, bastará con pulsar “X”, situada en el extremo superior derecho de la propia ventana, para cerrarla.

5.2 Aplicación desarrollada en VBA

Es importante destacar que para que este código funcione es necesario ejecutar el programa con la ventana del Product de la construcción abierta en CATIA.

Para ayudar a la comprensión del funcionamiento de la aplicación, se ha elaborado el siguiente diagrama de flujo:

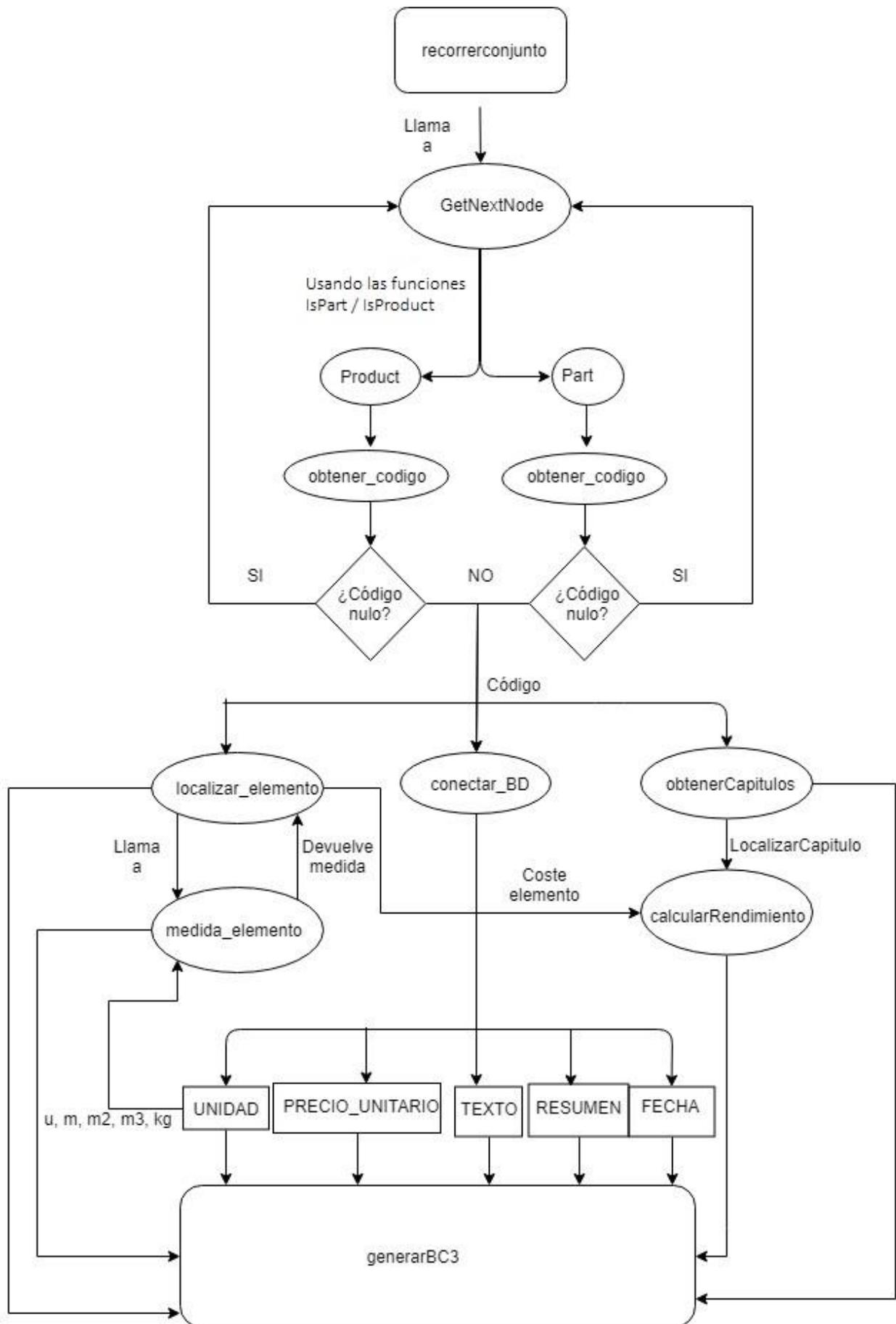


Figura 5-7. Diagrama de flujo de la aplicación

Rutina principal.

La subrutina principal es recorrerconjunto, y a partir de ahí se llama a otras subrutinas para que ejerzan las funciones que se irán describiendo. Primero llama a la subrutina GetNextNode.

Recorrer conjunto.

GetNextNode sirve para recorrer la estructura árbol del ensamblaje elemento a elemento.

Comprueba si el elemento seleccionado está contenido en un partdocument o productdocument usando las funciones IsPart / IsProduct. Esta distinción se realiza porque posteriormente para obtener el código del elemento las líneas de código varían si se trata de un partdocument o productdocument.

Obtención del código.

Seguidamente se obtiene el código del elemento a través de la subrutina obtener_codigo, que devuelve este código a GetNextNode donde se comprueba si el código es nulo o no. Si es nulo se muestra un mensaje por pantalla (que se puede obviar si comentamos la sentencia que lo activa) indicando que el código es nulo y se debe profundizar en el elemento, volviendo a direccionarte a GetNextNode para entrar dentro de la estructura árbol de ese elemento. Si es distinto de nulo se llama a la subrutina conectar_BD.

Conexión con Base de Datos Relacional.

Esta subrutina toma el código obtenido y enlace con la base de datos registrada en Access, recuperando los datos contenidos en los campos de dicha base de datos.

Listado de elementos.

GetNextNode llama ahora a la subrutina localizar_elemento, que almacena el código en una lista, poniéndolos en orden y actualizando siempre el valor de las mediciones y el importe para un código que se haya registrado con anterioridad en dicha lista.

La forma en la que se procede a la ordenación es comparando los códigos de cada elemento y ordenándolos de menor a mayor. Por tanto, los primeros elementos en aparecer serán los del capítulo 1, y desde el primer elemento hasta el último. Después aparecerán los elementos pertenecientes al capítulo 2 y así sucesivamente.

Para el ejemplo práctico se tiene como resultado este orden:

CÓDIGO	UNIDAD	PRECIO	MEDIDA	IMPORTE
0101	m2	250	0,785398163397448	196,349540849362
0102	m3	1500	0,016170860326995	24,2562904904924
0103	m	500	3	1500
0104	u	150	2	300
0201	m	100	0,24	24
0202	u	0,1	8	0,8
0203	m2	200	0,06	12
0301	u	50	2	100
0302	u	20	2	40

Tabla 5-1: Ordenación de códigos

El criterio mencionado para la ordenación de los códigos se ve fácilmente en la tabla superior.

Por otra parte, los capítulos también son presentados en orden desde el de menor número hasta el mayor. Así, se tiene que en el caso práctico aparecerán de este modo:

CÓDIGO	RENDIMIENTO
01	2020,60583133985
02	36,8
03	140

Tabla 5-2: Ordenación de capítulos

Hay que comentar que las *tablas 5-1* y *5-2* no se muestran como tal en el presupuesto final, el cual se presentará en el último apartado.

Estas tablas se han usado únicamente con el fin de verificar que las mediciones realizadas, los parámetros aportados, y la ordenación de los códigos era la correcta.

Una diferencia entre cómo se reflejan los resultados en las *tablas 5-1* y *5-2*, y cómo se reflejarían en el presupuesto final está en los decimales. De ninguna manera aparecerán en el presupuesto tantísimos decimales para ninguna medida, importe o rendimiento.

Para evitar este carrusel de decimales se ha implementado una restricción de 2 decimales tanto para el importe como para el rendimiento. Esto tiene una justificación muy sencilla, y es que, en términos económicos, la exactitud llega hasta el céntimo de euro. Por lo que no tiene sentido ir más allá.

Sin embargo, para la medida si se ha añadido un decimal más, resultando un total de tres. El objetivo de añadirle un decimal básicamente es incrementar la exactitud de la medición.

En cualquier caso, conviene explicar qué significan los campos que aparecen en dichas tablas, puesto que aparecerán igualmente en el presupuesto final, aunque con el formato debido.

Los campos de la *tabla 5-1* “UNIDAD” y “PRECIO” ya se conoce que son referidos a la magnitud que hay que medir y al precio unitario de ese elemento.

“MEDIDA” no es más que el resultado obtenido de la medición del elemento en unidades del Sistema Internacional. A modo de recordatorio véase la *tabla 5-3* mostrada seguidamente:

MAGNITUD MEDIBLE	UNIDAD EN S.I.
LONGITUD	m
ÁREA	m ²
VOLUMEN	m ³
MASA	Kg
ELEMENTOS	Unidades

Tabla 5-3: Unidades en Sistema Internacional

Por último, el importe del elemento lo obtiene a partir de la medición devuelta por la subrutina `medida_elemento` multiplicado por el precio unitario recogido de la base de datos anteriormente mediante `conectar_BD`.

Si observamos ahora la *tabla 5-2* de nuevo, cada capítulo lleva asociado un rendimiento. El rendimiento no es más que la suma del coste de cada unidad de obra contenida en un mismo capítulo.

Es decir, el rendimiento es el coste económico destinado a cada capítulo.

Medición de elementos.

Para realizar las mediciones, `localizar_elemento` llama a la subrutina `medida_elemento`, que junto con la entrada del campo "UNIDAD" recogido de la base de datos, determina que magnitud del elemento hay que medir.

Para el caso en el que la magnitud a medir sea la longitud, ya se ha explicado que es necesario definir una serie de puntos que delimiten dicha longitud. Sin embargo, también se requiere añadir una propiedad extra al elemento en cuestión. Hay que indicar si la longitud que queremos medir es abierta o cerrada, ya que el resultado variará de en función del tipo de longitud que se indique.

