

Trabajo Fin de Máster

Ingeniería Industrial

Valor alternativo al valor de Shapley para el problema del aeropuerto

Autor: María del Valle Delgado Jiménez

Tutor: Manuel Ordoñez Sánchez

Dpto. Matemática Aplicada II
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2022



Trabajo Fin de Máster
Ingeniería Industrial

Valor alternativo al valor de Shapley para el problema del aeropuerto

Autor:

María del Valle Delgado Jiménez

Tutor:

Manuel Ordoñez Sánchez

Dpto. Matemática Aplicada II

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2022

Trabajo Fin de Máster: Valor alternativo al valor de Shapley para el problema del aeropuerto

Autor: María del Valle Delgado Jiménez

Tutor: Manuel Ordoñez Sánchez

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2022

El Secretario del Tribunal

AGRADECIMIENTOS

En primer lugar, a mi familia, por darme siempre su apoyo incondicional y brindarme todas las oportunidades a su alcance. También a mis amigas y amigos, quienes tan bien me animan y creen en mí.

Por otro lado, me gustaría agradecer a mi tutor, Manuel Ordoñez, su trabajo. Ya que desde el primer día me ha demostrado su confianza e implicación, estando atento a cualquier inconveniente o ayuda que pudiese necesitar.

Asimismo, a la Escuela, pues ha contribuido e impulsado mi desarrollo tanto a nivel personal como profesional a lo largo de todos estos años.

RESUMEN

En este documento se presenta la comparación de dos índices, uno ampliamente extendido, Shapley, y otro nuevo, Simplicial, aplicados al problema clásico de distribución de costes del aeropuerto.

Este análisis, se ha realizado a partir de una simulación práctica utilizando como modelo el aeropuerto Adolfo Suárez Madrid-Barajas, ya que es una infraestructura icónica y de gran envergadura dentro del contexto aéreo nacional. Dicha aplicación, se basa en los fundamentos teóricos establecidos por la rama matemática de la Teoría de Juegos, a la que pertenece la familia de los juegos cooperativos de coalición basados en utilidad transferible, entre los que se encuentra el juego del aeropuerto.

La clave de estos juegos se centra en el poder del consenso como modelo para asumir una determinada distribución de costes, los cuales de manera independiente serían más complejos de afrontar. Por lo que en un contexto como el actual, donde por motivos sociopolíticos y económicos, el mercado está sufriendo unas consecuencias históricas, sobretodo en relación a los incrementos en las materias primas, y por ende en el transporte, resulta interesante analizar otras perspectivas que permitan flexibilizar las soluciones eficientes ya existentes. Y de este modo, posibilitar otros modelos de negocio cuyos principios teóricos puedan ser también justificados desde un conjunto menos restrictivo de la Teoría de Juegos.

La alternativa desgranada en este documento, la cual ha sido automatizada por medio de un programa adjuntado en los anexos del mismo, demuestra la utilidad de otros índices como el Simplicial. De manera que, mediante el juego del aeropuerto, se expone pormenorizadamente el cálculo no lineal de dicho índice desestimando el axioma de linealidad que impone Shapley.

ABSTRACT

This paper presents a comparison of the widely used Shapley index and the new Simplicial value applied to the classical airport cost allocation problem.

This analysis has been performed on the basis of a practical simulation using Adolfo Suárez Madrid-Barajas airport as a model, since it is an iconic and large-scale infrastructure within the national aviation context. This application is based on the theoretical foundations established by the mathematical branch of Game Theory, to which the family of cooperative coalition games based on transferable utility belongs, including the airport game.

The key to these games lies in the power of consensus as a model for assuming a certain distribution of costs, which would be more difficult to deal with on their own. Therefore, in a context such as the current one, where, for socio-political and economic reasons, the market is suffering historic consequences, especially in relation to increases in raw materials, and therefore in transport, it is interesting to analyse other perspectives that allow existing efficient solutions to be made more flexible. In this way, other business models whose theoretical principles can also be justified from a less restrictive set of Game Theory can be made possible.

The alternative described in this document, which has been automated by means of a programme attached in the annexes, demonstrates the usefulness of other values such as Simplicial. Thus, by means of the airport game, the non-linear calculation of this index is presented in detail, rejecting the linearity axiom imposed by Shapley.

ÍNDICE

Agradecimientos	v
Resumen	vi
Abstract	vii
Índice	viii
Índice de Tablas	x
Índice de Figuras	xii
1 Introducción	1
1.1. <i>Antecedentes y Motivación</i>	1
1.2. <i>Objetivos y Alcance</i>	3
1.3. <i>Estructura del documento</i>	3
2 Estado del arte	5
2.1. <i>Teoría de juegos cooperativos</i>	5
2.1.1. <i>Juego de distribución de costes</i>	8
2.2. <i>Aeropuerto de Barajas</i>	8
2.2.1. <i>Subsistema de Movimiento de Aeronaves</i>	10
2.3. <i>Herramienta utilizada</i>	13
3 Fundamento teórico	17
3.1. <i>Solución de los juegos de coalición</i>	17
3.2. <i>Valor de Shapley</i>	17
3.3. <i>Valor Simplicial</i>	18
4 Cálculos del juego inicial	21
4.1. <i>Datos de partida: División de secciones</i>	21
4.2. <i>Repartición base de costes</i>	22
5 Aplicación y comparación	27
5.1. <i>Distribución equivalente al valor de Shapley</i>	27
5.2. <i>Distribución según valor Simplicial</i>	28
5.3. <i>Comparación de las distribuciones</i>	35

6 Programa desarrollado	37
6.1. <i>Estructura programa</i>	37
6.2. <i>Librerías</i>	39
6.3. <i>Funciones</i>	40
7 Conclusiones	43
7.1. <i>Conclusiones del trabajo realizado</i>	43
7.2. <i>Líneas futuras</i>	44
Referencias	45
Anexo A: Códigos	47
Anexo B: Planificación	65
Anexo C: Presupuesto	71

ÍNDICE DE TABLAS

<i>Tabla 2-1. Dimensiones características de las pistas del aeropuerto Madrid-Barajas [10].</i>	10
<i>Tabla 2-2. Cantidad y modelos máximos de aeronaves admisibles.</i>	11
<i>Tabla 2-3. Características de los modelos de aeronaves [11]–[13].</i>	12
<i>Tabla 2-4. Ranking de lenguajes de programación según el índice TIOBE [14].</i>	13
<i>Tabla 4-1. Agrupación de los modelos de aeronave en función de la TOR necesaria.</i>	22
<i>Tabla 4-2. Costes totales de las secciones de la pista de despegue de la configuración Norte.</i>	23
<i>Tabla 4-3. Costes totales de las secciones de la pista de despegue de la configuración Sur.</i>	23
<i>Tabla 4-4. Distribución equitativa de los costes de las secciones CN por categorías de aeronaves.</i>	24
<i>Tabla 4-5. Distribución equitativa de los costes de las secciones CS por categorías de aeronaves.</i>	24
<i>Tabla 4-6. Fracción en tanto por uno de los costes de las secciones CN por categorías de aeronaves.</i>	24
<i>Tabla 4-7. Fracción en tanto por uno de los costes de las secciones CS por categorías de aeronaves.</i>	25
<i>Tabla 5-1. Costes totales para cada categoría de aeronaves en la configuración Norte.</i>	27
<i>Tabla 5-2. Costes totales para cada categoría de aeronaves en la configuración Sur.</i>	27
<i>Tabla 5-3. Valor representativo tradicional equivalente a Shapley para cada categoría de aeronaves en la CN.</i>	28
<i>Tabla 5-4. Valor representativo tradicional equivalente a Shapley para cada categoría de aeronaves en la CS.</i>	28
<i>Tabla 5-5. Número de coaliciones posibles asociados a cada sección de ambas configuraciones.</i>	29
<i>Tabla 5-6. Coaliciones y funciones correspondientes para el juego C de ambas configuraciones.</i>	29
<i>Tabla 5-7. Coaliciones y funciones correspondientes para el juego A de ambas configuraciones.</i>	30
<i>Tabla 5-8. Coaliciones y funciones correspondientes para el juego B de ambas configuraciones.</i>	31
<i>Tabla 5-9. Coaliciones y funciones correspondientes para el juego D de ambas configuraciones.</i>	31
<i>Tabla 5-10. Coaliciones y funciones correspondientes para el juego E de ambas configuraciones.</i>	32
<i>Tabla 5-11. Coaliciones y funciones correspondientes para el juego F de ambas configuraciones.</i>	32
<i>Tabla 5-12. Resumen de las coaliciones posibles y sus funciones en el juego C.</i>	33

<i>Tabla 5-13. Valores por etapa de cada jugador para ambas configuraciones.</i>	<i>34</i>
<i>Tabla 5-14. Valores calculados por etapa de cada jugador para ambas configuraciones.</i>	<i>34</i>
<i>Tabla 5-15. Valor Simplicial para cada categoría de aeronaves en la configuración Norte.</i>	<i>34</i>
<i>Tabla 5-16. Valor Simplicial para cada categoría de aeronaves en la configuración Sur.</i>	<i>35</i>
<i>Tabla A-1. Distribución horaria de los paquetes de trabajo de este TFM.</i>	<i>67</i>
<i>Tabla A-2. Coste de todos los conceptos de este TFM.....</i>	<i>73</i>
<i>Tabla A-3. Presupuesto resumen de este TFM.</i>	<i>73</i>

ÍNDICE DE FIGURAS

<i>Figura 1–1. Tráfico de aeronaves en España [3].</i>	2
<i>Figura 1–2. Tráfico de aeronaves en el aeropuerto AS Madrid-Barajas [3].</i>	2
<i>Figura 2–1. Conjunto de aeropuertos en España [8].</i>	9
<i>Figura 2–2. Plano de las configuraciones de pistas del aeropuerto AS Madrid-Barajas [9].</i>	10
<i>Figura 2–3. Evolución de la tendencia del índice TIOBE para diversos lenguajes de programación [14].</i> .	13
<i>Figura 3–1. Representación del símplice.</i>	19
<i>Figura 5–1. Representación de las coaliciones de dimensión 0 del juego C.</i>	33
<i>Figura 5–2. Representación de las coaliciones de dimensión 1 del juego C.</i>	33
<i>Figura 5–3. Representación de algunas de las coaliciones de dimensión 2 del juego C.</i>	33
<i>Figura 5–4. Distribución porcentual de la distribución de costes según Shapley.</i>	35
<i>Figura 5–5. Distribución porcentual de la distribución de costes según valor Símplice.</i>	36
<i>Figura 6–1. Diagrama de flujo del programa desarrollado.</i>	38
<i>Figura A–1. Distribución porcentual de las horas reales invertidas en los diferentes paquetes de trabajo de este TFM.</i>	68
<i>Figura A–2. Distribución porcentual de los costes de este TFM.</i>	73

1 INTRODUCCIÓN

En este primer capítulo se aborda tanto la justificación como el contexto de este trabajo. Además, se enuncia el alcance y objetivos del mismo.

Conjuntamente, en el último apartado, se expone la estructura del contenido del documento con objeto de facilitar y guiar la lectura de este.

1.1. Antecedentes y Motivación

El juego del aeropuerto, aunque es un problema que se ha analizado desde diferentes perspectivas en múltiples ocasiones, es un ejemplo muy conocido con el que ilustrar posibles distribuciones de costes de un servicio común.

En un contexto como el actual, donde la sociedad está sufriendo las consecuencias (como en otros momentos de la historia) de las tensiones políticas, causadas en su mayoría por la guerra entre Ucrania y Rusia, tiene mayor interés abordar la problemática de la distribución de costes. Ya que, con un mercado económico a la baja, las empresas centran sus esfuerzos en reducir costes para salvar sus dividendos.

En el caso de las compañías aéreas son diversos los factores que influyen en su contabilidad, la cual en último lugar, repercute sobre el pasajero. Entre todos ellos, hoy en día, destacan: el incremento de precio y la escasez de materias primas, el auge de la concienciación social por la reducción de la contaminación, y finalmente, la intensificación de la globalización.

La carencia de materias primas, como el combustible, y la dependencia energética entre los países, se está traduciendo en un aumento de los precios que está disparando los costes de las aerolíneas. Pues el fuel es un elemento principal que forma parte de la cadena de valor desde el principio, por lo que, existen incrementos en todos los eslabones que requieren de transporte.

A este acrecentamiento de los costes de operación, se le suman las tasas impuestas desde los diferentes organismos políticos (ya sea a nivel estatal o comunitario) con objeto de compensar la contaminación emitida por su actividad. De hecho, el transporte aéreo, en proporción con sus desplazamientos, es el medio que emite mayores cantidad de dióxido de carbono a la atmósfera [1]. Lo cual contribuye negativamente al calentamiento global, siendo este cada vez más crítico y difícil de ralentizar, como ha quedado patente en la última cumbre mundial del clima celebrada en Egipto (COP27) [2].

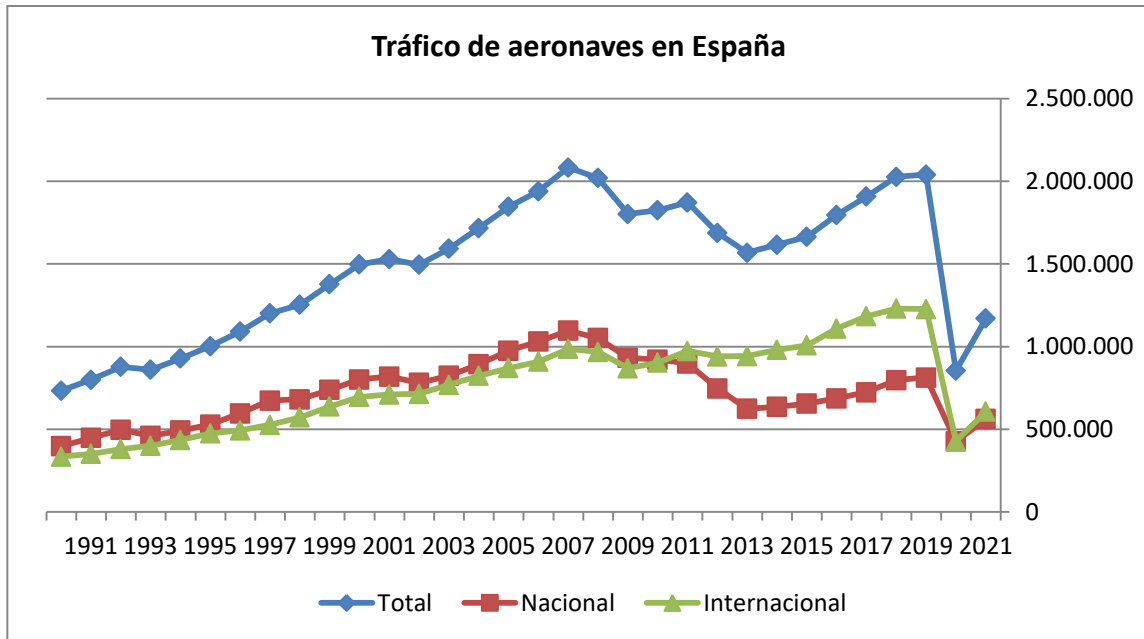


Figura 1–1. Tráfico de aeronaves en España [3].