Esto sucede porque la medida de la longitud se obtiene a partir de la distancia entre las coordenadas de los puntos creados. Así, si se tienen únicamente dos puntos, bastará con calcular la distancia que hay del primero al segundo. Sin embargo, si se tienen tres puntos, se pueden dar dos casos:

Caso A: Geometría abierta.

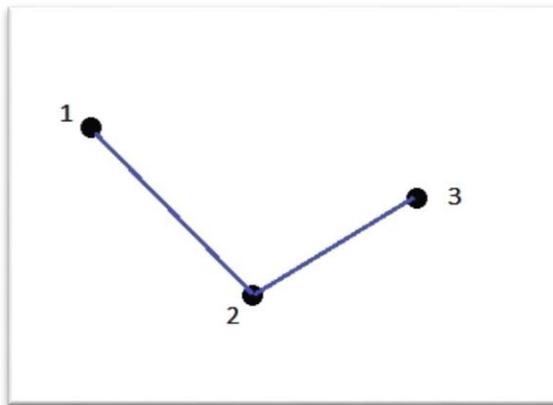


Figura 5-8. Ejemplo de geometría abierta

Para este supuesto habría que calcular la distancia del punto 1 al 2, y la distancia del punto 2 al 3. Siendo la longitud total la suma de ambas distancias.

Caso B: Geometría cerrada.

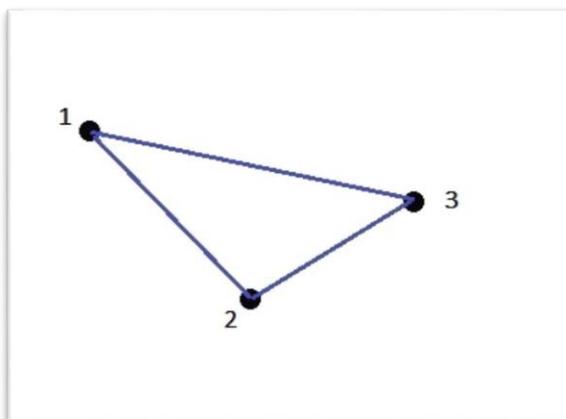


Figura 5-9. Ejemplo de geometría cerrada

En contraposición al caso A, a pesar de tener definidos los mismos puntos, la longitud para este supuesto no puede ser la misma debido a que cuenta con un nuevo segmento que une el punto 3 con el 1. Siendo la longitud total para el caso B la suma de este nuevo segmento con la longitud total para el caso A.

Debido a la necesidad demostrada de indicar si una longitud es abierta o cerrada, se define “TIPO LONGITUD” en CATIA como una propiedad del elemento tipo “String” dándole como posibles valores “CERRADA” o “ABIERTA”.

Para el cálculo de las longitudes mediante las coordenadas de los puntos se ha usado la siguiente expresión:

$$\sum_{i=1}^N \overline{P_i P_{i+1}} = \sum_{i=1}^N \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}$$

Donde x, y, z son las coordenadas de los puntos en el espacio tridimensional, y el segmento medido va del punto i al punto i+1. En consecuencia, la longitud total medida será la suma de las longitudes de cada segmento entre puntos.

Por otro lado, cuando recupere la unidad “m2”, la aplicación se encontrará con dos variantes: que el área esté definida por puntos o que se pretenda obtener el área mayor.

La aplicación decidirá si calcular el área definida por puntos o el área mayor dependiendo de si detecta que se le han añadido puntos con la nomenclatura detallada en el apartado 2.3. Si es así, entenderá que se pretende medir el área definida por puntos. En caso de no encontrar puntos con esa nomenclatura calculará el área mayor del elemento.

Además, se puede apreciar en la subrutina de medida_elemento que, tanto para el cálculo de una longitud, como el de un área definida por puntos, se ha usado un factor de conversión para que el resultado se encuentre en las unidades del Sistema Internacional de medida.

Cuando medida_elemento recibe el valor unidad “u”, cuenta un elemento. Así, si vuelve a aparecer el mismo código posteriormente al recorrer la estructura árbol, volverá a contar un elemento y lo acumulará al anterior.

Para la obtención del volumen se ha usado una función interna de CATIA, y para el cálculo de la masa se usa el volumen obtenido multiplicado por el valor de la densidad que se haya definido como propiedad en el elemento en CATIA.

Obtención de capítulos.

La subrutina obtenerCapitulos sirve para generar los capítulos existentes a partir de las dos primeras cifras del código del elemento que se esté midiendo.

Lista de capítulos.

La subrutina localizarCapitulo sirve para buscar el capítulo generado anteriormente en obtenerCapitulos para añadirle el nuevo elemento. Si no lo encuentra entonces crea un nuevo registro con el capítulo obtenido y lo ordena numéricamente junto con los ya existentes.

Rendimiento de los capítulos.

La subrutina calcularRendimiento sirve para sumar los costes económicos de todos los elementos pertenecientes a un mismo capítulo y así obtener el rendimiento del capítulo. Toma como entrada el coste de cada elemento y la lista de capítulos, así para cada elemento se deduce a qué capítulo pertenece y se añade su coste al capítulo correspondiente.

Generación del archivo de texto BC3.

La subrutina generarBC3 se encarga de dar el formato al archivo de texto que se va a crear siguiendo los tipos de registros ejemplificados en el apartado 4.3.

Por último, las subrutinas recuperarDatosConceptos y recuperarTextos sirven para recoger de la base de datos la información contenida en los campos de “RESUMEN” y “FECHA”, y “DESCRIPCIÓN” respectivamente. Estas subrutinas son llamadas desde generarBC3 para aportar al archivo de texto generado esta información a los registros tipo texto y tipo composición.

En el diagrama de flujo de la *figura 5-7* se ha indicado que estos campos se obtienen de conectar _BD porque perfectamente se podrían incluir en esa subrutina, ya que conecta con la base de datos de igual forma a través del código.

En la aplicación se ha decidido crearles subrutinas independientes para que cada una sea llamada cuando se necesite. Teniendo que para los campos resumen, fecha y descripción no se necesitan hasta la generación del archivo de texto BC3.

6 EJEMPLO DE ENSAMBLAJE: BALANCÍN.

Para poner en práctica todos los detalles e indicaciones que se han ido contando, se va a exponer un ejemplo de conjunto al que se le quiere realizar una medición y un posterior presupuesto.

Este ensamblaje va a ser un balancín de los que estamos acostumbrados a ver en los parques de cualquier ciudad.

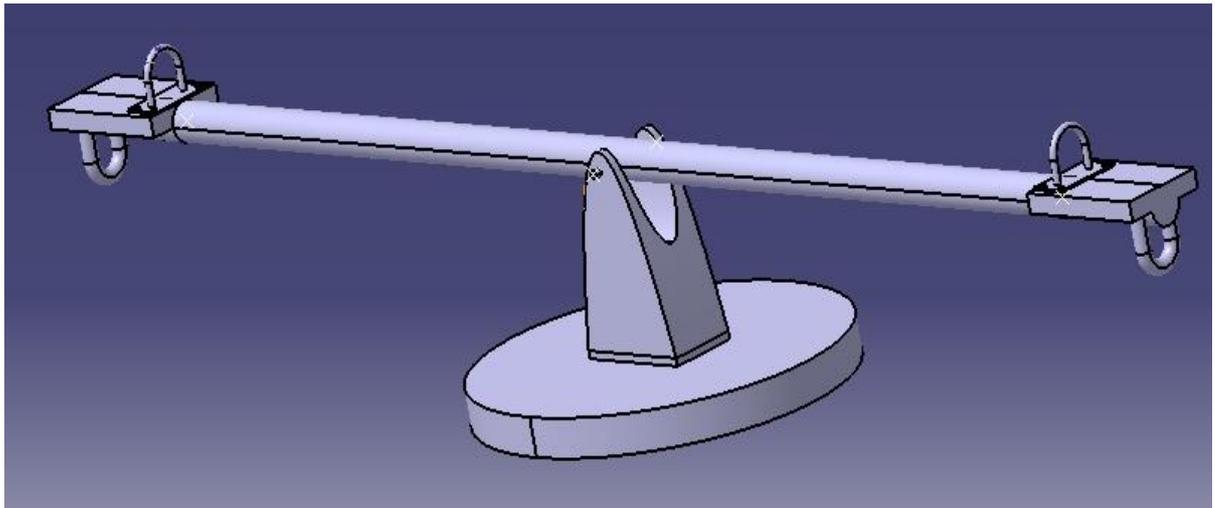


Figura 6-1. Ensamblaje balancín

6.1 Elementos del ensamblaje: Medición.

Se va a ir mostrando uno a uno los elementos empleados para la construcción de este ensamblaje y qué propiedades se han añadido a cada uno.

Se va a seguir el orden dado por la estructura árbol creada en CATIA que se presenta de esta manera:

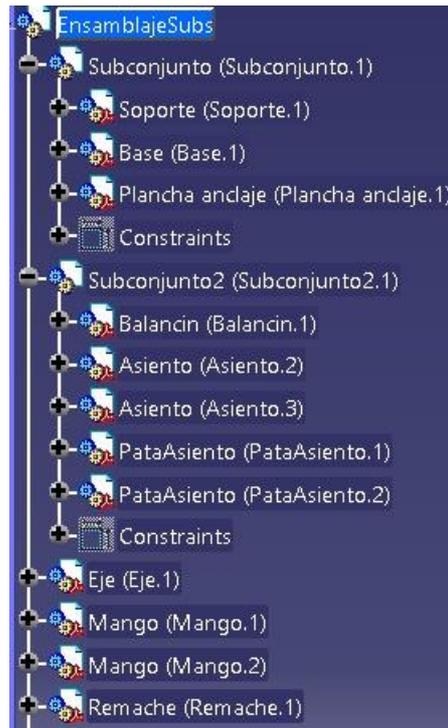


Figura 6-2. Estructura árbol ensamblaje

En la estructura árbol se presentan todos los componentes de la construcción. Además, también se puede ver que hay dos subconjuntos que están formados a su vez por otros elementos.

Hay que resaltar que hay varios elementos que aparecen varias veces, tantas como la cantidad de ellos que se haya utilizado.

En el caso del remache se han utilizado 8, pero como la estructura árbol ya no aportaba ningún elemento nuevo se ha decidido suprimir de la figura los 7 remaches restantes.

Hay que destacar que, para que el programa funcione, se debe ejecutar siempre y cuando la ventana activa de CATIA sea el Product con el conjunto que queramos medir. En caso contrario se nos reportará un error.

6.1.1 Subconjunto.

Con el objetivo de poner ejemplificar el supuesto en el que nos encontramos ante un elemento no medible se ha ideado el diseño de un subconjunto. El subconjunto está formado por los elementos que dotan al conjunto de estabilidad y, en consecuencia, son los elementos de mayor robustez. Estos elementos son: la base, la plancha de anclaje y el soporte.

Tras ejecutar el programa, se empieza a recorrer la estructura árbol de la construcción, y el primer elemento que se encuentra es este subconjunto.

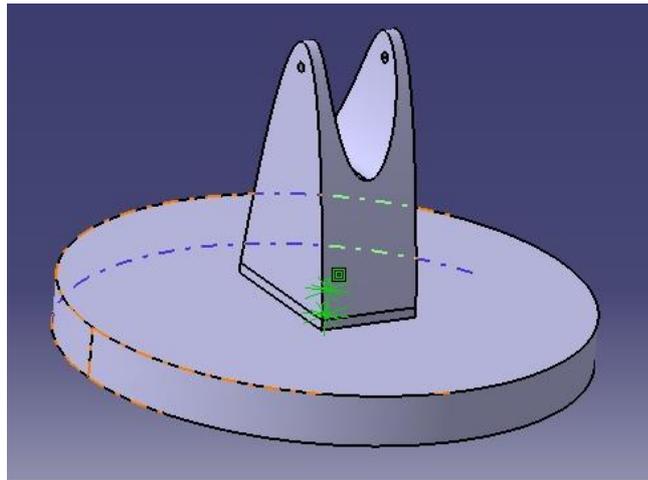


Figura 6-3. Subconjunto

Al no ser medible, automáticamente se profundiza en el subconjunto. Sin embargo, para hacer visible este proceso se ha establecido un cuadro de diálogo que nos comunica cuando tenga lugar esta situación. En consecuencia, aparece el siguiente cuadro de diálogo:

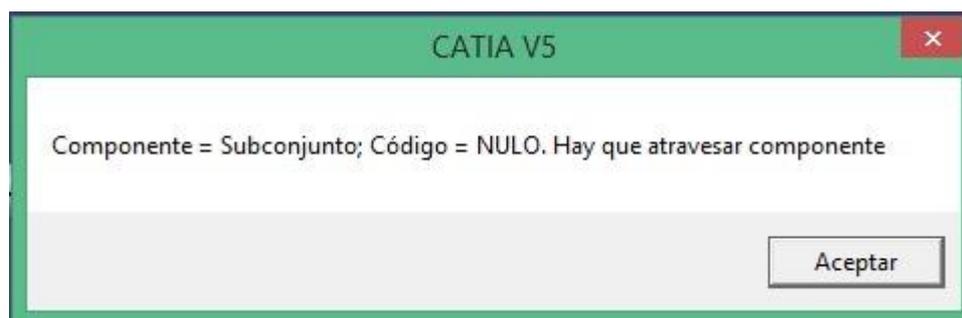


Figura 6-4. Cuadro de diálogo. Profundización en elemento no medible.

Si aceptamos, el programa sigue su curso acometiendo la profundización y encontrándose con el soporte, que es el primer elemento que aparece dentro de la estructura del subconjunto.

6.1.1.1 Soporte.

El soporte está identificado con el código 0102 en la base de datos, y además habrá que añadir en CATIA la propiedad al elemento como CÓDIGO = 0102.

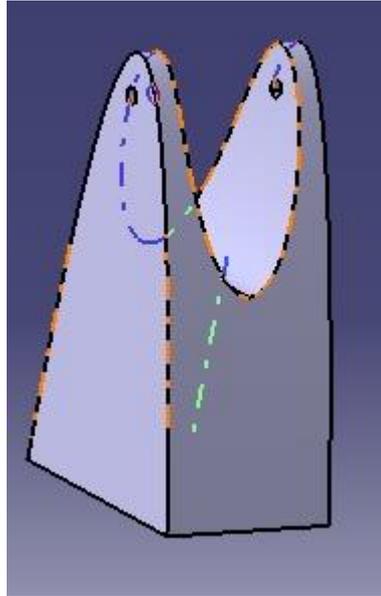


Figura 6-5. Soporte

Del soporte se pretende medir en un principio el volumen, por lo que hay que establecer en la base de datos el campo UNIDAD = m3.

Al ejecutar el programa y activar el cuadro de mensaje para que nos informe del resultado se tiene que:

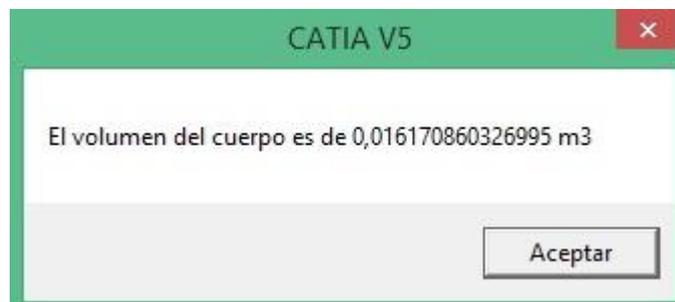


Figura 6-6. Cuadro de diálogo. Volumen soporte

Para activar estos cuadros de diálogo solo hay que quitar la comilla que convierte la sentencia en un comentario. Estas sentencias están en la subrutina medida_elemento y encabezadas por la función MsgBox.

El anexo con el código tiene estas sentencias comentadas con la comilla con la finalidad de que al ejecutar el programa automáticamente obtengamos el archivo bc3, sin tener que pasar por los muchos cuadros de diálogo que hay en el programa y que se han usado para seguir su correcto funcionamiento.

Sin embargo, para ver los resultados elemento a elemento, se está usando la estrategia de habilitar los cuadros de diálogo.

Ya tenemos calculado el volumen del soporte, pero pongamos la situación de que ahora nos dan un precio unitario de los kilogramos de material. Entonces, ya no necesitamos conocer el volumen, sino la masa. ¿Qué datos tengo que cambiar para obtener como resultado la masa?

Solo se necesita hacer un par de cambios en los datos:

- 1) Modificar el campo UNIDAD en la base de datos y establecer kg en lugar de m3.
- 2) Definir en CATIA la propiedad DENSIDAD, y darle el valor de densidad que tenga nuestro material. En nuestro caso, como se trata de un soporte de acero, se ha establecido que la DENSIDAD = 7850 kg/m^3 .

Tras estas modificaciones, ejecutamos el programa y obtenemos que:



Figura 6-7. Cuadro de diálogo. Masa soporte

Si se quiere se puede comprobar con una calculadora fácilmente que el cambio de volumen a masa se ha realizado correctamente.

6.1.1.2 Base.

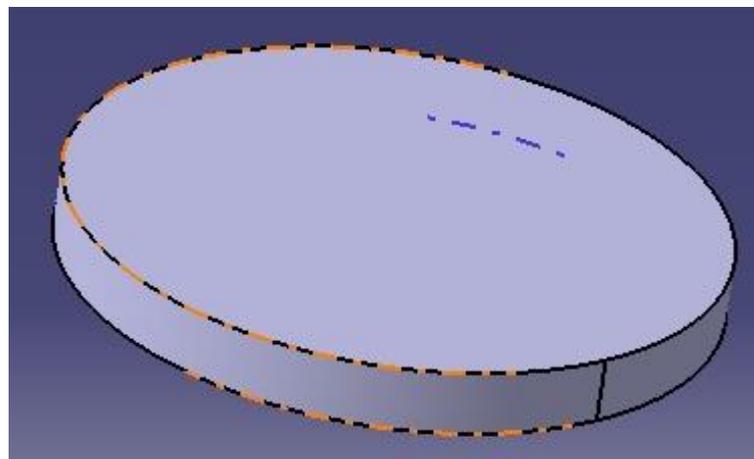


Figura 6-8. Base

Para la base de hormigón, se tiene el código 0101, y se establece UNIDAD en la base de datos como m2, por lo que se pretende medir el área mayor del cuerpo. En este caso, buscamos conocer el área de la superficie superior, que resulta ser la de mayor área junto con la superficie inferior.

Tras ejecutar el programa se obtiene el resultado de dicha área:

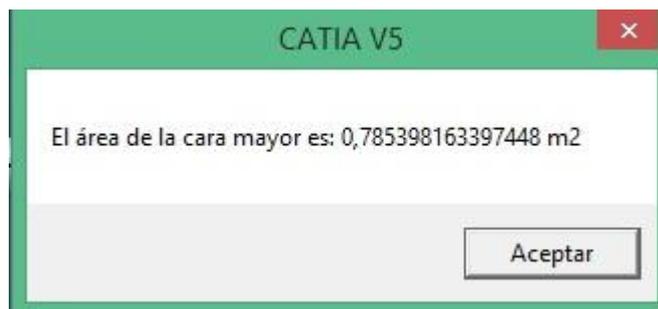


Figura 6-9. Cuadro de diálogo. Área mayor de la base

6.1.1.3 Plancha de anclaje.

La plancha de anclaje es un elemento de acero usado para fijar el soporte del balancín a la base de hormigón.