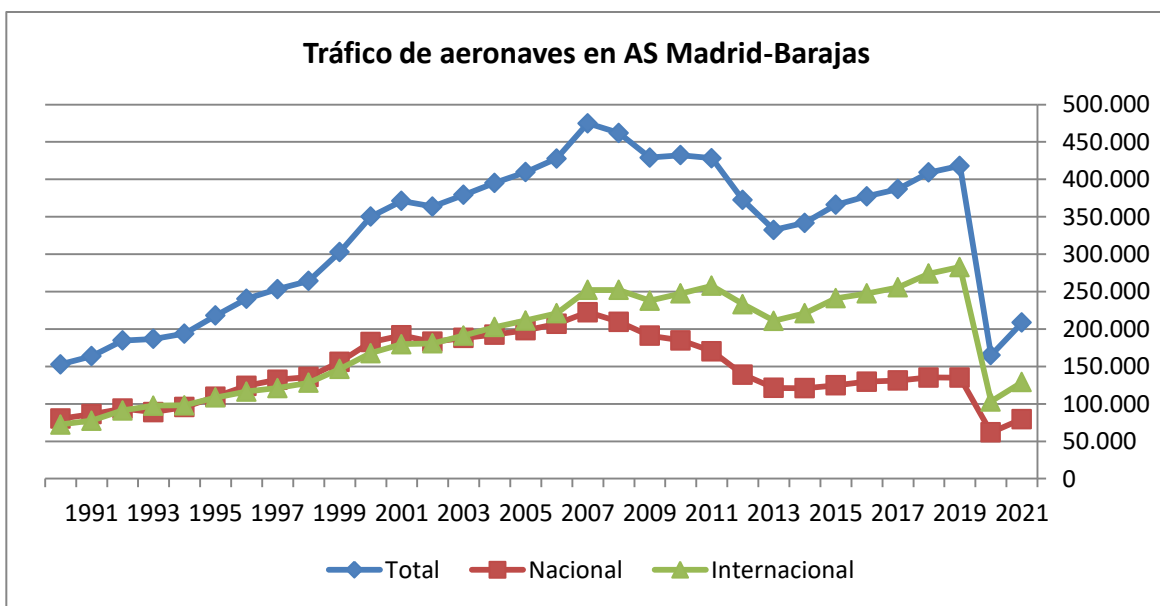


Figura 1–2. Tráfico de aeronaves en el aeropuerto AS Madrid-Barajas [3].

Por otro lado, en un mundo cada vez más globalizado, es indispensable este tipo de transporte para la comunicación. Aunque es una realidad que a raíz de la pandemia los viajes por motivos profesionales se han reducido debido al progreso de la digitalización y las nuevas tecnologías, son cada vez más las expediciones realizadas desde el ámbito del ocio, para las cuales no se dispone de mucho tiempo, lo que potencia el uso de esta clase de transporte sobretodo en destinos internacionales (pues todavía en el ámbito nacional se pueden utilizar otros medios alternativos más respetuosos con el medio ambiente, a la vez que admisibles en tiempo y/o presupuesto) como se puede observar en las siguientes figuras.

No obstante, aunque sus valores varíen, la división de los costes entre los diferentes modelos de aeronave que operen ya ha sido abordada. De hecho, en 1973, A. Littlechild y G. Owen propusieron una distribución lineal que coincide con el valor de Shapley del siguiente juego para un conjunto de

aeronaves N :

$$c(S) = \max_{i \in S} \{c_i\} \text{ para toda } S \subset N$$

Por otra parte, cabe destacar que esta problemática centrada en el binomio aeropuertos-aerolíneas como el mayor servicio comunitario de transporte, se puede extrapolar a otros modelos de mercado que están surgiendo hoy en día como respuesta a las demandas sociales. En ellos, la población ya no va a ser propietaria de vehículos (para pasajeros o mercancías), sino que su papel se transforma en usuario de un servicio común, cuyos costes tendrán que ser distribuidos del modo más justo posible como retribución de la asistencia prestada según las modalidades disponibles.

En definitiva, el juego del aeropuerto es un ejemplo que a pesar de haber sido empleado reiteradamente, es un paradigma muy ilustrativo de la distribución de costes de una utilidad y/o infraestructura común, que aunque ya ha sido resuelto de manera eficiente, resulta interesante abordarlo desde otro punto de vista (como el expuesto en este trabajo). Pues como se ha mencionado, además de representar un modelo de mercado de un sector en auge, este se puede extrapolar a otros ámbitos socio-económicos de características similares.

1.2. Objetivos y Alcance

Como se ha indicado en el apartado anterior, el juego del aeropuerto ya ha sido resuelto de forma eficiente como la solución de Littlechild y Owen coincidente con el valor de Shapley. Sin embargo, este es un valor cuyos axiomas imponen restricciones muy fuertes sobre la distribución de pagos posible.

De modo que el objetivo principal de este trabajo es analizar el valor Simplicial como alternativa a Shapley, pues las limitaciones de esta propuesta de índice son menos restrictivas.

Por lo que este propósito se puede subdividir en los siguientes hitos:

- ♦ Cálculo de las tablas bases para el juego del aeropuerto.
- ♦ Cálculo de la distribución de costes según el valor de Shapley.
- ♦ Cálculo de la distribución de costes según el valor Simplicial.
- ♦ Análisis de los resultados calculados según ambos índices representativos.

1.3. Estructura del documento

Mientras que en este primer capítulo se contextualiza y justifica el propósito de este trabajo, en los sucesivos se exponen tanto el desempeño realizado como los medios y datos necesarios para su desarrollo.

En el segundo capítulo, por un lado, se introduce brevemente el estado de la técnica en el que se incluye el juego del aeropuerto, y por otro, se comenta la infraestructura sobre la que se ha realizado el análisis, el aeropuerto Adolfo Suárez Madrid-Barajas, así como el lenguaje de programación utilizado para la simulación, Python.

Seguidamente, en el tercer capítulo se presentan los principios teóricos de ambos valores: Shapley y Simplicial, sobre los que se fundamentan las simulaciones recogidas en el quinto capítulo. Estas a su vez, se basan en los valores de partida explicitados en el cuarto capítulo.

A lo largo del sexto capítulo se detallan aspectos relevantes del programa desarrollado, como son las librerías y funciones empleadas.

Finalmente, en el último capítulo se enuncian las conclusiones derivadas de este trabajo, además de futuras líneas en las que se podría extender el mismo.

Adicionalmente, en los anexos de este documento, se incluyen los códigos desarrollados para la ejecución de las simulaciones, la planificación de este trabajo y un presupuesto orientativo del mismo.

2 ESTADO DEL ARTE

A continuación se introduce brevemente el contexto teórico del juego del aeropuerto, pues este pertenece a un conjunto de ejemplos abordados desde una rama de aplicación matemática denominada Teoría de Juegos. Se destaca que la mayoría de los conceptos expuestos provienen del libro “Introduction to the Theory of Cooperative Games” [4].

Adicionalmente, en el segundo apartado, se aborda una resumida descripción del aeropuerto Adolfo Suárez Madrid-Barajas basada en la información pública de su Plan Director. Este aeropuerto se va a emplear como infraestructura modelo para la simulación de la distribución de costes.

Finalmente, en el último apartado se comentan las ventajas y desventajas del lenguaje de programación en el que se ha desarrollado este trabajo, Python.

2.1. Teoría de juegos cooperativos

La Teoría de Juegos es una rama de aplicación matemática centrada en el estudio de escenarios de toma de decisiones en los que participan varios agentes mediante la formulación de modelos teóricos, donde el comportamiento de cada jugador condiciona tanto sus propios intereses como los resultados de los demás jugadores [5].

Estos problemas, al margen del correspondiente factor de azar o aleatoriedad, presentan patrones y pautas que se repiten de manera frecuente en el mundo real. De hecho, aunque en su origen fuese una herramienta desarrollada para comprender materias económicas [6] (a lo largo de las décadas de los treinta y cuarenta), su impacto ha trascendido más allá de este campo, aplicándose también en el análisis de fenómenos sociales o biológicos así como en ámbitos en boga, como la inteligencia artificial o la cibernética.

No obstante, dependiendo del marco de juego, se distinguen dos modelos que requieren de perspectivas de análisis distintas: los juegos cooperativos y los no cooperativos.

Mientras que los juegos no cooperativos (como el famoso dilema del prisionero formalizado por Tucker [7]) se centran en el estudio de las diversas estrategias que puede emplear cada jugador sin interés por alcanzar acuerdos con el resto de jugadores bajo la premisa de perseguir el beneficio personal, la clave de los juegos cooperativos reside en los acuerdos pactados por sus participantes, los cuales pueden estar especificados o no en las reglas del juego.

En relación con los juegos cooperativos, existen dos formas de relación, en estrategia o en coalición. Dividiéndose estos últimos, a su vez, en dos categorías: juegos con utilidades transferibles y juegos con utilidades no transferibles. En particular, los aspectos abordados en este trabajo corresponden a juegos de coalición con utilidades transferibles.

Dichos juegos se caracterizan por una función $v(S)$ que asocia el valor o beneficio del acuerdo al subconjunto no vacío que representa la coalición S dentro del conjunto de jugadores N . De manera que se entiende por coalición al conjunto de jugadores que cooperan con el propósito de obtener un beneficio común. Este, al concluir el juego, tendrá que ser repartido entre los miembros que forman dicha agrupación.

Las premisas de estos juegos de coalición contribuyen a la formulación de problemas ampliamente extendidos cuyo objetivo varía en función de la naturaleza de los mismos. Si el concepto de beneficio conlleva la utilidad positiva de ganancias, se pretende la maximización del mismo, como en el caso de la distribución de dividendos entre los accionistas de una empresa o miembros de una cooperativa. Mientras que si se formula en términos de pérdidas, la finalidad será minimizar dicho beneficio como en la repartición de costes de una infraestructura común o de un servicio logístico del que participen una cadena de usuarios.

En definitiva, es posible definir un juego de coalición con utilidades transferibles (juego UT) mediante un par (N, v) , donde $N = \{1, 2, \dots, n\}$ es un conjunto finito de jugadores y v la función característica que asocia a cada subconjunto S perteneciente a N con un número real $v(S)$, valor de la coalición. Se destaca que si la coalición está formada por un conjunto vacío se le asocia un valor nulo, $v(\emptyset) = 0$.

Por lo que si $G = (N, v)$ es un juego de coalición, entonces N , es el conjunto de jugadores del juego, y v , la función de coalición que permite asociar en el caso de que se forme una subcoalición S dentro de N en G un número real $v(S)$, denominado el beneficio de la coalición, a los miembros que pertenecen a S . Donde todas las posibles distribuciones factibles de $v(S)$ se pueden expresar como distribuciones de pagos de utilidad, siempre y cuando, como máximo se asocie dicho valor. De este modo, la coalición conformada puede conseguir todo conjunto de vectores resultados $x \in \mathbb{R}^S$ que sean factibles y que satisfagan:

$$\sum_{i \in S} x^i \leq v(S)$$

Seguidamente, se formulan ciertas propiedades básicas relevantes de los juegos UT como son la superaditividad y la convexidad, además de ciertas particularizaciones de los mismos: juegos de suma constante y juegos monotónicos [4].

Superaditividad

Se considera que un juego (N, v) es superaditivo si el beneficio que se alcanza al unirse dos coaliciones pertenecientes a N es como mínimo igual a la suma de los que obtendrían cada una si actuaran por separado.

$$(S, T \subseteq N \text{ y } S \cap T = \emptyset) \Rightarrow v(S \cup T) \geq v(S) + v(T)$$

Sin embargo, esta propiedad no siempre se cumple, bien porque la unión de ambas coaliciones se traduce en una disminución del beneficio común debido a factores externos (por ejemplo el incremento de los impuestos a medida que el tamaño de una sociedad aumenta), o bien porque la formación de grandes coaliciones sea ineficiente, pues es más complejo alcanzar acuerdos sobre las distribuciones de pagos.

Como consecuencia, se modera esta propiedad, de modo que se postula que un juego es débilmente superaditivo si el beneficio de la unión de cualquier jugador y de una coalición (a la que el jugador no pertenece) es como mínimo igual a la suma del que alcanzarían si actuaran de manera independiente.

$$v(S \cup \{i\}) \geq v(S) + v(\{i\}) \text{ para toda } S \subseteq N \text{ donde } i \notin S$$

Convexidad

Un juego (N, v) es convexo si el beneficio de la unión de dos coaliciones distintas formadas entre los N jugadores es mayor como consecuencia de su intersección en comparación con el obtenido por cada una de ellas si actuaran por separado.

$$v(S) + v(T) \leq v(S \cup T) + v(S \cap T) \text{ para toda } S, T \subseteq N$$

Esto implica que un juego convexo es superaditivo. Además, en línea con la atenuación de la propiedad anterior, se define un juego convexo, si y solo si, la inclusión de un jugador en una coalición se traduce en una contribución marginal monotónica no decreciente.

$$v(S \cup \{i\}) - v(S) \leq v(T \cup \{i\}) - v(T) \text{ para toda } S \subseteq T \subseteq N \setminus \{i\}$$

Juego de suma constante

Se postula que un juego (N, v) es de suma constante si el beneficio de la coalición es el que le corresponde por haberse formado la misma dentro del conjunto N de jugadores. Esto implica que la suma del beneficio de la coalición y del resto de jugadores sea el que se podría obtener del juego.

$$v(S) + v(N \setminus S) = v(N) \text{ para toda } S \subseteq N$$

Consecuentemente, si un juego es aditivo, la utilidad de una coalición es la suma de los beneficios de todos los jugadores que de ella forman parte. Lo cual es trivial desde el punto de vista de la Teoría de Juegos, ya que todo jugador reclama un provecho que da lugar a una distribución única de la ganancia distribuable por el juego, $v(N)$.

$$v(S) = \sum_{i \in S} v(\{i\}) \text{ para toda } S \subseteq N$$

Esta última expresión, mediante una equivalencia de términos con el conjunto de vectores resultados puede ser reformulada. De manera que si se considera N una coalición y \mathbb{R}^N el conjunto de todas las funciones de N a \mathbb{R}^N , cuando $x \in \mathbb{R}^N$ y $S \subseteq N$, entonces:

$$x(S) = \sum_{i \in S} x^i$$

Por consiguiente, si se considera x como una función de coalición, el juego (N, v) también puede definirse como:

$$x(S) = \sum_{i \in S} x^i \text{ para toda } S \subseteq N$$

Juego monotónico

Un juego (N, v) se denomina monotónico si el beneficio de una subcoalición de una coalición es como máximo igual al de la coalición a la que pertenece. Es decir, una subcoalición formada dentro de una

coalición obtendrá un beneficio igual o menor al que alcance la coalición de la que forma parte.

$$S \subseteq T \subseteq N \Rightarrow v(S) \leq v(T)$$

2.1.1. Juego de distribución de costes

Al margen de las particularizaciones de los juegos de coalición ya presentadas, existen diversas familias de juegos dependiendo de la aplicación última del modelo, y por ende de la interpretación de la función de utilidad v : juegos de mercado (útiles para modelar los acuerdos entre comerciantes), juegos de distribución de costes (se emplean para la repartición de costes de un servicio o infraestructura) y juegos simples (sirven para formular a un grupo de personas que toma decisiones, por ejemplo un comité).