Está identificada con el código 0203, y en el campo UNIDAD de la base de datos tiene establecido el parámetro m2, ya que se pretende calcular la superficie que está en contacto con el soporte, siendo esta superficie la de mayor área de la plancha de anclaje.

A continuación, se muestra tanto el diseño de la plancha de anclaje como el resultado del área obtenido:

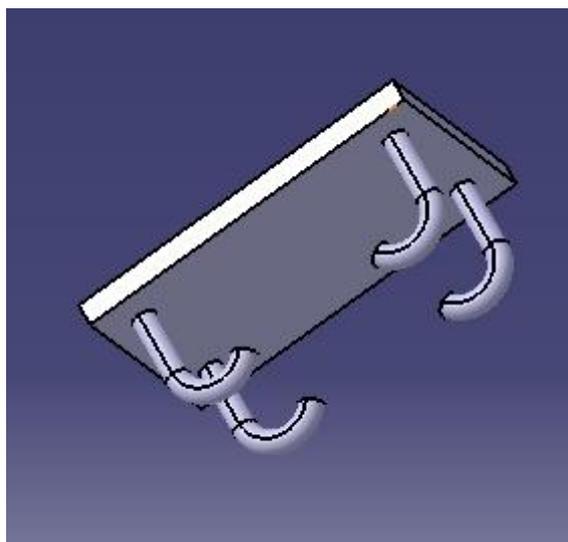


Figura 6-10. Plancha de anclaje

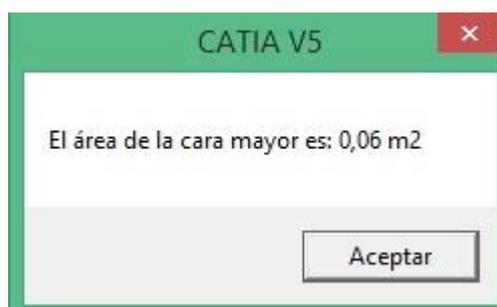


Figura 6-11. Cuadro de diálogo. Área superior plancha de anclaje

Con el propósito de demostrar que el área definida por puntos también obtiene el área correctamente, se van a definir cuatro puntos en la plancha de anclaje a partir de los que se hallará el área de la superficie de contacto de la plancha de anclaje con el soporte, con la esperanza de que coincida con el área ya calculada.

Los puntos se denominarán siguiendo la estructuración indicada en el apartado 2.3: “PuntoAreaPlancha anclaje.1”, “PuntoAreaPlancha anclaje.2”, “PuntoAreaPlancha anclaje.3”, y “PuntoAreaPlancha anclaje.4”.

Se debe respetar el espacio que hay entre “Plancha” y “anclaje”, puesto que es el nombre que se le ha dado al Part.

En la estructura árbol del elemento se observa la nomenclatura usada:



Figura 6-12. Estructura árbol plancha de anclaje

Tras definir los puntos con la nomenclatura indicada, automáticamente la aplicación entiende que el área que se pretende medir es la definida por esos puntos.

En el anexo del código, se puede ver que se le ha añadido un factor de conversión al resultado, puesto que el cálculo del área definida por puntos se realiza en mm^2 . Por tanto, se multiplica el resultado por 0.000001 para convertir a m^2 .

6.1.2 Subconjunto 2.

La creación del subconjunto 2 es la misma que la del subconjunto, probar que profundiza en él para

medir los elementos que lo componen. Está formado por la parte móvil del columpio durante su funcionamiento: balancín, asientos, y amortiguadores de asiento.

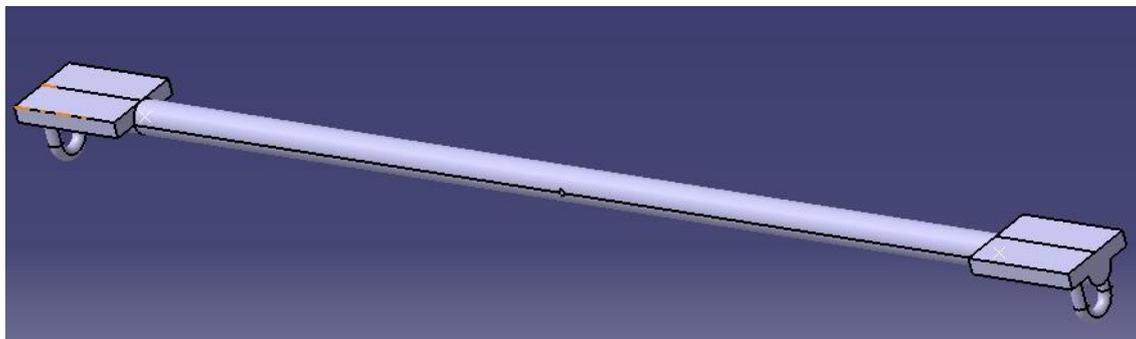


Figura 6-13. Subconjunto 2

6.1.2.1 Balancín.

El balancín es una barra de acero que se somete a momentos de balanceo cuando el columpio esté construido y esté en funcionamiento.

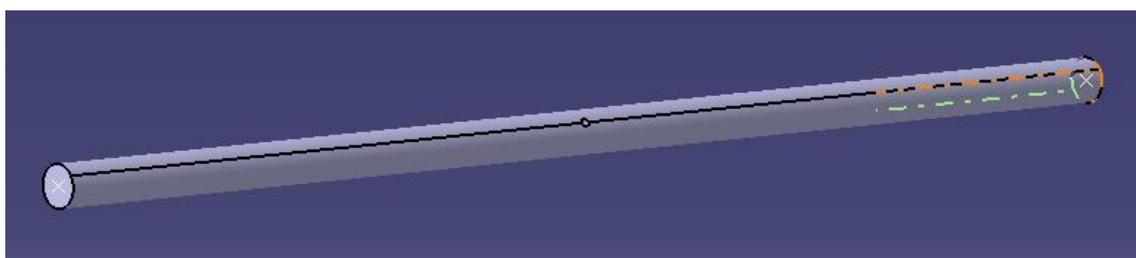


Figura 6-14. Balancín

Está identificado con el código 0103 y se pretende conocer la longitud, por lo que en la base de datos se establece “UNIDAD = m”, y se añade la propiedad “TIPO LONGITUD = ABIERTA”.



Figura 6-15. Cuadro de diálogo. Longitud balancín

Para el cálculo de la longitud se ha definido un punto en cada extremo de la barra, respetando la nomenclatura citada en el apartado 2.3: “PuntoLongBalancin.1” y “PuntoLongBalancin.2”.

6.1.2.2 Asientos.

Los asientos son de acero y están soldados al balancín. Están identificados con el código 0104 y de

ellos sólo se pretende conocer el número de unidades. Por tanto, el campo UNIDAD en la base de datos para el asiento deberá contener la letra u.

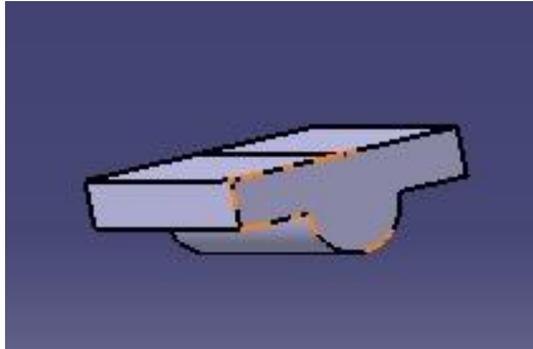


Figura 6-16. Asiento

Al ejecutar el programa, se cuenta una unidad cada vez que el código sea el 0104, por lo que en el archivo final se obtiene el número total de asientos. Este procedimiento es el mismo para cualquier elemento del que haya que conocer el número de unidades.

Se puede comprobar que el número de unidades obtenidas es la correcta si observamos la *tabla 6-1*. En el campo MEDIDA, para el código 0104, aparece el número 2.

6.1.2.3 Amortiguador.

Con el objetivo de reducir la fuerza de un posible impacto del asiento de acero con el suelo, se ha instalado un amortiguador de goma en la parte inferior del asiento.

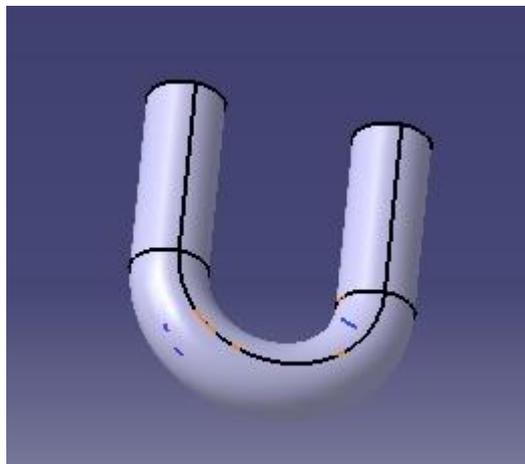


Figura 6-17. Amortiguador de goma

Este elemento se ha identificado con el código 0302, y se ha nombrado en catia como “PataAsiento”. De él solo se pretende contar las unidades utilizadas, por lo que en el campo UNIDAD de la base de datos se establece el valor “u”.

Al igual que en el asiento, si se mira la *tabla 6-1* se comprueba que en el campo medida se contabilizan 2 unidades.

6.1.3 Eje.

El eje es el elemento que aporta el grado de libertad de giro del balancín con respecto al soporte.

Está identificado con el código 0201, y se quiere calcular su longitud, por lo que se definen dos puntos tal y como se hizo en el apartado 2.3: “PuntoLongEje.1” y “PuntoLongEje.2”.

Añadiendo la propiedad “TIPO LONGITUD = ABIERTA”, y estableciendo en el campo UNIDAD de la base de datos el valor “m”, se tiene:



Figura 6-18. Cuadro de diálogo. Longitud eje

6.1.4 Mango.

Se añaden dos mangos de acero que irán remachados uno a cada asiento, para lo que dispone de una placa soldada con cuatro taladros.

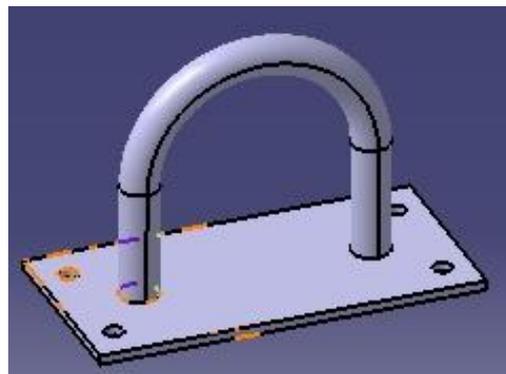


Figura 6-19. Mango

El mango está identificado con el código 0301. Se pretende contabilizar las unidades empleadas, por lo que “UNIDAD = u” en la base de datos.

Si se vuelve a mirar la *tabla 6-1*, se comprueba que se tienen contabilizadas en el campo “MEDIDA” dos unidades.

6.1.5 Remache.

Los remaches se utilizan para fijar los mangos a los asientos.

Están identificados con el código 0202, y se define el campo “UNIDAD = u” en la base de datos, ya que se pretende contabilizar el número de unidades de remaches.

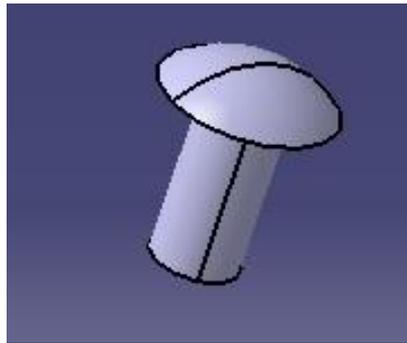


Figura 6-20. Remache

Al igual que para el resto de los elementos en los que hay que contar sus unidades, basta con mirar el campo “MEDIDA” de la *tabla 6-1*. Se comprueba que se emplean 8 unidades, 4 para la fijación de cada mango a su respectivo asiento.

6.2 Organización del ensamblaje en capítulos.

El ensamblaje se estructura en 3 capítulos distintos. Como ya se ha comentado anteriormente, las dos primeras cifras del código indican el capítulo al que pertenece cada elemento. Así que, siguiendo esta regla, se obtiene la siguiente organización:

6.2.1 Capítulo 1: Estructura principal.

El capítulo 1 lo forman la base, el soporte y la barra de balancín. Estos 3 elementos se caracterizan por ser los de mayor tamaño y envergadura, y los que componen el exoesqueleto principal de un balancín.

6.2.2 Capítulo 2: Uniones.

El capítulo 2, como su propio nombre indica, contiene los elementos que unen otros dos elementos de la construcción: remaches, eje y plancha de anclaje.

Los remaches fijan el mango al asiento.

El eje une la barra de balancín al soporte, aportando el grado de libertad necesario para el movimiento de rotación del balancín.

La plancha de anclaje fija el soporte de acero a la base de hormigón extendida sobre el terreno.

6.2.3 Capítulo 3: Elementos de seguridad.

El capítulo 3 lo forman el mango y el amortiguador situado debajo del asiento. Ambos elementos aumentan la seguridad del usuario del balancín.

Por un lado, el mango permite una sujeción extra con la que evitar posibles caídas del balancín.

Por otro lado, el amortiguador de goma permite suavizar el posible impacto del asiento contra el suelo, reduciendo así la probabilidad de contusiones en el coxis.

6.3 Tablas de presupuesto de comprobación.

En este apartado se va a mostrar una tabla semejante a la *tabla 6-1*, con los cambios implementados en la en el campo “UNIDAD” durante la medición de los elementos realizada.

CÓDIGO	UNIDAD	PRECIO	MEDIDA	IMPORTE
0101	m2	250	0,785398163397448	196,349540849362
0102	kg	10	126,94125356691	1269,4125356691
0103	m	500	3	1500
0104	u	150	2	300
0201	m	100	0,24	24
0202	u	0,1	8	0,8
0203	m2	200	0,06	12
0301	u	50	2	100
0302	u	20	2	40

Tabla 6-1: Mediciones y presupuesto tras modificaciones

Se puede observar como el elemento soporte, con código 0102, se ha modificado la unidad de “m3” a “kg”, por lo que se ha necesitado también cambiar el precio unitario, ahora referido a €/kg. Además, se puede comprobar que la medida obtenida coincide con la mostrada en la *figura 6-7: 126'94 kg*.

En cuanto a los rendimientos de los capítulos, sólo varía el rendimiento del capítulo 1 en comparación con la *tabla 6-2*, puesto que el único cambio se produce en el importe para el soporte, que pertenece al capítulo 1.

Así pues, se tienen los siguientes rendimientos:

CÓDIGO	RENDIMIENTO
01	3265,76207651847
02	36,8
03	140

Tabla 6-2: Presupuesto rendimientos tras modificación

6.4 Archivo bc3 generado.

Como ya sabemos, el objetivo de este trabajo es la obtención del archivo de texto FIEBDC que sea importable por cualquier otro programa de generación de presupuestos como el mencionado PRESTO.

Tras las modificaciones realizadas, el archivo de texto con el Formato de Intercambio Estándar de Bases de Datos para la Construcción (FIEBDC), se genera con los siguientes registros tipo:

```
~V|FIEBDC-3/2012|MEDyPRES||TFG|
~K|\3\3\3\2\2\2\EUR\0|
~C|TOTAL##|
~D|TOTAL##|01#\1\02#\1\03#\1|
```

~C|01#|Estructura principal|3265.76|31/07/2019|0|
 ~D|01#\0101\1\0.785\0102\1\126.941\0103\1\3.000\0104\1\2.000\|
 ~C|02#||Uniones|36.80|31/07/2019|0|
 ~D|02#\0201\1\0.240\0202\1\8.000\0203\1\0.060\|
 ~C|03#||Elementos de seguridad|140.00|31/07/2019|0|
 ~D|03#\0301\1\2.000\0302\1\2.000\|
 ~C|0101|m2|Base|250.00|18/07/2019|0|
 ~C|0102|kg|Soporte|10.00|18/07/2019|0|
 ~C|0103|m|Balancín|500.00|18/07/2019|0|
 ~C|0104|u|Asiento|150.00|18/07/2019|0|
 ~C|0201|m|Eje|100.00|18/07/2019|0|
 ~C|0202|u|Remache|0.10|18/07/2019|0|
 ~C|0203|m2|Plancha anclaje|200.00|18/07/2019|0|
 ~C|0301|u|Mango|50.00|18/07/2019|0|
 ~C|0302|u|Apoyo de goma|20.00|31/07/2019|0|
 ~T|0101|Base de hormigón que sirve para dotar de estabilidad a la estructura frente a la irregularidad del terreno|
 ~T|0102|Soporte de acero unido a la base y sobre el que descansa el mecanismo del balancín|
 ~T|0103|Barra de acero que se somete a momentos de balanceo|
 ~T|0104|Asiento de acero soldado a la barra de acero del balancín|
 ~T|0201|Eje de acero que dota de libertad de balanceo a la estructura|
 ~T|0202|Remache|
 ~T|0203|Plancha de acero para el anclaje del soporte a la base|
 ~T|0301|Mango|
 ~T|0302|Apoyo de goma de seguridad para reducir la fuerza del impacto contra el suelo|
 ~M|01#\0101|01\1\0.785||
 ~M|01#\0102|01\2\126.941||
 ~M|01#\0103|01\3\3.000||
 ~M|01#\0104|01\4\2.000||
 ~M|02#\0201|02\1\0.240||
 ~M|02#\0202|02\2\8.000||
 ~M|02#\0203|02\3\0.060||
 ~M|03#\0301|03\1\2.000||
 ~M|03#\0302|03\2\2.000||

En el apartado 4.3 ya se explicaron los distintos tipos de registros que se iban a usar en este archivo de texto “.bc3”.

En este apartado se verifica que para el caso práctico se cumple la estructura impuesta para este tipo

de archivos.

Los registros tipo propiedad (~V) y coeficiente (~K), son los mismos que se ejemplificaron en el mencionado apartado 4.3.

En los registros tipo concepto (~C), para los capítulos aparece primero el código del capítulo, después aparece su nombre, el rendimiento, y la fecha en la que se añadió al proyecto. Para los elementos medibles, si miramos de izquierda a derecha aparece: el código del elemento, la unidad establecida, el nombre del elemento, el coste unitario, y la fecha en la que se añadió al proyecto.

Para los registros tipo descomposición (~D), sólo tienen sentido los capítulos, que se desgranar en el código de cada elemento que contiene y en su medición.

En los registros tipo texto (~T), aparece el código que identifica cada elemento y posteriormente su descripción recogida de la base de datos.

Por último, en las mediciones (~M), se detalla el código del capítulo que contiene al elemento medido, el código del propio elemento, el lugar que ocupa dicho elemento en el capítulo, y el resultado de la medición realizada.