Particularmente, este trabajo se centra en la familia de distribución de costes. En ella, se define como juego de costes a la tupla (N, c) formada por la coalición N de potenciales usuarios que hacen uso del servicio o instalación y la función de coalición c , la cual representa la función de costes del problema. De modo que $c(S)$ representa, como ya se ha introducido, el coste mínimo para cumplir el servicio prestado a los miembros de $S \in N$ de la manera más eficiente.

Sin embargo, la función de costes no puede interpretarse como una función ordinaria de coalición. Por lo que, mediante la siguiente transformación, al juego de costes (N, c) se le asocia un juego ordinario (N, v) , denominado juego de ahorros. Este representa la cantidad que se ahorra al prestar el servicio a un conjunto de usuarios en forma de coalición en lugar de independientemente.

$$v(S) = \sum_{i \in S} c(\{i\}) - c(S) \text{ para toda } S \subseteq N$$

En relación con la transformación de ambos juegos, se destaca que las propiedades básicas (ya definidas), superaditividad y convexidad, para el juego ordinario de ahorros (N, v) , se mantienen siempre y cuando el juego de costes cumpla sus homólogas: subaditividad y concavidad. Es decir, el juego de costes se considera subaditivo y cóncavo si el coste en el que se incurre al cubrir el servicio para dos coaliciones de manera conjunta es como máximo el que habría que asumir al atender ambas por separado.

Subaditividad

$$(S, T \subseteq N \text{ y } S \cap T = \emptyset) \Rightarrow c(S \cup T) \leq c(S) + c(T)$$

Concavidad

$$c(S) + c(T) \geq c(S \cup T) + c(S \cap T) \text{ para toda } S, T \subseteq N$$

2.2. Aeropuerto de Barajas

Tal y como en el apartado anterior se recoge, una utilidad habitual de los juegos de coalición es la repartición de costes de una infraestructura común para varios agentes. Bajo este contexto se enmarca el problema del aeropuerto, el cual se va a emplear como eje transversal de los planteamientos recogidos en este documento.



Figura 2–1. Conjunto de aeropuertos en España [8].

En particular, se ha escogido para los diferentes desarrollos el aeropuerto Adolfo Suárez Madrid-Barajas, ya que es un enclave estratégico dentro de la geografía española (localizado a unos 13 km al Noroeste de la capital española) tanto como punto de entrada de turistas internacionales como de destino consolidado entre los ciudadanos regionales. De modo que el tráfico aéreo de esta infraestructura, explotada y mantenida por el Ente Público de Aeropuertos Españoles y Navegación Aérea (Aena), requiere de servicios operacionales que se extiendan durante 24 h y de instalaciones de gran envergadura.

Según la Organización de Aviación Civil Internacional (OACI), este aeropuerto se clasifica como un aeródromo de clase 4E. De manera que en relación a su código, la longitud de su campo de referencia (longitud mínima necesaria para despegar bajo las condiciones prescritas por la autoridad certificadora) es mayor a los 1.800 m, y además, las dimensiones de la envergadura de las aeronaves que acoge se encuentran entre los 52 y 65 m con distancias externas del tren de aterrizaje principal de las aeronaves de 9 a 14 m.

Por otro lado, a partir de la información pública disponible, la estructura de este sistema aeroportuario puede clasificarse en los siguientes subsistemas diferenciados [9]:

- ◆ Subsistema Movimiento de Aeronaves
 - TMA/TCR
 - Campos de vuelos: pistas de vuelo, calles de rodaje y franjas de seguridad
 - Plataforma: zona de espera, seguridad y estacionamiento de aeronaves
 - Viales y aparcamiento de vehículos de servicio
 - Puestos de carga
 - Instalaciones para equipos y vehículos de servicio
 - Zonas de Acceso restringido de Terminales
- ◆ Subsistema de Actividades Aeroportuarias
 - Zona de Pasajeros
 - Zona de Carga

- Zona Industrial
- Zona de Servicios
- Zona de Abastecimiento Energético
- Zonas Libres
- ◆ Reservas Aeroportuarias

Sin embargo, puesto que este trabajo se va centrar únicamente en la distribución de costes relacionados con las pistas aeroportuarias, seguidamente, solo se van a describir los subsistemas directamente relacionados con los aspectos de movimiento de aeronaves.

2.2.1. Subsistema de Movimiento de Aeronaves

Campo de vuelo

Este aeropuerto dispone de tres pistas principales, 15-33, 18L-36R y 18R-36L, cuyas dimensiones características (TORA= recorrido de despegue disponible, TODA= distancia de despegue disponible, ASDA= distancia de aceleración-parada disponible, LDA= distancia de aterrizaje disponible) se recogen en la siguiente tabla.

Tabla 2-1. Dimensiones características de las pistas del aeropuerto Madrid-Barajas [10].

Pista	Dimensiones (m)	TORA (m)	TODA (m)	ASDA (m)	LDA (m)
15	4.100 x 45	4.100	4.200	4.150	4.100
33	4.100 x 45	4.100	4.200	4.150	3.050
18L	3.700 x 45	3.700	3.800	3.750	3.003
36R	3.700 x 45	3.700	3.800	3.750	3.700
18R	4.350 x 60	No utilizable	No utilizable	No utilizable	3.365
36L	4.350 x 60	4.350	4.610	4.350	No utilizable

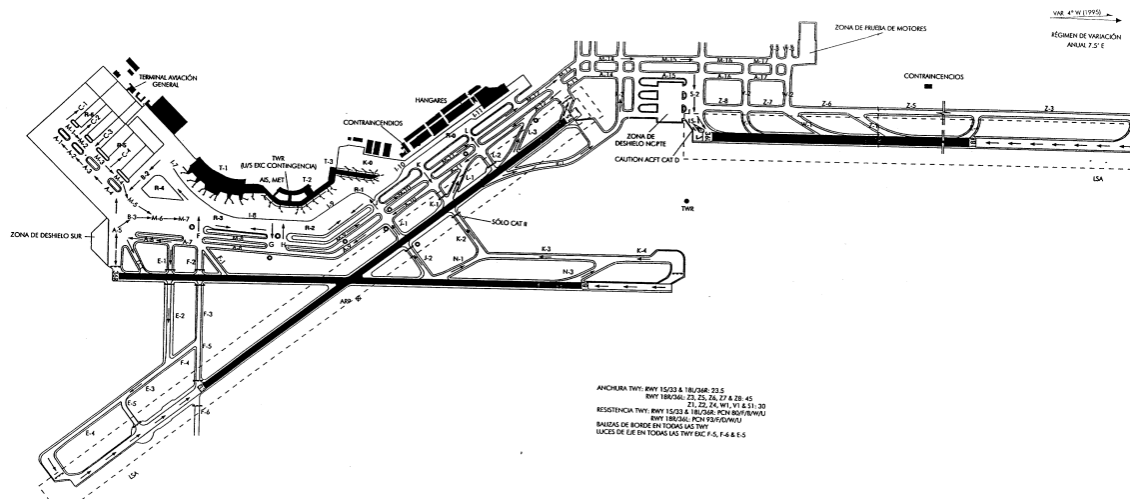


Figura 2–2. Plano de las configuraciones de pistas del aeropuerto AS Madrid-Barajas [9].

No obstante, aunque existen 3 pistas (Figura 2–2), estas se utilizan en dos configuraciones: Norte y Sur. En la Norte (la más habitual), en condiciones normales, se utiliza la pista 36L para el despegue mientras que para los aterrizajes la 33, quedando fuera de uso la 15. Por el contrario, en la configuración Sur, los despegues se realizan desde la pista 15 y se utiliza la pista 18R para los aterrizajes, de modo que la pista 33 queda desautorizada para este uso.

Plataforma de aeronaves

Otras zonas destacables son los 180 puestos de estacionamiento de la plataforma que ocupan aproximadamente unos 1.585.000 m² de superficie. Estos, en función de su accesibilidad se dividen en 3 grupos: aquellos que son asistidos por pasarela (35), los anexos al edificio de la terminal pero sin pasarela (4) y las posiciones remotas (141). A su vez, según el Servicio de Información Aeronáutica de España (AIP España), estos puestos se clasifican por rampas en función del tamaño máximo de aeronave admisible y otros factores (modo de acceso o incompatibilidades con posiciones próximas). En la siguiente tabla se resumen dichas posiciones por los tamaños máximos de aeronaves admitidos.

Tabla 2-2. Cantidad y modelos máximos de aeronaves admisibles.

Rampa	Cantidad Posiciones	Modelo Aeronave
R0	7	
	7	A321
R1	24	
	1	B737
	14	B757
	4	B767
	5	ATR
R2	14	
	1	A300
	2	ATR
	2	B767
	9	B757
R3	19	
	1	B757
	1	B737
	2	A321
	1	MD11
	4	B747
	5	B757
	5	ATR
R4	19	
	1	A320
	1	MD87
	2	A321
	4	B767
	5	B757
	6	B744
R5	30	
	2	B767
	4	A321
	9	B757
	15	B744

R6	46	
	1	B747
	2	B744
	2	DC8
	3	SW4
	4	B757
	4	B767
	7	BAe146
	8	DC9
	15	A321
NAT	21	
	2	B747
	2	DC8
	4	FK50
	5	B727
	8	SW4

Conjuntamente, en la Tabla 2-3 se han recopilado las plazas totales en función del tipo de modelo máximo admisible. En ella se han incluido características relevantes de cada modelo en su fase más crítica, el despegue, ya que tanto la distancia como la masa máxima de aterrizaje son de menor valor que los de la etapa inicial de la travesía. Además, en función de la masa máxima de despegue se han agrupado los modelos según la categoría del criterio denominado “*Wake Turbulence Category*” (WTC).

Según la masa máxima de la aeronave en el despegue se pueden clasificar las categorías del siguiente modo:

- J (Super) es una clase específica para el A380 el cual alcanza una masa de 560 ton.
- H (Heavy) para aeronaves cuya masa en el despegue sea igual o mayor a 136 ton.
- M (Medium) incluye el grupo mayor que 7 ton y menor que 136 ton.
- L (Light) a este pertenecen las aeronaves con una masa máxima igual o menor a 7 ton.

Cabe señalar que los datos proporcionados son aproximados pues no se disponen de fichas técnicas oficiales de sus fabricantes.

Tabla 2-3. Características de los modelos de aeronaves [11]–[13].

Modelo Aeronave	Cantidad Posiciones	Masa máxima despegue (MTOW)	Carrera de despegue (TOR)	Capacidad (Asientos)	Tipo Aeronave (WTC)
A300	1	165 ton	2.300 m	270	H
A320	1	79 ton	2.090 m	165	M
A321	30	97 ton	2.560 m	190	M
ATR	12	23 ton	1.300 m	72	M
B727	5	95 ton	3.000 m	295	M
B737	2	55 ton	2.000 m	150	M
B744	23	412 ton	3.300 m	524	H
B747	7	330 ton	3.250 m	420	H
B757	47	124 ton	2.377 m	295	M
B767	16	160 ton	2.420 m	340	H
BAe146	7	43 ton	1.400 m	100	M

DC8	4	161 ton	3.043 m	189	H
DC9	8	50 ton	2.100 m	125	M
FK50	4	22 ton	1.350 m	50	M
MD11	1	285 ton	3.100 m	390	H
MD87	1	68 ton	1.850 m	120	M
SW4	11	8 ton	1.200 m	20	M

2.3. Herramienta utilizada

En la actualidad, son numerosos los lenguajes y herramientas existentes para programar, no obstante, cada uno de ellos suele estar enfocado a campos concretos y/o a niveles de desarrollo según la aplicación.

Concretamente, el empleado en este trabajo ha sido Python, pues el uso de este lenguaje de alto nivel está muy extendido ya que permite la integración de múltiples librerías que incluyen funciones previamente desarrolladas para mayor agilidad de los usuarios.

Tabla 2-4. Ranking de lenguajes de programación según el índice TIOBE [14].

Lenguaje de programación	Posicionamiento Oct 2022	Calificación
Python	1	17.08%
C	2	15.21%
Java	3	12.84%
C++	4	9.92%
C#	5	4.42%

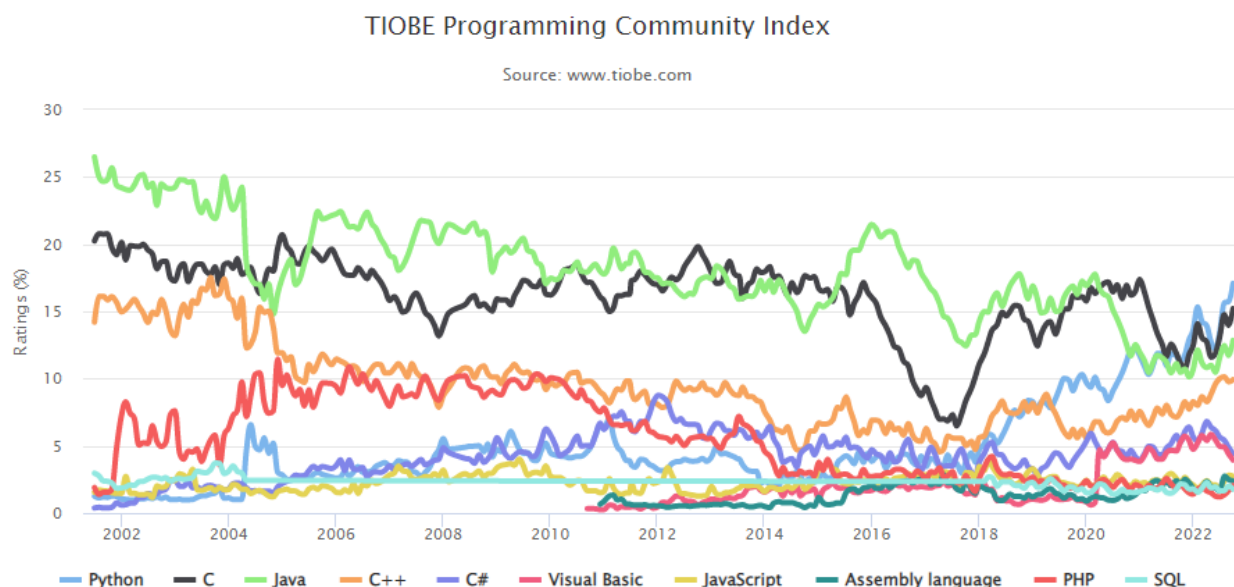


Figura 2–3. Evolución de la tendencia del índice TIOBE para diversos lenguajes de programación [14].

De hecho, Python es el primer lenguaje por segundo año consecutivo según el índice TIOBE [14], el cual es un ranking (actualizado todos los meses) que valora la popularidad de los diferentes lenguajes de programación en función del uso y la evolución de los mismos.