6.5 Importación del archivo bc3 desde Arquímedes.

Con el archivo de texto bc3 generado, procedemos a realizar la importación desde Arquímedes y se obtiene la siguiente disposición del ensamblaje:

Código	Doc.	Pli Ud	Resumen	Cant	Coste	Importe
BALANCÍN				1,000	3.442,46	3.442,46
01			Estructura principal	1,000	3.265,66	3.265,66
0101		m2	Base	0,785	250,00	196,25
0102		kg	Soporte	126,941	10,00	1.269,41
0103		m	Balancin	3,000	500,00	1.500,00
0104		u	Asiento	2,000	150,00	300,00
02			Uniones	1,000	36,80	36,80
0201		m	Eje	0,240	100,00	24,00
0202		u	Remache	8,000	0,10	0,80
0203		m2	Plancha anclaje	0,060	200,00	12,00
03			Elementos de seguridad	1,000	140,00	140,00
0301		u	Mango	2,000	50,00	100,00
0302		u	Apoyo de goma	2,000	20,00	40,00

Código	Ud	Resumen	Cant	Precio	Coste	Importe
0102	kg	Soporte	1,000	10,00	10,00	10,00
		Costes indirectos		10,00		
0102	kg	Soporte	126,941	10,00	10,00	1.269,41

Figura 6-21. Disposición del ensamblaje en Arquímedes

Por último, se van a ilustrar los documentos de las mediciones y el presupuesto generados por Arquímedes, lo cual era el objetivo final de este trabajo.

Así, tenemos que en primer lugar las mediciones se reflejarían con la siguiente estructuración:

Presupuesto parcial nº 1 Estructura principal

Nº	Ud	Descripción	Medición
1.1	M2	Base de hormigón que sirve para dotar de estabilidad a la estructura frente a la irregularidad del terreno	
			Total m2: 0,785
1.2	Kg	Soporte de acero unido a la base y sobre el que descansa el mecanismo del balancón	
			Total kg: 126,941
1.3	M	Barra de acero que se somete a momentos de balanceo	
			Total m: 3,000
1.4	U	Asiento de acero soldado a la barra de acero del balancón	
			Total u: 2,000

Figura 6-22. Medición capítulo 1

Presupuesto parcial nº 2 Uniones

Nº	Ud	Descripción	Medición
2.1	M	Eje de acero que dota de libertad de balanceo a la estructura	
			Total m: 0,240
2.2	U	Remache	
			Total u: 8,000
2.3	M2	Plancha de acero para el anclaje del soporte a la base	
			Total m2: 0,060

Figura 6-23. Medición capítulo 2

Presupuesto parcial nº 3 Elementos de seguridad

Nº	Ud	Descripción	Medición
3.1	U	Mango	
			Total u: 2,000
3.2	U	Apoyo de goma de seguridad para reducir la fuerza del impacto contra el suelo	
			Total u: 2,000

Sevilla, 9 de septiembre de 2019
 Ingeniero Aeroespacial
 José Luis Lucas Curado

Figura 6-24. Medición capítulo 3

Como resultado final, el documento con las mediciones y el presupuesto tiene el siguiente formato:

Presupuesto parcial nº 1 Estructura principal

Nº	Ud	Descripción	Medición	Precio	Importe
1.1	M2	Base de hormigón que sirve para dotar de estabilidad a la estructura frente a la irregularidad del terreno			
		Total m2	0,785	250,00	196,25
1.2	Kg	Soporte de acero unido a la base y sobre el que descansa el mecanismo del balanceo			
		Total kg	126,941	10,00	1.269,41
1.3	M	Barra de acero que se somete a momentos de balanceo			
		Total m	3,000	500,00	1.500,00
1.4	U	Asiento de acero soldado a la barra de acero del balanceo			
		Total u	2,000	150,00	300,00
Total presupuesto parcial nº 1 Estructura principal :					3.265,66

Figura 6-25. Presupuesto capítulo 1

Presupuesto parcial nº 2 Uniones

Nº	Ud	Descripción	Medición	Precio	Importe
2.1	M	Eje de acero que dota de libertad de balanceo a la estructura			
		Total m	0,240	100,00	24,00
2.2	U	Remache			
		Total u	8,000	0,10	0,80
2.3	M2	Plancha de acero para el anclaje del soporte a la base			
		Total m2	0,060	200,00	12,00
Total presupuesto parcial nº 2 Uniones :					36,80

Figura 6-26. Presupuesto capítulo 2

Presupuesto parcial nº 3 Elementos de seguridad

Nº	Ud	Descripción	Medición	Precio	Importe
3.1	U	Mango			
		Total u	2,000	50,00	100,00
3.2	U	Apoyo de goma de seguridad para reducir la fuerza del impacto contra el suelo			
		Total u	2,000	20,00	40,00
Total presupuesto parcial nº 3 Elementos de seguridad :					140,00

Figura 6-27. Presupuesto capítulo 3

Presupuesto de ejecución material

1 Estructura principal	3.265,66
2 Uniones	36,80
3 Elementos de seguridad	140,00
Total	3.442,46

Asciende el presupuesto de ejecución material a la expresada cantidad de TRES MIL CUATROCIENTOS CUARENTA Y DOS EUROS CON CUARENTA Y SEIS CÉNTIMOS.

Sevilla, 9 de septiembre de 2019
Ingeniero Aeroespacial
José Luis Lucas Curado

Figura 6-28. Presupuesto de ejecución material total

ANEXO CÓDIGO

Type UdO

codigo As String

unidad As String

precio As Double

medida As Double

importe As Double

End Type

Type Capitulo

codigo As String

rendimiento As Double

End Type

Dim ListaUdOs() As UdO

Dim ListaCapitulos() As Capitulo

Dim contador_codigos As Integer

Dim nombreEnsamblaje As String

Dim nombreFicheroSalida As String

Dim unidad As String

Dim precio As Double

Dim cod As String

Dim resumen As String

Dim fecha As String

Dim Texto As String

Dim EnsamblajeAc As ProductDocument

Dim partDocument1 As PartDocument

Dim part1 As Part

Dim myProduct As Product

Dim myParameters As Parameters

Sub recorrerconjunto()

```

contador_codigos = 1
nombreFicheroSalida="C:\Users\JoséLuis\Documents\5°
GIA\TFG\Ensamblaje\Recorrer_conjunto_prueba.bc3"

ReDim Preserve ListaUdOs(1)
  ListaUdOs(1).codigo = "00"
  ListaUdOs(1).unidad = ""
  ListaUdOs(1).precio = 0
  ListaUdOs(1).medida = 0
  ListaUdOs(1).importe = 0

GetNextNode CATIA.ActiveDocument.Product

  Debug.Print "CÓDIGO"; Tab(25); "UNIDAD"; Tab(45); "PRECIO"; Tab(60); "MEDIDA";
  Tab(80); "IMPORTE"
  Debug.Print "-----"
  For m = 2 To UBound(ListaUdOs)
    Debug.Print  ListaUdOs(m).codigo;  Tab(25);  ListaUdOs(m).unidad;  Tab(45);
  ListaUdOs(m).precio; Tab(60); ListaUdOs(m).medida; Tab(80); ListaUdOs(m).importe
  Next m

  obtenerCapitulos

  Debug.Print
  Debug.Print "CÓDIGO"; Tab(25); "RENDIMIENTO"
  Debug.Print "-----"
  For m = 1 To UBound(ListaCapitulos)
    Debug.Print ListaCapitulos(m).codigo; Tab(25); ListaCapitulos(m).rendimiento
  Next m

  'Debug.Print

  generarBC3

  'Eliminar las listas y liberar memoria
  Erase ListaUdOs

```

Erase ListaCapitulos

End Sub

Sub GetNextNode(oCurrentProduct As Product)

```
' Loop through every tree node for the current product
Dim oCurrentTreeNode As Product
Dim Contador_componentes As Integer
Dim i As Integer
Contador_componentes = oCurrentProduct.Products.Count
'MsgBox "El número de componentes es " & Contador_componentes
For i = 1 To Contador_componentes
  'MsgBox ("i = " & i)
  Set oCurrentTreeNode = oCurrentProduct.Products.Item(i)
  obtener_codigo oCurrentTreeNode

  If cod <> "NULO" Then
    conectar_BD
    'MsgBox "Componente = " & oCurrentTreeNode.PartNumber & "; Código = " & cod
    & "; UNIDAD = " & unidad
    localizarElemento oCurrentTreeNode
  Else
    'MsgBox "Componente = " & oCurrentTreeNode.PartNumber & "; Código = " & cod
    & ". Hay que atravesar componente"
    If oCurrentTreeNode.Products.Count > 0 Then
      GetNextNode oCurrentTreeNode
    End If
  End If

Next i
```