La definición de este índice se basa en el número de ingenieros especializados en todo el mundo, los cursos y los proveedores de terceros. Cabe resaltar que este índice no busca establecer cuál es el mejor lenguaje ni en cuál se han escrito más códigos, sino que está enfocado para guiar a los usuarios en las tendencias del mercado ante decisiones sobre la actualización de conocimientos o la estrategia a seguir ante un nuevo desarrollo software.

La popularidad de este lenguaje no solo se debe al crecimiento de uso por los profesionales dentro de las empresas, sino que debido a su simplicidad, versatilidad y potencial, son mucho los jóvenes (incluso de campos diversos a los tecnológicos) que se están interesando en él, lo cual refuerza la elección del mismo para el desarrollo de las simulaciones recogidas.

“Python se trata de un lenguaje de programación multiparadigma (principalmente orientado a objetos), interpretado y con un tipado dinámico y fuerte, cuya filosofía hace hincapié en una sintaxis sencilla, clara y legible. Es por ello por lo que consta de una amplia variedad de estructuras de alto nivel, que hacen de él lenguaje sumamente expresivo. Considerado un magnífico lenguaje de propósito general y prototipado, permite al usuario tanto acceder al código fuente (código abierto) como hacer uso de él con independencia del sistema operativo utilizado (multiplataforma)” [15].

Este lenguaje es multiparadigma debido a que permite la combinación de diferentes estilos de desarrollo de programas, aunque su paradigma principal es la orientación a objetos. Este tipo de programación, se diferencia de la tradicional estructurada en que esta no sigue un diseño modular basado en una lista de instrucciones secuenciales, sino que en lugar de dar relevancia a la función, pone el énfasis en los datos. De manera que trata de adaptar el lenguaje al problema, en lugar de tratar de encajar el problema en la estructura establecida por el algoritmo procedimental tradicional.

Además, debido a que su sintaxis es sencilla, clara y legible, este se asemeja más al lenguaje natural (lo cual también lo capacita de expresividad) que al máquina interpretado por un computador (bajo nivel). Sin embargo, esto ralentiza su ejecución en comparación con los de bajo nivel.

Se caracteriza por ser de tipado dinámico, por lo que las variables no tienen que ser declaradas con anterioridad a su ejecución, sino que en el momento en el que se le asigna un valor, estas toman el tipo de este. No obstante, una vez asignado un tipo a una variable, este no puede ser violado hasta hacer una conversión de tipos, ya que es un lenguaje fuerte. Dependiendo del tipo de la variable, esta tendrá unas características y funciones, como una cadena de caracteres o un número entero.

Este lenguaje tiene multitud de aplicaciones, pues permite resolver todo tipo de problemas o al menos prototiparlos para después desarrollarlos en otro lenguaje de programación que resulte más idóneo para esa cuestión concreta.

Por último, destacar que es un lenguaje que mediante un intérprete puede ser compatible con multitud de plataformas, lo que junto a su característica de código abierto (libremente utilizable y distribuible), permite al usuario un amplio abanico de posibilidades de explotación.

De entre sus múltiples ventajas, resaltar que es un lenguaje que se aprende fácilmente en comparación a otros. Además, debido a su estructura y a ser un código abierto, este favorece las sinergias entre programadores, incrementando la productividad y la mejora de los códigos.

Respecto a sus desventajas, se señala que debido a su gestión de datos, no es la opción más idónea si se pretenden optimizar recursos de memoria. Adicionalmente, como ya se ha indicado, por ser de alto

nivel, es más lento que otros lenguajes.

Por otro lado, cabe destacar que el programa de este trabajo se ha desarrollado en la plataforma Google Colaboratory, la cual sirve para poder escribir y ejecutar desde el navegador online Python y otros lenguajes.

La ventaja principal de este servicio en la nube es que no necesita de una configuración previa del entorno a diferencia de otros (cuya complejidad de configuración impide su posterior empleo) sino que muchos de los paquetes y librerías ya están preinstalados y listos para su uso. Además, es una herramienta con GPU gratis, extendiendo los límites de los dispositivos de los que disponen los programadores. Finalmente, se destaca que permite tanto almacenar archivos directamente en Google Drive como importarlos desde Github (el cual es un repositorio de código muy extendido) o desde el propio ordenador. En definitiva, es un servicio sencillo que fomenta la interacción y cooperación de la comunidad de programadores, pues al estar en la nube permite el acceso y la edición por parte de los usuarios con permisos para ello.

3 FUNDAMENTO TEÓRICO

En el presente capítulo se introduce el planteamiento teórico de los dos valores relevantes en los que se basan las distribuciones de costes simuladas en este trabajo.

Principalmente cabe destacar que mientras la repartición tradicional sigue una división lineal de los costes totales entre los usuarios, el nuevo modelo que aquí se incluye, basado en el valor Simplicial, responde a una distribución no lineal más compleja que tiene en consideración todas las posibles coaliciones que se pueden formar entre los participantes.

3.1. Solución de los juegos de coalición

Siendo U un conjunto de jugadores y (N, v) un juego, el conjunto de todos los vectores resultantes factibles, solución de dicho juego, es $X^*(N, v)$.

$$X^*(N, v) = \{x \in \mathbb{R}^N | x(N) \leq v(N)\}$$

En el caso de que no fuese un solo juego sino un conjunto Γ de ellos, una solución en Γ , es una función σ que asocia con cada juego $(N, v) \in \Gamma$ un subconjunto $\sigma(N, v)$ de $X^*(N, v)$. Donde cada solución de $\sigma(N, v)$ es considerada una posible distribución de pago final de (N, v) .

Por otra parte, cada solución puede determinarse en función de un sistema de restricciones “razonables” en su correspondiente $X^*(-, -)$, por ejemplo se puede garantizar la “estabilidad” de alguna solución de $\sigma(N, v)$ o se puede caracterizar σ por un conjunto de axiomas.

De hecho, existen soluciones ya formuladas ampliamente extendidas, como el “Core” (núcleo) de un juego o el valor de Shapley, que derivan de la imposición de ciertas restricciones.

3.2. Valor de Shapley

El valor de Shapley surge ante la necesidad que tiene cada jugador de evaluar la “perspectiva” que puede emerger como resultado de una jugada. En relación con ello, este valor, proporciona una evaluación a priori de todo juego de coalición.

Dicha evaluación del juego de coalición realizada por cada jugador, debe ser una solución de valor único,

es decir, la solución σ de un conjunto de juegos $\Gamma \subseteq \Gamma_U$ (conjunto de juegos con jugadores pertenecientes al conjunto U) es de valor único si $|\sigma(N, v)| = 1$ para todo $(N, v) \in \Gamma$.

Además del criterio de unicidad del valor, Shapley asume otros tres axiomas, que desde su punto de vista debería cumplir la distribución óptima de pagos. Por lo que partiendo de ellos, demostró que solo una solución los cumplía todos. Esa asignación, determina de manera única el valor que se reparte a cada jugador de cada posible juego, y se la denomina valor de Shapley.

Considerando N una coalición y \mathcal{V} el conjunto de todas las funciones de coalición en 2^N , de modo que \mathcal{V} puede ser identificado como el conjunto de todos los juegos cuyos conjunto de jugadores es N (donde se asume que $N = \{1, \dots, n\}$). Existe una única solución σ en el conjunto de todos los juegos, cuyo conjunto de jugadores es N , que satisface las propiedades de: No Vacío, Igualdad de Trato, Jugador Nulo y Aditividad.

Para $T \subseteq N, T \neq \emptyset$, se define la unanimidad del juego en T , u_T , como

$$u_T(S) = \begin{cases} 1 & \text{si } S \supseteq T \\ 0 & \text{los demás} \end{cases}$$

De forma que el conjunto $\{u_T \mid \emptyset \neq T \subseteq N\}$ es una base lineal de \mathcal{V} , es decir, que los juegos de unanimidad son linealmente independientes.

Considerando que $\mathcal{K} \subseteq \mathcal{V}$ es un cono convexo que contiene todos los juegos de unanimidad, el único valor en \mathcal{K} es el valor de Shapley.

De modo que si \mathcal{V}^S es el conjunto de todos los juegos superaditivos en \mathcal{V} , entonces ϕ es el único valor en \mathcal{V}^S . Al igual que si \mathcal{V}^m es el conjunto de todos los juegos monotónicos en \mathcal{V} , donde ϕ es el único valor en \mathcal{V}^m . De hecho, el valor de Shapley es el único valor en el conjunto de todos los juegos convexos en \mathcal{V} .

El valor de Shapley ϕ en \mathcal{V} para todo juego $v \in \mathcal{V}$ y para todo jugador $i \in N$ es dado por:

$$\phi^i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} (v(S \cup \{i\}) - v(S))$$

La interpretación dada por Shapley (1953): *“Los jugadores en N acuerdan jugar el juego v en una gran coalición. De manera que esta comienza con un solo miembro, a la que se van añadiendo cada vez otro jugador hasta que todos están admitidos. El orden en el que los jugadores se unen es determinado por la suerte, donde todos los acuerdos son igualmente probables. Cada jugador en su admisión, exige y se le promete la cantidad que su adhesión contribuye al valor de la coalición (como determina la función v). Por lo que, la gran coalición, deberá jugar eficientemente para obtener un beneficio $v(N)$ exactamente suficiente para cumplir todas las promesas”.*

3.3. Valor Simplicial

Un símplice es el cierre convexo de k puntos afínmente independientes en un espacio m dimensional. Puede verse, a nivel geométrico, como la generalización de un triángulo de k vértices. La dimensión del símplice es $k-1$.

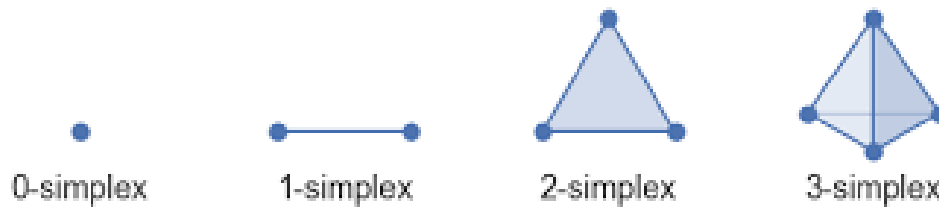


Figura 3–1. Representación del símplex.

Específicamente un símplex k -dimensional generado por $k+1$ puntos puede definirse como:

$$\Delta_{\{k\}} = \{a_0 u_0 + \dots + a_k u_k \mid \sum a_s = 1, a_s \geq 0.\}$$

Es claro que se puede asociar a una coalición un símplex en la manera siguiente:

Se escoge un vector de orden n : (x_1, x_2, \dots, x_n) , donde los elementos de este vector pueden ser ceros o unos. Si son unos los incluimos en la coalición asociada y si son cero no. Por ejemplo el vector $(1,0,0,1,1)$ se corresponde con la coalición (145) . Tomando entonces los jugadores de las coaliciones como los vértices de un símplex se tiene la identificación.

Ahora se supone que se ordenan de menor a mayor los valores que toma el juego. Suponiendo que son $Im(v) = \{a_1, \dots, a_k\}$, se van creando los siguientes subsímplexes que formarán lo que se llama una filtración. Así $k_{a_i} = \{S \subseteq N : v(S) \leq a_i\}$

Por ejemplo si se supone que:

$$v(1) = 1$$

$$v(2) = 2 = v(12)$$

$$v(3) = v(13) = v(23) = v(123) = 3$$

$$v(4) = v(14) = v(24) = v(34) = v(124) = v(134) = v(234) = v(N) = 4$$

Esta estructura daría la filtración del 3-símplex de la Figura 3–1.

El valor que se va a usar depende fuertemente de las coaliciones que aparecen en cada elemento de la filtración. Observar que en el primer elemento solo aparece 1 a nivel 1. En la segunda 2 y 12 a nivel 2, en la siguiente 3, 13, 23, 123 a nivel 3 y el resto a nivel 4.

El valor entonces es:

$$\varphi_i(N, v) = \sum_{s=1:k} \frac{1}{M_s} \sum_{\{T, i \in T \subset K_s - K_{s-1}\}} \frac{1}{|T|} [v(K_s) - v(K_{s-1})] \quad (1)$$

Donde M_s indica el número de coaliciones nuevas que aparecen en cada elemento nuevo de la filtración.

La primera suma indica que sumamos en cada elemento de la filtración y la segunda que en cada elemento de la filtración sumamos en esas coaliciones el inverso de su cardinal. Además de ir multiplicando por el valor de los saltos.

Observar que las contribuciones marginales se hacen en función de las diferencias de valores entre las filtraciones multiplicadas por la media de los inversos de los cardinales de las nuevas contribuciones.

Que en este caso es:

$$[1, 1/2(1 + 1/2), 1/4(1 + 1/2 + 1/2 + 1/3), 1/8(1 + 3/2 + 3/3 + 1/4)] = [1, \frac{3}{4}, \frac{7}{12}, \frac{15}{32}]$$

Aquí el jugador 1 es el que más costo/beneficio recibe en función del tipo de juego.

4 CÁLCULOS DEL JUEGO INICIAL

En este capítulo se muestran los datos de partida para los cálculos realizados en el sucesivo. Estos representan los costes totales de las pistas de despegue en el aeropuerto AS Madrid-Barajas.

Primeramente, se especifican las divisiones realizadas de las pistas que darán lugar a los posteriores juegos de distribución de costes, y a continuación, se recogen sus valores en correspondencia con la utilidad de las mismas.

4.1. Datos de partida: División de secciones

Como se ha indicado anteriormente, el despegue es la operación más crítica de las llevadas a cabo en las pistas del aeropuerto. Es por ello que se ha considerado conveniente enfocar la distribución de costes en función del parámetro de la carrera de despegue máxima necesaria (TOR).

En base a esta característica, se han agrupado los modelos ya presentados en seis conjuntos según el orden creciente de dicha distancia. Particularmente, las subdivisiones contempladas se corresponden con:

- ◆ Conjunto A: donde se incluyen las aeronaves que precisan de una TOR menor o igual a 1.500 m.
- ◆ Conjunto B: al que pertenecen los modelos que requieren una TOR mayor de 1.500 m pero menor o igual a 2.000 m.
- ◆ Conjunto C: es el que comprende las aeronaves con una TOR mayor de 2.000 m y menor o igual que 2.300 m.
- ◆ Conjunto D: en el que se agrupan los modelos que necesitan una TOR mayor de 2.300 m hasta los 2.500 m.
- ◆ Conjunto E: cuyos modelos asociados precisan de una distancia de despegue mayor de 2.500 m pero menor o igual a 2.800 m.
- ◆ Conjunto F: al que pertenecen todas las aeronaves que requieren de distancias superiores a los 2.800 m.

Debido a la subdivisión de modelos planteada, se destaca que todos los conjuntos podrán despegar desde ambas configuraciones del aeropuerto AS Madrid-Barajas, ya que las distancias TOR que pueden llegar a precisar las aeronaves de estas agrupaciones son menores a las longitudes de despegue de las pistas 36L y 15, asignadas en condiciones normales a dicha operación según la configuración Norte (CN) y Sur (CS) respectivamente, cuyas distancias de despegue disponible (TODA) superan los 3.300 m que

precisaría el modelo de aeronave de mayores dimensiones contemplado.