End Sub

Function medida_elemento(elemento_a_medir As Product) As Double

```
If unidad = "u" Then 'contar unidades
```

```
medida_elemento = 1
'MsgBox "El número de " & elemento_a_medir.PartNumber & " usados es " &
medida_elemento

ElseIf unidad = "m" Or unidad = "m2" Then 'medir longitud

Dim partdocumentL As PartDocument
Set partdocumentL = CATIA.Documents.Open("C:\Users\JoséLuis\Documents\5°
GIA\TFG\Ensamblaje\" & elemento_a_medir.PartNumber & ".CATPart")

Dim partL As Part
Set partL = partdocumentL.Part

Dim bodies1 As Bodies
Set bodies1 = partL.Bodies

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

Dim oSelp As Selection
Set oSelp = CATIA.ActiveDocument.Selection

oSelp.Clear
If unidad = "m" Then
oSelp.Search "Name = PuntoLong" & elemento_a_medir.PartNumber & ".*,all"
ElseIf unidad = "m2" Then
oSelp.Search "Name = PuntoArea" & elemento_a_medir.PartNumber & ".*,all"
End If
'MsgBox "El número de puntos seleccionados es " & oSelp.Count
If oSelp.Count > 0 Then

Dim j As Integer
Dim contador_puntos As Integer
contador_puntos = 0
Dim origen(2)
```

```
Dim anterior(2)
```

```
Dim longitud As Double
```

```
longitud = 0
```

```
Dim AreaDada As Double
```

```
AreaDada = 0
```

```
For j = 1 To oSelp.Count
```

```
    Dim oSelEl As SelectedElement
```

```
    Set oSelEl = oSelp.Item(j)
```

```
    'MsgBox "Name " & oSelEl.Name
```

```
    Dim coordenadas(2)
```

```
    oSelEl.Value.GetCoordinates coordenadas
```

```
    If contador_puntos = 0 Then
```

```
        origen(0) = coordenadas(0)
```

```
        origen(1) = coordenadas(1)
```

```
        origen(2) = coordenadas(2)
```

```
        anterior(0) = origen(0)
```

```
        anterior(1) = origen(1)
```

```
        anterior(2) = origen(2)
```

```
        contador_puntos = contador_puntos + 1
```

```
    Else
```

```
        longitud = longitud + Sqr((coordenadas(0) - anterior(0)) * (coordenadas(0) - anterior(0)) + (coordenadas(1) - anterior(1)) * (coordenadas(1) - anterior(1)) + (coordenadas(2) - anterior(2)) * (coordenadas(2) - anterior(2)))
```

```
        AreaDada = AreaDada + (coordenadas(0) - anterior(0)) * (coordenadas(1) + anterior(1))
```

```
        anterior(0) = coordenadas(0)
```

```
        anterior(1) = coordenadas(1)
```

```
        anterior(2) = coordenadas(2)
```

```
        contador_puntos = contador_puntos + 1
```

```
        'MsgBox "longitud parcial = " & longitud * 0.001 & " m"
```

```
    End If
```

```
Next j
```

```
If unidad = "m" Then
```

```

Dim TIPO_LONG As String
TIPO_LONG = myParameters.Item(part1.Name & "\Properties\TIPO
LONGITUD").ValueAsString
If TIPO_LONG = "CERRADA" Then
    longitud = longitud + Sqr((origen(0) - anterior(0)) * (origen(0) - anterior(0)) +
(origen(1) - anterior(1)) * (origen(1) - anterior(1)) + (origen(2) - anterior(2)) * (origen(2) -
anterior(2)))
    medida_elemento = longitud * 0.001
    MsgBox "longitud total = " & medida_elemento & " m"

ElseIf TIPO_LONG = "ABIERTA" Then
    medida_elemento = longitud * 0.001
    MsgBox "longitud total definida = " & medida_elemento & " m"
End If
ElseIf unidad = "m2" Then
    AreaDada = AreaDada + (origen(0) - anterior(0)) * (origen(1) + anterior(1))
    AreaDada = Abs(AreaDada * 0.5)
    medida_elemento = AreaDada * 0.000001
    MsgBox "Area total definida = " & medida_elemento & " m2"
End If

oSelp.Clear
partdocumentL.Close

ElseIf oSelp.Count = 0 And unidad = "m2" Then 'medir área

'Dim partdocument2 As PartDocument
'Set partdocument2 = CATIA.Documents.Open("C:\Users\JoséLuis\Documents\5°
GIA\TFG\Ensamblaje\" & elemento_a_medir.PartNumber & ".CATPart")

Dim oSelcara As Selection
Set oSelcara = partdocumentL.Selection

oSelp.Search "Type = Face,all"

```

```
Dim NumCara As Integer
NumCara = oSelcara.Count
'MsgBox "El número total de caras es de " & NumCara 'Cuento el número total de caras

Dim cont_caras As Integer
Dim Cara As SelectedElement

Dim spabench As SPAWorkbench
Dim mymeas As Measurable
Dim refl As Reference
Dim myans As Double
Dim Area As Double

For cont_caras = 1 To NumCara
Set Cara = oSelcara.Item(cont_caras)

'MsgBox "Nombre: " & Cara.Value.Name

'Dim Area(NumCara)

Set refl = oSelcara.Item(cont_caras).Reference
Set spabench = partdocumentL.GetWorkbench("SPAWorkbench")
Set mymeas = spabench.GetMeasurable(refl)
myans = mymeas.Area
If cont_caras = 1 Then
medida_elemento = myans
Else
If myans > medida_elemento Then
medida_elemento = myans
Else
medida_elemento = medida_elemento
End If
End If
Next cont_caras
partdocumentL.Close
```

```
oSelcara.Clear
```

```
MsgBox "El área de la cara mayor es: " & medida_elemento & " m2"
```

```
End If
```

```
ElseIf unidad = "m3" Or unidad = "kg" Then 'medir volumen
```

```
Set objRef = part1.CreateReferenceFromObject(part1.MainBody)
```

```
Set objSPAWkb = CATIA.ActiveDocument.GetWorkbench("SPAWorkbench")
```

```
Set objMeasurable = objSPAWkb.GetMeasurable(objRef)
```

```
'MsgBox objMeasurable.Volume
```

```
Dim Volumen As Double
```

```
Volumen = objMeasurable.Volume 'Expresado en m3
```

```
If unidad = "m3" Then
```

```
medida_elemento = Volumen
```

```
MsgBox "El volumen del cuerpo es de " & Volumen & " m3 "
```

```
Else
```

```
Dim DENS
```

```
DENS = myParameters.Item(part1.Name & "\Properties\DENSIDAD").Value
```

```
medida_elemento = Volumen * CInt(DENS)
```

```
MsgBox "La masa es de " & medida_elemento & " kg."
```

```
End If
```

```
End If
```

```
End Function
```

```
Function IsPart(oCurrentProduct As Product) As Boolean
```

```
Dim oTestPart As PartDocument
```

```
Set oTestPart = Nothing
```

```
On Error Resume Next
```

```
Set oTestPart = CATIA.Documents.Item(oCurrentProduct.PartNumber & ".CATPart")
```

```
If Not oTestPart Is Nothing Then
```

```
    IsPart = True
```

```
Else
```

```
    IsPart = False
```

```
End If
```

End Function

Function IsProduct(oCurrentProduct As Product) As Boolean

```
Dim oTestProduct As ProductDocument
```

```
Set oTestProduct = Nothing
```

```
On Error Resume Next
```

```
Set oTestProduct = CATIA.Documents.Item(oCurrentProduct.PartNumber & ".CATProduct")
```

```
If Not oTestProduct Is Nothing Then
```

```
    IsProduct = True
```

```
Else
```

```
    IsProduct = False
```

```
End If
```

End Function

Sub conectar_BD()

```
Dim BaseDatos As String
```

```
Dim oCon, oCom, oRec
```

```
BaseDatos = "DBQ=" & C:\Users\JoséLuis\Documents\5º  
GIA\TFG\Ensamblaje\BasedeCostes.mdb; "
```

```
Set oCon = CreateObject("ADODB.Connection")
```

```
oCon.Open BaseDatos & "DRIVER={Microsoft Access Driver (*.mdb)}"
```

```
Set oCom = CreateObject("ADODB.Command")
Set oRec = CreateObject("ADODB.Recordset")
Dim sentenciaSQL As String
oCom.CommandText = "SELECT UNIDAD, PRECIO_UNITARIO, DESCRIPCION FROM
Tabladecostes WHERE CÓDIGO = " & cod & """"
oCom.ActiveConnection = oCon
Set oRec = oCom.Execute
'Debug.Print cod
unidad = Trim(oRec("UNIDAD").Value)
precio = Trim(oRec("PRECIO_UNITARIO").Value)
'descripcion = Trim(oRec("DESCRIPCION").Value

'Debug.Print "Unidad = " & unidad
'Debug.Print "Precio = " & precio
oRec.Close
oCon.Close
Set oCon = Nothing
Set oCom = Nothing
Set oRec = Nothing
```

End Sub

Sub obtener_codigo(nodo As Product)

```
' Determine if the current node is a part, product or component
If IsPart(nodo) = True Then
'MsgBox oCurrentTreeNode.PartNumber & " is a part"
'Dim partDocument1 As PartDocument
Set partDocument1 = CATIA.Documents.Item(nodo.PartNumber & ".CATPart")
'Set partdocument1 = oCurrentTreeNode.GetMasterShapeRepresentation(True)

'Dim part1 As Part
Set part1 = partDocument1.Part

'Dim myProduct As Product
```

```
Set myProduct = CATIA.ActiveDocument.GetItem(part1.Name)

'Dim myParameters As Parameters
Set myParameters = myProduct.UserRefProperties

cod = myParameters.Item(part1.Name & "\Properties\CÓDIGO").ValueAsString

ElseIf IsProduct(nodo) = True Then
  'MsgBox oCurrentTreeNode.PartNumber & " is a product"

  Dim productdocument1 As ProductDocument
  Set productdocument1 = CATIA.Documents.Item(nodo.PartNumber & ".CATProduct")

  Dim product1 As Product
  Set product1 = productdocument1.Product

  Dim parameters1 As Parameters
  Set parameters1 = product1.UserRefProperties

  cod = parameters1.Item(product1.Name & "\Properties\CÓDIGO").ValueAsString
  'MsgBox "Código = " & cod

Else
  'MsgBox oCurrentTreeNode.PartNumber & " is a component"
End If
```

End Sub

Sub localizarElemento(elemento_a_localizar As Product)