Tabla 4-1. Agrupación de los modelos de aeronave en función de la TOR necesaria.

Agrupación modelos	Modelo aeronave	TOR [m]	WTC	Cantidad por modelo	Cantidad por categoría
A	SW4	1.200	M	11	34
	ATR	1.300	M	12	
	FK50	1.350	M	4	
	BAe146	1.400	M	7	
B	MD87	1.850	M	1	3
	B737	2.000	M	2	
C	A320	2.090	M	1	10
	DC9	2.100	M	8	
	A300	2.300	H	1	
D	B757	2.377	M	47	63
	B767	2.420	H	16	
E	A321	2.560	M	30	35
	B727	2.800	M	5	
F	DC8	3.043	H	4	35
	MD11	3.100	H	1	
	B747	3.250	H	7	
	B744	3.300	H	23	

4.2. Repartición base de costes

El problema del aeropuerto, como ya se ha introducido, es un ejemplo muy conocido de la Teoría de Juegos utilizado para la descripción de una distribución justa de costes basada en el uso de las instalaciones aeroportuarias y sus servicios.

Aunque en otros contextos sociales como la sanidad o la educación, la repartición de costes (o desde la perspectiva de la contribución a los servicios públicos en España bajo la denominación de impuestos) no se debe plantear bajo el lema de “pagar lo que se consume y/o utiliza” (ya que la distribución de la riqueza según la Constitución Española debe asegurar unos derechos fundamentales a todos los ciudadanos independientemente de su capacidad económica), en el caso de las infraestructuras de transporte, es razonable plantear la repartición de costes en función del uso de las mismas (al igual que el caso de compra de una vivienda), puesto que ciertas partes no necesitarían de la construcción y mantenimiento de instalaciones mayores para su desempeño.

Alineado con este razonamiento, se considera que es justo repartir los costes de las pistas aeroportuarias entre las aerolíneas en función de los requerimientos que precisen sus diversas aeronaves de las instalaciones para su operación.

Tal y como se ha adelantado, esta simulación se va centrar únicamente en los costes de las pistas de despegue. Además, como se refleja en la Tabla 4-1 se han agrupado en categorías los diversos modelos de aeronaves según la TOR máxima que precisan para dicha operación. De modo que las aerolíneas poseedoras de aeronaves de un conjunto deben sufragar solamente los metros de pista que necesiten para su operación, como sugiere la solución óptima propuesta por S. Littlechild y G. Owen.

Con objeto de aproximar las simulaciones desarrolladas a casos reales, se ha estimado el coste de la pista del aeropuerto alrededor de los 4.000 € por metro de largo. Aunque este valor depende tanto del material como del tipo de obra que se necesite según la topografía de la zona donde se localice la infraestructura entre otros factores.

Teniendo en consideración estos datos, seguidamente se recopilan los costes asignados a las secciones en las que se han subdividido las pistas de despegue en función de los metros de las mismas.

Tabla 4-2. Costes totales de las secciones de la pista de despegue de la configuración Norte.

Secciones de pista	Longitud de las secciones [m]	Coste de las secciones [€]
A	1.500	6.000.000
B	500	2.000.000
C	300	1.200.000
D	200	800.000
E	300	1.200.000
F	1.810	7.240.000

Tabla 4-3. Costes totales de las secciones de la pista de despegue de la configuración Sur.

Secciones de pista	Longitud de las secciones [m]	Coste de las secciones [€]
A	1.500	6.000.000
B	500	2.000.000
C	300	1.200.000
D	200	800.000
E	300	1.200.000
F	1.400	5.600.000

De las tablas se resalta que mientras que los costes de las secciones de la A a la E son coincidentes para ambas configuraciones, los de la última sección difieren, pues los metros longitudinales de las mismas también lo hacen.

Aunque en este trabajo no se contemplan aeronaves que requieran de más de 3.300 m, se ha decidido no restringir las secciones F a este límite, ya que en el caso de que las aerolíneas poseyeran modelos con TOR mayores, las TODA de las pistas tendrían que ser iguales o superiores a estas longitudes, por lo que se considera conveniente incluir la longitud total de estas secciones finales para las simulaciones.

Dichos costes totales se distribuyen de manera equitativa entre las agrupaciones de aeronaves que precisan el uso de una sección determinada para el despegue (Tabla 4-4 y Tabla 4-5). Es decir, mientras un modelo de aeronave A solo requiere de la sección con esa misma denominación para dicha operación, una aeronave agrupada en D requiere de las secciones de la A a la D para poder despegar, por lo cual tendrá que aportar una retribución acorde con dicho uso como propusieron Littlechild y Owen.

Estas divisiones equitativas de costes, recogidas en las tablas sucesivas, sientan las base para los cálculos que se realizarán posteriormente según dos perspectivas de valores diversas, cuyo propósito en común es establecer el pago que debería abonar cada compañía aérea por el uso de las pistas de despegue en función del modelo de aeronave con el que opere.

Con objeto de expresar la distribución de costes de forma representativa, se expresan las cantidades millonarias en valores en tanto por uno (Tabla 4-6 y Tabla 4-7). Estos últimos son los que se emplean en los desarrollos posteriores para los cálculos de los índices.

Tabla 4-4. Distribución equitativa de los costes de las secciones CN por categorías de aeronaves.

Costes pista despegue configuración Norte por categoría						
	A	B	C	D	E	F
A	1000000.0	0.0	0.0	0.0	0.0	0.0
B	1000000.0	400000.0	0.0	0.0	0.0	0.0
C	1000000.0	400000.0	300000.0	0.0	0.0	0.0
D	1000000.0	400000.0	300000.0	266666.6666666667	0.0	0.0
E	1000000.0	400000.0	300000.0	266666.6666666667	600000.0	0.0
F	1000000.0	400000.0	300000.0	266666.6666666667	600000.0	7240000.0

Tabla 4-5. Distribución equitativa de los costes de las secciones CS por categorías de aeronaves.

Costes pista despegue configuración Sur por categoría						
	A	B	C	D	E	F
A	1000000.0	0.0	0.0	0.0	0.0	0.0
B	1000000.0	400000.0	0.0	0.0	0.0	0.0
C	1000000.0	400000.0	300000.0	0.0	0.0	0.0
D	1000000.0	400000.0	300000.0	266666.6666666667	0.0	0.0
E	1000000.0	400000.0	300000.0	266666.6666666667	600000.0	0.0
F	1000000.0	400000.0	300000.0	266666.6666666667	600000.0	5600000.0

Tabla 4-6. Fracción en tanto por uno de los costes de las secciones CN por categorías de aeronaves.

Fracción del total de sección por categoría pista despegue configuración Norte por categoría						
	A	B	C	D	E	F
A	0.16666666666666666	0.0	0.0	0.0	0.0	0.0
B	0.16666666666666666	0.2	0.0	0.0	0.0	0.0
C	0.16666666666666666	0.2	0.25	0.0	0.0	0.0
D	0.16666666666666666	0.2	0.25	0.33333333333333337	0.0	0.0
E	0.16666666666666666	0.2	0.25	0.33333333333333337	0.5	0.0
F	0.16666666666666666	0.2	0.25	0.33333333333333337	0.5	1.0

Tabla 4-7. Fracción en tanto por uno de los costes de las secciones CS por categorías de aeronaves.

Fracción del total de sección por categoría pista despegue configuración Sur por categoría						
	A	B	C	D	E	F
A	0.16666666666666666	0.0	0.0	0.0	0.0	0.0
B	0.16666666666666666	0.2	0.0	0.0	0.0	0.0
C	0.16666666666666666	0.2	0.25	0.0	0.0	0.0
D	0.16666666666666666	0.2	0.25	0.3333333333333337	0.0	0.0
E	0.16666666666666666	0.2	0.25	0.3333333333333337	0.5	0.0
F	0.16666666666666666	0.2	0.25	0.3333333333333337	0.5	1.0

5 APLICACIÓN Y COMPARACIÓN

A lo largo de este capítulo se muestran los resultados obtenidos a partir de la simulación del juego del aeropuerto según los dos métodos de cálculo contemplados: Shapley y Simplicial, así como su comparación.

Cabe destacar que el código programado en Python para esta aplicación se adjunta en los anexos, de modo que en estos apartados únicamente se muestran los valores más relevantes.

5.1. Distribución equivalente al valor de Shapley

El resultado de la distribución tradicional de costes entre las diferentes categorías de aviones, equivalente al valor de Shapley, se obtiene a partir de la suma independiente del valor que le corresponde por requerir de una determinada sección de la pista para el despegue (tal y como demostraron Littlechild y Owen).

Por lo que el coste que deberá asumir cada compañía aérea dependiendo del tipo de aeronave asciende al valor lineal acumulado por filas de la Tabla 4-4 y Tabla 4-5 para las configuraciones Norte y Sur respectivamente.

Tabla 5-1. Costes totales para cada categoría de aeronaves en la configuración Norte.

Categorías de aeronaves	Costes totales [€]
A	1.000.000
B	1.400.000
C	1.700.000
D	1.966.666
E	2.566.666
F	9.806.666

Tabla 5-2. Costes totales para cada categoría de aeronaves en la configuración Sur.

Categorías de aeronaves	Costes totales [€]
-------------------------	--------------------

A	1.000.000
B	1.400.000
C	1.700.000
D	1.966.666
E	2.566.666
F	8.166.666

De dichas tablas se puede observar como a medida que se precisan de mayores metros para el despegue los costes de cada categoría aumentan siguiendo la distribución lineal ya anunciada. Redondeando los valores en tanto por uno para sendas configuraciones de las Tablas 4-6 y 4-7 al primer decimal y, siguiendo el mismo procedimiento de adición acumulada en las filas, se obtienen los valores equivalentes a Shapley que se recogen en las siguientes tablas.

Tabla 5-3. Valor representativo tradicional equivalente a Shapley para cada categoría de aeronaves en la CN.

Categorías de aeronaves	Valor
A	0,2
B	0,4
C	0,7
D	1
E	1,5
F	2,5

Tabla 5-4. Valor representativo tradicional equivalente a Shapley para cada categoría de aeronaves en la CS.

Categorías de aeronaves	Valor
A	0,2
B	0,4
C	0,7
D	1
E	1,5
F	2,5

En ellas se refleja que para cada categoría, el valor representativo también recoge el aumento de la longitud de pista. Sin embargo, no presenta diferencias para la categoría de mayor tamaño de aeronaves, ya que mientras que en las Tablas 5-1 y 5-2 se detallan los costes reales, estos valores responden a la participación que tiene cada categoría sobre la distribución, y en ambas configuraciones, solamente F requiere de esta sección.

5.2. Distribución según valor Simplicial

Para este caso, con anterioridad al cálculo del valor Simplicial, en primer lugar se deben contabilizar todas las posibles coaliciones entre los participantes de un conjunto de juegos (como lo es la distribución de costes).

Ya que, en función de los potenciales jugadores N , en este caso las categorías de aeronaves, se podrán formar 2^N coaliciones, cada una de las cuales tendrá su respectiva función de coalición $v(S)$.

Como la cantidad de jugadores es idéntica para sendas configuraciones, y también las secciones en las que se divide cada pista, el número de coaliciones posibles que pueden emerger es igual para ambas.

Tabla 5-5. Número de coaliciones posibles asociados a cada sección de ambas configuraciones.

Secciones de pista	Número de coaliciones
A	64
B	32
C	16
D	8
E	4
F	2

En las siguientes tablas se detallan todas las posibles coaliciones, pues como ya se ha indicado, estas se deben contabilizar para el cálculo del valor Simplicial. Con objeto de automatizar el proceso, se ha empleado la codificación binaria de manera que si el jugador i participa de la coalición S , este tiene en la fila un valor igual a la unidad y un 0 en caso contrario. Si dos o más jugadores forman S , el valor de su función de utilidad es igual a la suma de dichas funciones en el caso de formar individualmente la coalición (cuyos valores son los recogidos en la Tabla 4-6 y Tabla 4-7 respectivamente para cada configuración, los cuales coinciden para ambas).

Tabla 5-6. Coaliciones y funciones correspondientes para el juego C de ambas configuraciones.

Coaliciones Juego C						
A	B	C	D	E	F	$v(S)$
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.2
0.0	0.0	0.0	0.0	1.0	0.0	0.2
0.0	0.0	0.0	1.0	0.0	0.0	0.2
0.0	0.0	1.0	0.0	0.0	0.0	0.2
0.0	0.0	0.0	0.0	1.0	1.0	0.5
0.0	0.0	0.0	1.0	0.0	1.0	0.5
0.0	0.0	0.0	1.0	1.0	0.0	0.5
0.0	0.0	1.0	0.0	0.0	1.0	0.5
0.0	0.0	1.0	0.0	1.0	0.0	0.5
0.0	0.0	1.0	1.0	0.0	0.0	0.5
0.0	0.0	0.0	1.0	1.0	1.0	0.8
0.0	0.0	1.0	0.0	1.0	1.0	0.8
0.0	0.0	1.0	1.0	0.0	1.0	0.8
0.0	0.0	1.0	1.0	1.0	0.0	0.8
0.0	0.0	1.0	1.0	1.0	1.0	1.0

Para el juego C (Tabla 5-6), donde solo intervienen los jugadores del C al F (como se refleja en las Tablas de la 4-4 a la 4-7), las funciones de las posibles coaliciones son:

$$v(\emptyset) = 0$$

$$v(C) = v(D) = v(E) = v(F) = 0,25 \approx 0,2$$

$$v(CD) = v(CE) = v(CF) = v(DE) = v(DF) = v(EF) = 0,25 + 0,25 = 0,5$$

$$v(CDE) = v(CDF) = v(CEF) = v(DEF) = 0,25 + 0,25 + 0,25 = 0,75 \approx 0,8$$

$$v(CDEF) = v(N) = 1$$

Tabla 5-7. Coaliciones y funciones correspondientes para el juego A de ambas configuraciones.