'Localizar elementos con el mismo código

```
Dim elementoEncontrado As Boolean
Dim contador As Integer
Dim indice, posicion As Integer
```

```

elementoEncontrado = False
contador = 1
indice = 1
posicion = 1

'MsgBox "Lista de Unidades de Obra " & UBound(ListaUdOs)
'Debug.Print UBound(ListaUdOs) // sale 1 porque no he generado ninguna lista de
elementos como tal, por eso resultado erróneo.
While (Not elementoEncontrado And (contador <= UBound(ListaUdOs)))
  If (cod = ListaUdOs(contador).codigo) Then
    elementoEncontrado = True
    ListaUdOs(contador).medida      =      ListaUdOs(contador).medida      +
medida_elemento(elemento_a_localizar)
    ListaUdOs(contador).importe    =      ListaUdOs(contador).medida      *
ListaUdOs(contador).precio
  Else
    contador = contador + 1
  End If
Wend
If (Not elementoEncontrado) Then
  contador_codigos = contador_codigos + 1
  contador = UBound(ListaUdOs) + 1
  ReDim Preserve ListaUdOs(contador)
  Do While (indice <= contador - 1)
    If cod > ListaUdOs(indice).codigo Then posicion = posicion + 1
    indice = indice + 1
  Loop
  Do While (contador > posicion)
    ListaUdOs(contador) = ListaUdOs(contador - 1)
    contador = contador - 1
  Loop
  ListaUdOs(posicion).codigo = cod
  ListaUdOs(posicion).unidad = unidad
  ListaUdOs(posicion).precio = precio
  ListaUdOs(posicion).medida = medida_elemento(elemento_a_localizar)
  ListaUdOs(posicion).importe      =      ListaUdOs(posicion).medida      *
ListaUdOs(posicion).precio

```

```
End If
```

```
End Sub
```

```
Sub obtenerCapitulos()
```

```
Dim cont_cap As Integer
```

```
Dim i As Integer
```

```
Dim cod_cap As String
```

```
cont_cap = 0
```

```
For i = 2 To UBound(ListaUdOs)
```

```
cod_cap = Left(ListaUdOs(i).codigo, 2)
```

```
If cont_cap = 0 Then
```

```
ReDim Preserve ListaCapitulos(1)
```

```
ListaCapitulos(1).codigo = cod_cap
```

```
calcularRendimiento cod_cap, 1
```

```
Else
```

```
localizarCapitulo cod_cap
```

```
End If
```

```
cont_cap = cont_cap + 1
```

```
Next i
```

```
End Sub
```

```
Sub localizarCapitulo(cod As String)
```

```
'Localizar elementos con el mismo código
```

```
Dim elementoEncontrado As Boolean
```

```
Dim contador As Integer
```

```
Dim indice, posicion As Integer
```

```
elementoEncontrado = False
```

```
contador = 1
```

```
indice = 1
```

```
posicion = 1
```

```
While (Not elementoEncontrado And (contador <= UBound(ListaCapitulos)))
```

```
If (cod = ListaCapitulos(contador).codigo) Then
```

```
    elementoEncontrado = True
Else
    contador = contador + 1
End If
Wend
If (Not elementoEncontrado) Then
    contador = UBound(ListaCapitulos) + 1
    ReDim Preserve ListaCapitulos(contador)
    Do While (indice <= contador - 1)
        If cod > ListaCapitulos(indice).codigo Then posicion = posicion + 1
        indice = indice + 1
    Loop
    Do While (contador > posicion)
        ListaCapitulos(contador) = ListaCapitulos(contador - 1)
        contador = contador - 1
    Loop
    ListaCapitulos(posicion).codigo = cod
    calcularRendimiento cod, posicion
End If
```

End Sub

Sub calcularRendimiento(codig As String, pos As Integer)

```
    Dim i As Integer
    Dim rend As Double

    rend = 0
    For i = 1 To UBound(ListaUdOs)
        If Left(ListaUdOs(i).codigo, 2) = codig Then rend = rend + ListaUdOs(i).importe
    Next i
    ListaCapitulos(pos).rendimiento = rend
```

End Sub

Sub generarBC3()


```
    Debug.Print cadena
Next i

'Conceptos asociados a unidades de obra
For i = 2 To UBound(ListaUdOs)
    recuperarDatosConceptos ListaUdOs(i).codigo
    Print #1, Chr(126) & "C|" & ListaUdOs(i).codigo & "|" & ListaUdOs(i).unidad & "|" &
resumen & "|" & FormatNumber(ListaUdOs(i).precio, 2) & "|" & fecha & "|0|"
    Debug.Print Chr(126) & "C|" & ListaUdOs(i).codigo & "|" & ListaUdOs(i).unidad & "|" &
resumen & "|" & FormatNumber(ListaUdOs(i).precio, 2) & "|" & fecha & "|0|"
Next i

'Textos
For i = 2 To UBound(ListaUdOs)
    recuperarTextos ListaUdOs(i).codigo
    Print #1, Chr(126) & "T|" & ListaUdOs(i).codigo & "|" & Texto & "|"
    Debug.Print Chr(126) & "T|" & ListaUdOs(i).codigo & "|" & Texto & "|"
Next i

'Mediciones
For i = 1 To UBound(ListaCapitulos)
    orden = 1
    For j = 2 To UBound(ListaUdOs)
        If Left(ListaUdOs(j).codigo, 2) = ListaCapitulos(i).codigo Then
            Print #1, Chr(126) & "M|" & ListaCapitulos(i).codigo & "#" & ListaUdOs(j).codigo &
|" & ListaCapitulos(i).codigo & "\" & orden & "\" & FormatNumber(ListaUdOs(j).medida, 3)
& "||"
            Debug.Print Chr(126) & "M|" & ListaCapitulos(i).codigo & "#" &
ListaUdOs(j).codigo & "|" & ListaCapitulos(i).codigo & "\" & orden & "\" &
FormatNumber(ListaUdOs(j).medida, 3) & "||"
            orden = orden + 1
        End If
    Next j
Next i

'Cerrar el fichero de salida
Close #1
```

End Sub

Sub recuperarDatosConceptos(cod As String)

```
Dim oCon, oCom, oRec
BaseDatos = "DBQ=" & C:\Users\JoséLuis\Documents\5º
GIA\TFG\Ensamblaje\BasedeCostes.mdb; "

Set oCon = CreateObject("ADODB.Connection")
oCon.Open BaseDatos & "DRIVER={Microsoft Access Driver (*.mdb)}"

Set oCom = CreateObject("ADODB.Command")
Set oRec = CreateObject("ADODB.Recordset")
oCom.CommandText = "SELECT RESUMEN, FECHA FROM Tabladecostes WHERE
CÓDIGO = " & cod & ""
'MsgBox "El código para recuperar Resumen es " & cod 'El código obtenido es 1, por eso da
error.
oCom.ActiveConnection = oCon
Set oRec = oCom.Execute
resumen = Trim(oRec("RESUMEN").Value)
fecha = Trim(oRec("FECHA").Value)
oRec.Close
oCon.Close
Set oCon = Nothing
Set oCom = Nothing
Set oRec = Nothing
```

End Sub

Sub recuperarTextos(cod As String)

```
Dim oCon, oCom, oRec
BaseDatos = "DBQ=" & C:\Users\JoséLuis\Documents\5º
GIA\TFG\Ensamblaje\BasedeCostes.mdb; "

Set oCon = CreateObject("ADODB.Connection")
oCon.Open BaseDatos & "DRIVER={Microsoft Access Driver (*.mdb)}"
```

```
Set oCom = CreateObject("ADODB.Command")
Set oRec = CreateObject("ADODB.Recordset")
oCom.CommandText = "SELECT DESCRIPCION FROM Tabladecostes WHERE CÓDIGO
= " & cod & ""
oCom.ActiveConnection = oCon
Set oRec = oCom.Execute
Texto = Trim(oRec("DESCRIPCION").Value)
oRec.Close
oCon.Close
Set oCon = Nothing
Set oCom = Nothing
Set oRec = Nothing
```

End Sub

BIBLIOGRAFÍA

- 1) UNE 157001:2014
Es la norma que recoge la documentación a aportar para acometer una obra de ingeniería.
- 2) Tema 2: Documentos de un proyecto técnico (Proyectos de Ingeniería Aeroespacial).
Tema 6: Mediciones y Presupuesto (Proyectos de Ingeniería Aeroespacial).
Usado como base teórica para describir la información recogida en los documentos de una obra de ingeniería.
- 3) Libro VB Scripting for CATIA V5: how to program catia macros, Ross Emmett, 2015.
Este libro ha sido mi toma de contacto con la programación en Visual Basic en el entorno de CAD 3D. Describe las funciones básicas y pone ejemplos de macros sencillas con las que empezar a practicar.
- 4) <http://www.coe.org> : Community Of Experts of Dassault Systèmes Solutions.
En esta web he encontrado respuestas sobre problemas y errores obtenidos durante la programación en Visual Basic.
- 5) Proyecto Fin de Grado. Fundamentos de programación de VBA en Catia V5: Aplicación al diseño de Rodamientos, José Miguel Peña Rodríguez, 2015.
Ha servido como apoyo en el aprendizaje sobre la programación en Visual Basic.
- 6) www.fiebdc.es :
En esta web se accede a la información sobre la constitución del formato FIEBDC y a sus características.
- 7) <http://presupuestodeobra.com/que-es-el-presupuesto-de-obra.html>:
De esta web se ha recabado información sobre los datos que se deben proporcionar en un presupuesto de obra.
- 8) <https://www.rae.es/>:
De la Real Academia Española se han extraído definiciones literales.