Coaliciones Juego A						
A	B	C	D	E	F	v(S)
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.2
0.0	0.0	0.0	0.0	1.0	0.0	0.2
0.0	0.0	0.0	1.0	0.0	0.0	0.2
0.0	0.0	1.0	0.0	0.0	0.0	0.2
0.0	1.0	0.0	0.0	0.0	0.0	0.2
1.0	0.0	0.0	0.0	0.0	0.0	0.2
1.0	0.0	0.0	0.0	0.0	1.0	0.3
1.0	0.0	0.0	0.0	1.0	0.0	0.3
0.0	1.0	1.0	0.0	0.0	0.0	0.3
1.0	0.0	0.0	1.0	0.0	0.0	0.3
0.0	1.0	0.0	1.0	0.0	0.0	0.3
0.0	1.0	0.0	0.0	1.0	0.0	0.3
0.0	1.0	0.0	0.0	0.0	1.0	0.3
1.0	0.0	1.0	0.0	0.0	0.0	0.3
0.0	0.0	1.0	1.0	0.0	0.0	0.3
0.0	0.0	1.0	0.0	1.0	0.0	0.3
0.0	0.0	1.0	0.0	0.0	1.0	0.3
1.0	1.0	0.0	0.0	0.0	0.0	0.3
0.0	0.0	0.0	1.0	1.0	0.0	0.3
0.0	0.0	0.0	1.0	0.0	1.0	0.3
0.0	0.0	1.0	0.0	0.0	1.0	0.3
1.0	1.0	0.0	0.0	0.0	0.0	0.5
1.0	0.0	1.0	0.0	1.0	0.0	0.5
1.0	0.0	0.0	1.0	1.0	0.0	0.5
1.0	0.0	0.0	1.0	0.0	1.0	0.5
1.0	0.0	1.0	0.0	0.0	1.0	0.5
1.0	1.0	0.0	0.0	1.0	0.0	0.5
0.0	1.0	0.0	1.0	1.0	0.0	0.5
0.0	1.0	1.0	1.0	0.0	0.0	0.5
0.0	1.0	1.0	0.0	1.0	0.0	0.5
0.0	1.0	1.0	0.0	0.0	1.0	0.5
1.0	0.0	1.0	1.0	0.0	0.0	0.5
0.0	1.0	0.0	1.0	0.0	1.0	0.5
0.0	1.0	0.0	0.0	1.0	1.0	0.5
1.0	1.0	1.0	0.0	0.0	0.0	0.5
0.0	0.0	1.0	1.0	1.0	0.0	0.5
0.0	0.0	1.0	0.0	1.0	1.0	0.5
0.0	0.0	0.0	1.0	1.0	1.0	0.5
1.0	1.0	0.0	1.0	0.0	0.0	0.5
1.0	1.0	0.0	0.0	1.0	1.0	0.7
1.0	1.0	0.0	1.0	0.0	1.0	0.7
1.0	1.0	0.0	1.0	1.0	0.0	0.7
1.0	1.0	1.0	1.0	0.0	0.0	0.7
0.0	1.0	1.0	1.0	0.0	1.0	0.7
1.0	0.0	1.0	1.0	0.0	1.0	0.7
1.0	0.0	1.0	0.0	1.0	1.0	0.7
1.0	0.0	0.0	1.0	1.0	1.0	0.7
0.0	1.0	1.0	1.0	1.0	0.0	0.7
1.0	1.0	1.0	0.0	0.0	1.0	0.7
0.0	1.0	0.0	1.0	1.0	1.0	0.7
0.0	0.0	1.0	1.0	1.0	1.0	0.7
1.0	0.0	1.0	1.0	1.0	0.0	0.7
1.0	1.0	1.0	0.0	1.0	0.0	0.7
1.0	1.0	1.0	1.0	0.0	1.0	0.8
1.0	1.0	1.0	0.0	1.0	1.0	0.8
0.0	1.0	1.0	1.0	1.0	1.0	0.8
1.0	0.0	1.0	1.0	1.0	1.0	0.8
1.0	1.0	1.0	1.0	1.0	0.0	0.8
1.0	1.0	0.0	1.0	1.0	1.0	0.8
1.0	1.0	1.0	1.0	1.0	1.0	1.0

Tabla 5-8. Coaliciones y funciones correspondientes para el juego B de ambas configuraciones.

Coaliciones Juego B						
A	B	C	D	E	F	v(S)
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.2
0.0	0.0	0.0	0.0	1.0	0.0	0.2
0.0	1.0	0.0	0.0	0.0	0.0	0.2
0.0	0.0	0.0	1.0	0.0	0.0	0.2
0.0	0.0	1.0	0.0	0.0	0.0	0.2
0.0	1.0	0.0	1.0	0.0	0.0	0.4
0.0	0.0	1.0	1.0	0.0	0.0	0.4
0.0	0.0	1.0	0.0	1.0	0.0	0.4
0.0	0.0	1.0	0.0	0.0	1.0	0.4
0.0	1.0	1.0	0.0	0.0	0.0	0.4
0.0	0.0	0.0	1.0	1.0	0.0	0.4
0.0	0.0	0.0	1.0	0.0	1.0	0.4
0.0	0.0	0.0	0.0	1.0	1.0	0.4
0.0	1.0	0.0	0.0	0.0	1.0	0.4
0.0	1.0	0.0	0.0	1.0	0.0	0.4
0.0	1.0	1.0	1.0	0.0	0.0	0.6
0.0	1.0	0.0	1.0	1.0	0.0	0.6
0.0	1.0	0.0	1.0	0.0	1.0	0.6
0.0	0.0	1.0	1.0	0.0	1.0	0.6
0.0	0.0	1.0	0.0	1.0	1.0	0.6
0.0	0.0	0.0	1.0	1.0	1.0	0.6
0.0	1.0	0.0	0.0	1.0	1.0	0.6
0.0	1.0	1.0	1.0	0.0	1.0	0.8
0.0	0.0	1.0	1.0	1.0	1.0	0.8
0.0	1.0	0.0	1.0	1.0	1.0	0.8
0.0	1.0	1.0	1.0	1.0	0.0	0.8
0.0	1.0	1.0	0.0	1.0	1.0	0.8
0.0	1.0	1.0	1.0	1.0	1.0	1.0

Tabla 5-9. Coaliciones y funciones correspondientes para el juego D de ambas configuraciones.

Coaliciones Juego D						
A	B	C	D	E	F	v(S)
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.3
0.0	0.0	0.0	0.0	1.0	0.0	0.3
0.0	0.0	0.0	1.0	0.0	0.0	0.3
0.0	0.0	0.0	0.0	1.0	1.0	0.7
0.0	0.0	0.0	1.0	0.0	1.0	0.7
0.0	0.0	0.0	1.0	1.0	0.0	0.7
0.0	0.0	0.0	1.0	1.0	1.0	1.0

Tabla 5-10. Coaliciones y funciones correspondientes para el juego E de ambas configuraciones.

Coaliciones Juego E						
A	B	C	D	E	F	v(S)
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.5
0.0	0.0	0.0	0.0	1.0	0.0	0.5
0.0	0.0	0.0	0.0	1.0	1.0	1.0

Tabla 5-11. Coaliciones y funciones correspondientes para el juego F de ambas configuraciones.

Coaliciones Juego F						
A	B	C	D	E	F	v(S)
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	1.0

Con objeto de ejemplificar detalladamente la simulación del juego C, seguidamente se calcula el valor de la etapa (término a término) correspondiente de cada jugador por participar en este juego. Para ello, se utilizan los valores recogidos en la Tabla 5-12 y se reformula la ecuación (1) aplicada a esta simulación particularizándola para el juego C.

Adicionalmente, algunas de las coaliciones posibles se han representado en las figuras consecutivas (de Figura 5-1 a Figura 5-3).

$$\Delta_C^C = \sum_{s=0}^3 \frac{1}{\binom{3}{s}} [v^C(K_s) - v^C(K_{s-1})] \frac{\text{Apariciones en nuevas coaliciones}}{s + 1}$$

$$\Delta_C^C = \frac{1}{\binom{3}{0}} [0,2 - 0] \frac{1}{(0 + 1)} + \frac{1}{\binom{3}{1}} [0,5 - 0,2] \frac{3}{(1 + 1)} +$$

$$+ \frac{1}{\binom{3}{2}} [0,8 - 0,5] \frac{3}{(2 + 1)} + \frac{1}{\binom{3}{3}} [1 - 0,8] \frac{1}{(3 + 1)} =$$

$$= \frac{1}{1} [0,2 - 0] \frac{1}{1} + \frac{1}{3} [0,5 - 0,2] \frac{3}{2} + \frac{1}{3} [0,8 - 0,5] \frac{3}{3} + \frac{1}{1} [1 - 0,8] \frac{1}{4} =$$

$$= 0,2 + 0,15 + 0,1 + 0,05 = 0,5$$

Cabe destacar, que al ser equiprobable la participación de los diferentes jugadores en las diversas coaliciones que se pueden formar (Tabla 5-12), los valores por etapa de cada uno de ellos según el juego son idénticos. Por lo que, para el juego C:

$$\Delta_C^C = \Delta_D^C = \Delta_E^C = \Delta_F^C$$

Tabla 5-12. Resumen de las coaliciones posibles y sus funciones en el juego C.

$v^C(K_{-1}) = 0$	$v^C(K_0) = 0,2$	$v^C(K_1) = 0,5$	$v^C(K_2) = 0,8$	$v^C(K_3) = 1$
\emptyset	C, D, E, F	CD, CE, CF, DE, DF, EF	CDE, CDF, CEF, DEF	CDEF

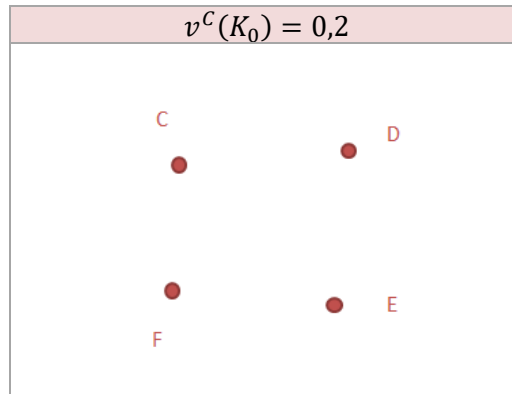


Figura 5–1. Representación de las coaliciones de dimensión 0 del juego C.

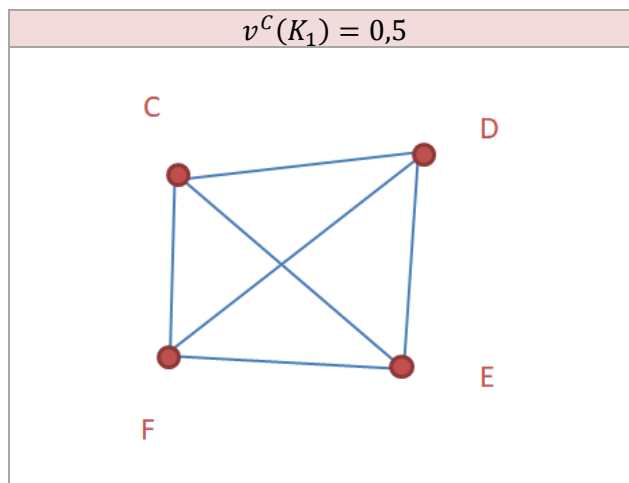


Figura 5–2. Representación de las coaliciones de dimensión 1 del juego C.

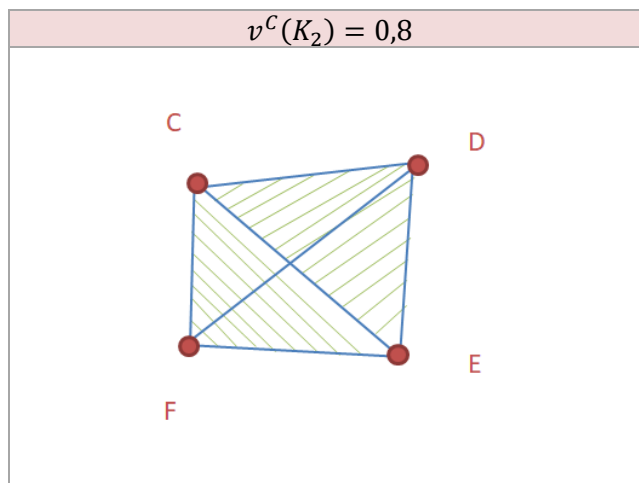


Figura 5–3. Representación de algunas de las coaliciones de dimensión 2 del juego C.

Repitiendo el mismo procedimiento para todos los juegos, se obtienen los valores por etapa para cada jugador que interviene en el juego, de manera que la suma de todos ellos da lugar al valor Simplicial asociado a cada jugador. En las Tablas 5-15 y 5-16, se muestran dichos índices para ambas configuraciones (redondeando al primer decimal como en el caso anterior).

$$\begin{aligned} \varphi_A &= \Delta_A^A \\ \varphi_B &= \Delta_B^A + \Delta_B^B \\ \varphi_C &= \Delta_C^A + \Delta_C^B + \Delta_C^C \\ \varphi_D &= \Delta_D^A + \Delta_D^B + \Delta_D^C + \Delta_D^D \\ \varphi_E &= \Delta_E^A + \Delta_E^B + \Delta_E^C + \Delta_E^D + \Delta_E^E \\ \varphi_F &= \Delta_F^A + \Delta_F^B + \Delta_F^C + \Delta_F^D + \Delta_F^E + \Delta_F^F \end{aligned}$$

Tabla 5-13. Valores por etapa de cada jugador para ambas configuraciones.

		Juegos					
		A	B	C	D	E	F
Jugadores	A	Δ_A^A	-	-	-	-	-
	B	Δ_B^A	Δ_B^B	-	-	-	-
	C	Δ_C^A	Δ_C^B	Δ_C^C	-	-	-
	D	Δ_D^A	Δ_D^B	Δ_D^C	Δ_D^D	-	-
	E	Δ_E^A	Δ_E^B	Δ_E^C	Δ_E^D	Δ_E^E	-
	F	Δ_F^A	Δ_F^B	Δ_F^C	Δ_F^D	Δ_F^E	Δ_F^F

Tabla 5-14. Valores calculados por etapa de cada jugador para ambas configuraciones.

		Juegos					
		A	B	C	D	E	F
Jugadores	A	0,42	0,0	0,0	0,0	0,0	0,0
	B	0,42	0,46	0,0	0,0	0,0	0,0
	C	0,42	0,46	0,50	0,0	0,0	0,0
	D	0,42	0,46	0,50	0,60	0,0	0,0
	E	0,42	0,46	0,50	0,60	0,75	0,0
	F	0,42	0,46	0,50	0,60	0,75	1,00

Tabla 5-15. Valor Simplicial para cada categoría de aeronaves en la configuración Norte.

Categorías de aeronaves	Valor
A	0,4
B	0,9
C	1,4
D	2,0
E	2,7
F	3,7

Tabla 5-16. Valor Simplicial para cada categoría de aeronaves en la configuración Sur.

Categorías de aeronaves	Valor
A	0,4
B	0,9
C	1,4
D	2,0
E	2,7
F	3,7

5.3. Comparación de las distribuciones

Al contrastar los valores, cabe destacar, que las distribuciones representativas resultantes para cada jugador según Shapley (Tabla 5-3 y Tabla 5-4) y el valor Simplicial (Tabla 5-15 y Tabla 5-16) no coinciden, pues mientras Shapley representa un reparto eficiente cuya suma de valores es el total, el nuevo valor alternativo, da lugar a la distribución que tiene que ser aplicada al total para hallar la distribución eficiente correspondiente a cada categoría. No obstante, en ambos casos, estos reflejan el orden creciente según la extensión que necesita cada categoría de aeronave para su operación (participando de la distribución con un valor mayor). Esto también se ilustra en la representación porcentual (sobre el total) de las distribuciones de las Figuras 5-4 y 5-5.

En relación con los valores asociados a cada jugador, como con los resultados derivados de Shapley, los valores Simpliciales también son iguales para ambas configuraciones de pistas, pues aunque los metros longitudinales del último tramo (F) difieran, bajo estos índices se contempla la parte correspondiente por el uso global según la categoría. De modo que al ser el único jugador para dicho juego, debe asumir el coste íntegro según el planteamiento de ambos índices.

Por otra parte, se resalta que para la obtención de los valores Simpliciales no se está asumiendo el cumplimiento del axioma lineal, mientras que para la solución única de Shapley, la condición de linealidad se asume desde el principio, restringiendo la aplicación del mismo.

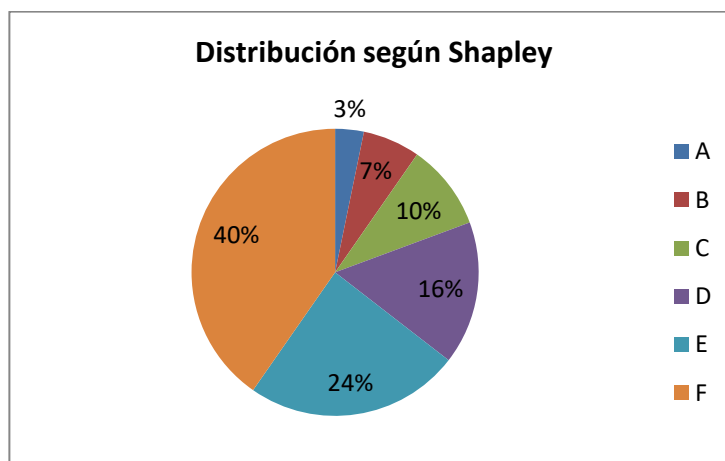


Figura 5-4. Distribución porcentual de la distribución de costes según Shapley.

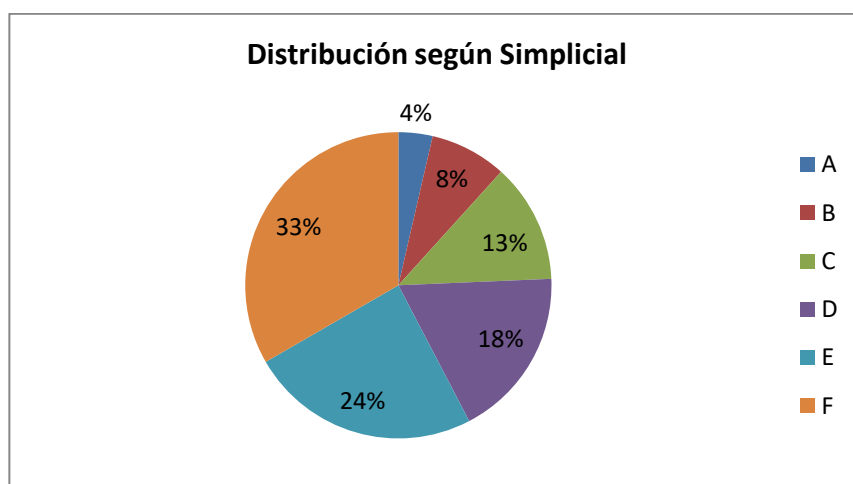


Figura 5–5. Distribución porcentual de la distribución de costes según valor Simplicial.

Respecto al caso simulado acerca del aeropuerto AS Madrid-Barajas, se señala que la contribución de las categorías no tiene por qué responder a un orden creciente, puesto que esta depende a su vez de la cantidad de aeronaves de una determinada categoría que hagan uso de las instalaciones.

6 PROGRAMA DESARROLLADO

En el presente capítulo se va a exponer la estructura del programa desarrollado para la simulación del caso del aeropuerto AS Madrid Barajas, por medio del cual se han obtenido los resultados mostrados en los capítulos anteriores.

No obstante, como ya se ha mencionado, el código completo se encuentra disponible en los anexos, así como los resultados que derivan de cada parte.

6.1. Estructura programa

El código programado en Python para la simulación sigue la estructura clásica característica de este lenguaje. Esta se refleja en el diagrama de flujo mostrado en la Figura 6–1.

En primer lugar, como es habitual en este lenguaje, se declaran todas las librerías de las que se va a hacer uso en el mismo. Esta es una de las principales ventajas de Python, pues estas son funciones que se importan y que ya han sido definidas previamente por otros programadores a bajo nivel, de forma que son muy útiles, y por lo general, permiten que los códigos sean más reducidos, evitando las partes que no generan valor.

Seguidamente, se han definido las funciones que han sido específicamente programadas para este trabajo, de las cuales en un apartado posterior se comentarán sus entradas y salidas. Como cualquier función, el potencial de las mismas reside en que solo se han tenido que definir en una ocasión, sin embargo, se van a llamar reiteradamente a lo largo del código, sobre todo teniendo en cuenta que los planteamientos para las configuraciones Norte y Sur son equivalentes.

El bloque consecutivo se corresponde con la definición de todos los datos necesarios para realizar la simulación. Estos principalmente se tratan de las secciones en las que se han dividido las pistas del aeropuerto así como su coste, los cuales ya se han indicado en el capítulo 4.

Posteriormente, se define el juego inicial también presentado en el capítulo 4, de manera que este resulta en la distribución equitativa expresada en términos totales y en tanto por uno de las secciones de pista entre los diferentes jugadores (las categorías de aeronaves). Tal y como ya se ha expuesto, estos valores son la base de los sucesivos cálculos para obtener ambos índices.

A partir de este punto, se podría bien calcular tan solo uno de los índices por jugador, o ambos de ellos. Como además el código se ha programado en bloques separados de la plataforma Google Colaboratory, estos pueden ser o no ejecutados de manera independiente y en cualquier orden.

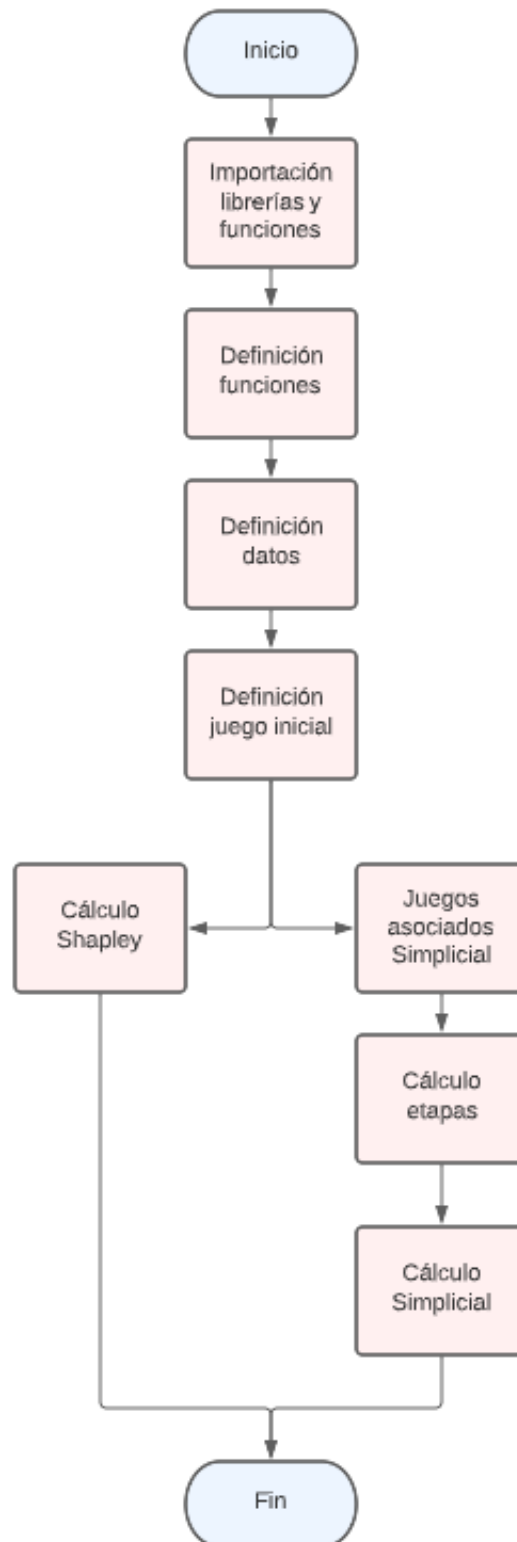


Figura 6–1. Diagrama de flujo del programa desarrollado.

En relación con el cálculo según Shapley, este es muy sencillo para este caso en particular, pues solo requiere de una adición acumulada. Sin embargo, los valores Simpliciales requieren de un preprocesamiento de la base del juego para automatizar el cálculo de los mismos.

Para ello, primero se desarrollan todas las posibles coaliciones que pueden suceder en cada juego asociado a cada sección de pista, por lo que en cada uno de ellos la cantidad de jugadores difiere, pues a medida que las aeronaves no precisan de esos metros para el despegue estas no participan de la distribución de costes. Como se especifica en el capítulo 5, se ha utilizado la codificación binaria para la automatización del proceso y obtención de todas las coaliciones.

Tras la obtención de las utilidades de dichas coaliciones, las cuales se han definido de manera aditiva según los jugadores que las formen, se hallan los valores por etapa de cada jugador, lo cual es el equivalente de la distribución equitativa para el índice lineal según Shapley. De modo que similarmente al índice tradicional, para esta nueva alternativa, también se calcula la adición de los términos que le corresponden a cada jugador por participar de un juego.

Finalmente, cabe destacar que un código puede ser programado de múltiples formas y no existe un camino único. Por lo que este se ha realizado de la manera que se ha considerado más clara y precisa para la finalidad señalada en los objetivos.

6.2. Librerías

Las librerías importadas evitan la programación a bajo nivel facilitando y agilizando el proceso. Particularmente, para este trabajo no se han requerido demasiadas y las empleadas están muy extendidas.

Numpy

Es un módulo que permite el uso de matrices además de un conjunto de funciones para procesar y operar con las mismas. Principalmente, este se ha utilizado para almacenar datos en estructuras matriciales, pues la mayoría de las tablas siguen esta configuración, donde las filas se corresponden con los jugadores y las columnas con los juegos correspondientes a cada sección de pista.

Math

Es una librería que recoge funciones y valores matemáticos en el campo de los números reales. En este trabajo se ha utilizado para el cálculo factorial de los números combinatorios.

Prettytable

Es una función importada de la librería con la misma denominación. Esta permite presentar los resultados de manera organizada en formato tabla. No obstante, debido a su definición solo puede estar formada por un encabezado superior diferenciado y una serie de filas idénticas. Los datos se pueden agregar tanto en filas como en columnas.

Fraction

Al igual que la anterior, esta función procede de una librería con el mismo nombre y sirve para convertir en formato fraccional los decimales definidos. Cabe destacar que para ello, estos tienen que ser anteriormente convertidos en cadenas de caracteres y que en ocasiones no refleja el valor oportuno debido a la inexactitud del cálculo decimal computacional.

Itertools

Se trata de una librería nativa que facilita la creación y gestión de operaciones iterativas. Específicamente, se ha empleado la función producto, la cual permite generar todas las posibles combinaciones binarias a partir del número de participantes en una sola línea de código, por lo que se considera la función más destacable de las empleadas en base a su utilidad, pues estas son la clave para el cálculo de los valores Simpliciales.

6.3. Funciones

En total han sido 8 funciones las programadas para este trabajo, de las cuales cabe mencionar que 3 están relacionadas con la representación en tablas de los resultados, 2 con posibles redondeos para eliminar términos no relevantes y el resto, son implementaciones necesarias para los cálculos de los índices.

Debido a la proporción de 5 funciones útiles para la identificación y comprensión de resultados frente a las 3 relacionadas con el cálculo, se destaca la relevancia del formato para la finalidad de la simulación. Ya que a partir de los resultados obtenidos es que se obtendrán las conclusiones, por lo que se precisa de una presentación clara.

A continuación, se van a detallar la utilidad y las cabeceras de todas las funciones, las cuales están configuradas por la palabra clave `def`, el nombre de la función y las `entradas` de la misma entre paréntesis. A diferencia de otros lenguajes, la `salida` se define a lo largo o al final de su contenido, y no pertenece a la cabecera. Cabe destacar que las variables locales y globales funcionan de manera similar a otros lenguajes.

Imprimir matriz

En esta función se define una tabla a partir de un matriz `m` que es entrada de la misma, junto a una serie de caracteres que servirán para definir la cabecera de la tabla (`den`) y otra cadena de caracteres para añadirle título a la misma.

La salida, en lugar de ser devuelta al llamar a la función, es directamente impresa por pantalla.

```
def imprimir_matriz(m,den,titulotabla):  
    ...  
    print(Tabla.get_string(title=titulotabla))
```

Imprimir coaliciones

Al igual que la anterior, sirve para imprimir una tabla de resultados, pero aquí los datos se añaden a la tabla en forma de fila. No obstante, los elementos de los que consta son idénticos.

```
def imprimir_coaliciones(m,den,titulotabla):  
    ...  
    print(Tabla.get_string(title=titulotabla))
```

Imprimir vector

Similarmente a las anteriores, esta función saca por pantalla una tabla que en lugar de ser una matriz, viene de un vector (v), que se crea añadiendo esa única fila a la estructura.

```
def imprimir_vector(v,den,titulotabla):
    ...
    print(Tabla.get_string(title=titulotabla))
```

Redondeo v

La aplicación de esta función es simplemente redondear el valor de la utilidad de las diferentes coaliciones. Sus entradas son un matriz (M) de la que forman parte estos valores en la última columna y la cifra decimal a la cual se quiere redondear, es decir, si se tiene por ejemplo 2,55, y se indica como entrada para redondear 1 (dec_red), el resultado será 2,6. Cabe destacar que para este trabajo se redondea hacia arriba a partir del 5, y hacia el número menor cuando es menor de 5.

La salida es la propia matriz, la cual se devuelve pero con la última fila redondeada.

```
def redondeo_v(M,dec_red):
    ...
    return M
```

Redondeo M

La utilidad de esta función es similar a la anterior, por lo que está conformada por los mismos elementos, pero en este caso se redondean todos los términos de la matriz.

```
def redondeo_M(M,dec_red):
    ...
    return M
```

Num combinatorio

En esta función se calcula el resultado de un número combinatorio siguiendo la fórmula clásica matemática definida a partir de los términos factoriales. Por lo que las entradas se corresponden con dichos valores n y m , y se devuelve el mencionado resultado.

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

```
def num_combinatorio(n,m):
    ...
    return nCm
```

Juegos asociados

Como ya se ha indicado, a partir de esta función se definen los juegos asociados a las secciones de las pistas. Esta se subdivide en tres etapas que a continuación se especifican.

En primer lugar, se obtiene la cantidad de coaliciones posibles en base a los jugadores que participan en cada juego asociado. Seguidamente, se definen todas las coaliciones por codificación binaria y, a partir de las mismas, se calcula el valor de la función de utilidad por adición de cada una tal y como anteriormente se ha especificado. Respecto a estos valores hallados, cabe destacar que se ordenan de menor a mayor con objeto de facilitar el análisis y las posteriores operaciones relacionadas con las filtraciones.

Las entradas de esta función se corresponden con la matriz resultante del juego inicial (M) sobre la que se basan los posteriores desarrollos, las cadenas de caracteres para denominar las cabeceras de las tablas (den , den_coa) y el número tanto de jugadores como de juegos de los que se desean obtener los comentados valores.

Respecto a las salidas de esta función, cabe mencionar que algunas se sacan directamente por pantalla, y la única que se devuelve, se corresponde con una lista ($v_posibles$) en la que en cada posición se guarda una matriz concerniente al juego asociado de cada sección (la cual recoge las combinaciones binarias de las coaliciones y sus funciones de utilidad correspondientes).

```
def juegos_asociados(M,den,den_coa,njuegos,njugadores):
    ...
    return v_posibles
```

C marginales juegos

En esta función se calculan los valores por etapa que le corresponden a cada jugador por juego en dos pasos.

Primeramente, se distinguen las distintas funciones de utilidad determinadas en cada juego, ya que es relevante contabilizar las coaliciones agrupadas por cada una de ellas así como sus valores. Y así, a continuación poder calcular dichos valores por etapa según la fórmula indicada en el capítulo 5.

La entrada a esta función es la lista que proviene de la anteriormente definida (Mv), así como el número de jugadores y juegos. Por otro lado, la variable devuelta de salida $C_marginales$ es una matriz que contiene todas las contribuciones ordenadas por filas y columnas respectivamente para jugador y juego, siguiendo la estructura de la tabla base del juego inicial.

```
def C_marginales_juegos(Mv,njuegos,njugadores):
    ...
    return C_marginales
```


7 CONCLUSIONES

7.1. Conclusiones del trabajo realizado

Dado que el principal propósito de este trabajo consiste en resolver y demostrar el juego de aeropuertos aplicando el valor Simplicial como alternativa a la distribución eficiente ya demostrada por Littlechild y Owen (equivalente a solución única de Shapley), se puede concluir que el objetivo indicado se ha logrado.

De manera que es patente la resolución del juego de costes desde otra perspectiva menos restrictiva, pues aunque partiendo de los mismos datos representativos, los planteamientos de cálculo seguidos difieren en una de sus características esenciales. Ya que, como se puede comprobar comparando el detalle del juego C, al contrario que los valores lineales según Shapley, la aproximación por el fundamento no lineal del Simplicial desestima la asunción del axioma lineal del índice tradicional.

En relación con dicha relevante desigualdad, el marco de aplicación del valor Simplicial se extiende, pues las limitaciones son menos impositivas, y por ende, el conjunto de juegos resolubles también se amplía.

No obstante, de las distribuciones de ambos valores, se observa el beneficio común de repartir los costes a partir de una alianza o coalición entre jugadores, puesto que si cada miembro tuviese que asumir la cuantía completa individualmente quizás esta no podría ser retribuida. De modo que la unión de fuerzas para el provecho conjunto trae consigo ventajas obvias (ahorro en costes) desde cualquier aproximación empleada.

Por otra parte, se destaca el interés mostrado en el potencial del valor Simplicial al desarrollar este trabajo. Pues su contraste con dos elementos recurrentes del estado del arte de la Teoría de Juegos, como son el juego del aeropuerto y la solución de Shapley, demuestra que a pesar de disponer de soluciones eficientes ampliamente extendidas, en ocasiones es necesario ampliar el cerco con objeto de poder incluir en este ámbito otros contextos que no cumplan con los axiomas clásicos.

Al margen de la contribución puramente teórica, la simulación de un caso real como es el aeropuerto AS Madrid-Barajas (un icono de las infraestructuras aéreas españolas) ante un escenario como el actual, donde este medio de transporte se está popularizando en contraste con la preocupación generalizada por el ajuste de los costes (los cuales están a la alza por diversos motivos sociopolíticos y económicos), supone un estudio conveniente ante la búsqueda de alternativas que sirvan para arrojar claridad sobre la resolución de estos dilemas dando lugar a otros modelos de reparto, puesto que sus consecuencias, la mayoría del tiempo, recaen directamente sobre el propio usuario.

7.2. Líneas futuras

Tras concluir el desarrollo del análisis planteado, en este apartado, se mencionan otras propuestas en las que se podría trabajar para avanzar en el presente estudio realizado (tanto a nivel teórico como práctico).

En primer lugar, se expone la cuestión de distribuir los costes no sólo en base al uso propio que se realice de la infraestructura común, sino que se extienda a la importancia relativa que pueda tener para ese jugador. Pues aunque en el juego del aeropuerto no tenga tanta relevancia, si se aplican estos índices a un servicio de renting o carpooling, la premura o necesidad del traslado podrían ser variables de interés.

Por otra parte, una perspectiva menos individualista podría establecer el bien común como premisa, de manera que las partes con mayor poder o capacidad asistan a las menos pudientes o desfavorecidas. En este sentido, se han definido índices denominados solidarios, que distribuyen los costes en función de esta vertiente colaborativa.

Sin embargo, si se aplica a empresas, la priorización del beneficio propio es una realidad a pesar de sus principios o valores de marca. De modo que, un punto intermedio que combine varias perspectivas también sería atractivo de comparar con los demás planteamientos referidos.

REFERENCIAS

- [1] “Cambio climático: ¿cuál es el medio de transporte que más contamina? | El Economista.” [Online]. Available: <https://www.economista.net/tendencias/Cambio-climatico-cual-es-el-medio-de-transporte-que-mas-contamina-20190826-0038.html>. [Accessed: 16-Nov-2022].
- [2] “Las claves de la COP27, la cumbre mundial del clima de Egipto.” [Online]. Available: https://www.nationalgeographic.com.es/ciencia/estas-son-claves-cop27-cumbre-mundial-clima_19003. [Accessed: 16-Nov-2022].
- [3] “Instituto Nacional de Estadística.” [Online]. Available: <https://ine.es/>.
- [4] B. Peleg, *Introduction to the Theory of Cooperative Games*, 2nd ed. 2007. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.
- [5] M. Wooldridge, “Does game theory work?,” *IEEE Intell. Syst.*, vol. 27, no. 6, pp. 76–80, 2012.
- [6] “Game Theory (Stanford Encyclopedia of Philosophy).” [Online]. Available: <https://plato.stanford.edu/entries/game-theory/>. [Accessed: 25-Oct-2022].
- [7] D. G. Arce, “Economics, Ethics and the Dilemma in the Prisoner’s Dilemmas,” *Am. Econ.*, vol. 55, no. 1, pp. 49–57, 2010.
- [8] “Datos de negocio del aeropuerto de Madrid Adolfo Suárez | Aena.” [Online]. Available: <https://www.aena.es/es/aerolineas/aeropuertos-y-destinos/nuestros-aeropuertos/adolfo-suarez-madrid-barajas.html>. [Accessed: 25-Oct-2022].
- [9] “Plan Director del aeropuerto Adolfo Suárez Madrid-Barajas | Ministerio de Transportes, Movilidad y Agenda Urbana.” [Online]. Available: <https://www.mitma.gob.es/areas-de-actividad/aviacion-civil/politicas-aeroportuarias/integracion-territorial-aeroportuaria/planes-directores/plan-director-del-aeropuerto-de-madrid-barajas>. [Accessed: 22-Sep-2022].
- [10] “AIP España.” [Online]. Available: <https://aip.enaire.es/AIP/>. [Accessed: 22-Sep-2022].
- [11] “Modern Airlines and all things aviation.” [Online]. Available: <https://modernairliners.com/>. [Accessed: 22-Sep-2022].
- [12] “Aero Corner.” [Online]. Available: <https://aerocorner.com/>. [Accessed: 22-Sep-2022].
- [13] “About The ATR Aircraft Family | ATR Aircraft.” [Online]. Available: <https://www.atr-aircraft.com/our-aircraft/aircraft-family/>. [Accessed: 23-Sep-2022].

- [14] “TIOBE Index - TIOBE.” [Online]. Available: <https://www.tiobe.com/tiobe-index/>. [Accessed: 03-Nov-2022].
- [15] P. García Entrambasaguas, *Aprendiendo a programar en Python* . Málaga: UMA, 2019.

ANEXO A: CÓDIGO

Librerías importadas

```
import numpy as np
import math
from prettytable import PrettyTable
from fractions import Fraction
import itertools
```

Funciones

```
def imprimir_matriz(m,den,titulotabla):
    Tabla = PrettyTable()
    Tabla.add_column(" ",den)
    for j in range(len(den)):
        Tabla.add_column(den[j],m[:,j])
    print(Tabla.get_string(title=titulotabla))
```

```
def imprimir_coaliciones(m,den,titulotabla):
    Tabla = PrettyTable(den)
    for i in range(np.shape(m)[0]):
        Tabla.add_row(m[i,:])
    print(Tabla.get_string(title=titulotabla))
```

```
def imprimir_vector(v,den,titulotabla):
    Tabla = PrettyTable(den)
    Tabla.add_row(v)
    print(Tabla.get_string(title=titulotabla))
```

```
def redondeo_v(M,dec_red):
    for i in range(0,len(M)):
        M[i,-1]=round(M[i,-1],dec_red)
    return M
```

```
def redondeo_M(M,dec_red):
    for i in range(0,np.shape(M)[0]):
        for j in range(0,np.shape(M)[1]):
            M[i,j]=round(M[i,j],dec_red)
    return M
```

```
def num_combinatorio(n,m):
    nCm=math.factorial(n)/(math.factorial(m)*math.factorial(n-m))
    return nCm
```

```

def juegos_asociados(M, den, den_coa, njuegos, njugadores):
    Num_coa_posibles=np.arange(njuegos)*0
    for j in range (0,njuegos):
        Num_coa_posibles[j]=pow(2, (njugadores-j))
        imprimir_vector(np.array(Num_coa_posibles), den, "Número coaliciones posibles en los juegos asociados a cada sección")

    coa_posibles=list()
    for j in range (0,njuegos):
        coa_posibles.append(list(itertools.product([0, 1], repeat=(njugadores-j))))

    v_posibles=list()

    for j in range (0,njuegos):
        M_coa=np.zeros((Num_coa_posibles[j], (njuegos+1)))
        for k in range (0,len(coa_posibles[j])):
            v_coa=0
            for t in range (0,len(coa_posibles[j][k])):
                if coa_posibles[j][k][t]==1:
                    M_coa[k, (t+j)]=1
                    v_coa=v_coa+M[(t+j), j]
            M_coa[k, 6]=round(v_coa, 1)
            M_coa=M_coa[M_coa[:, -1].argsort()]
            v_posibles.append(M_coa)
            imprimir_coaliciones(v_posibles[j], den_coa, ("Coaliciones Juego"+" "+den_coa[j]))
        return v_posibles

def C_marginales_juegos(Mv, njuegos, njugadores):
    vjuego=list()
    for i in range (0,njuegos):
        v_aux=np.arange(njuegos-i)*0.0
        for j in range (0,len(v_aux)):
            for k in range (0,len(Mv[i])):
                if j==0 and Mv[i][k, -1]!=0 and v_aux[j]==0:
                    v_aux[j]=Mv[i][k, -1]
                    break
                elif j!=0 and v_aux[j]<v_aux[j-1] and Mv[i][k, -1]!=v_aux[j-1]:
                    v_aux[j]=Mv[i][k, -1]
            vjuego.append(v_aux)

    C_marginales=np.zeros((njuegos, njugadores))
    for i in range(0, njuegos):
        for j in range(0, njugadores):
            for k in range(0, len(vjuego[i])):
                apariciones=0
                for r in range(0, len(Mv[i])):
                    if Mv[i][r, -1]==vjuego[i][k] and Mv[i][r, j]==1:

```



```

        apariciones=apariciones+1
    if k==0:
        C_marginales[i,j]=C_marginales[i,j]+[(1/num_combinatorio((len(vj
uego[i])-1),(k)))*(vjuego[i][k]-0)*(apariciones/(k+1)))]
    else:
        C_marginales[i,j]=C_marginales[i,j]+[(1/num_combinatorio((len(vj
uego[i])-1),(k)))*(vjuego[i][k]-vjuego[i][k-1])*(apariciones/(k+1)))]
    return C_marginales

```

Datos aeropuerto

```

DN= 4610 # metros de la pista despegue Norte (36L)
DS= 4200 # metros de la pista despegue Sur (15)
C= 4000 # coste en euros del metro lineal de pista
TPos=180 # numero de posiciones totales para aeronaves

```

Datos distribución coste

```

ctgs_aero=6 # categorias de aeronaves
sec_pista=6 # secciones de pista
den=["A", "B", "C", "D", "E", "F"]

TORmax_A=1500 # m
TORmax_B=2000 # m
TORmax_C=2300 # m
TORmax_D=2500 # m
TORmax_E=2800 # m
TORmin_F=2800 # m
TORlim=[TORmax_A, TORmax_B, TORmax_C, TORmax_D, TORmax_E, TORmin_F]

```

Configuración Norte

```

#Coste de las secciones de la Pista Norte
Coste_sec_DN_A=TORmax_A*C
Coste_sec_DN_B=(TORmax_B-TORmax_A)*C
Coste_sec_DN_C=(TORmax_C-TORmax_B)*C
Coste_sec_DN_D=(TORmax_D-TORmax_C)*C
Coste_sec_DN_E=(TORmax_E-TORmax_D)*C
Coste_sec_DN_F=(DN-TORmin_F)*C
Coste_sec_DN=[Coste_sec_DN_A, Coste_sec_DN_B, Coste_sec_DN_C, Coste_sec_DN_D,
Coste_sec_DN_E, Coste_sec_DN_F]
Coste_sec_DN=np.array(Coste_sec_DN)

```

```

print("-----
Caso: una aeronave por categoría en configuración Norte-----
-----")

MDNud =np.ones((ctgs_aero,sec_pista))
MDNud_frac=np.ones((ctgs_aero,sec_pista))
MDNud_frac_total=np.ones((ctgs_aero,sec_pista))
Totales_sec_DN=np.arange(sec_pista)*0
Totales_ctgs_DN=np.arange(sec_pista)*0
for j in range (0,sec_pista):
    for i in range (0,ctgs_aero):
        if i<j or Coste_sec_DN[i]==0:
            MDNud[i,j]=0
        else:
            MDNud[i,j]=Coste_sec_DN[j]/(ctgs_aero-j-
len(Coste_sec_DN[np.where(Coste_sec_DN == 0)]))
            """"MDNud[i,j]=round(MDNud[i,j],6) #Redondeo""""
#Comprobación
            Totales_sec_DN[j]=sum(MDNud[:,j])

for i in range (0,ctgs_aero):
    Totales_ctgs_DN[i]=sum(MDNud[i,:])

imprimir_vector(Totales_sec_DN,den,"Costes totales secciones configuración
Norte")
imprimir_matriz(MDNud,den,"Costes pista despegue configuración Norte por c
ategoría")
imprimir_vector(Totales_ctgs_DN,den,"Costes totales categorías configuraci
ón Norte")

for j in range (0,sec_pista):
    for i in range (0,ctgs_aero):
        MDNud_frac[i,j]=(MDNud[i,j]/Totales_sec_DN[j])
        """"MDNud_frac[i,j]=round(MDNud_frac[i,j],6)""""
        """"print(Fraction(str(MDNud_frac[i,j])))""""

imprimir_matriz(MDNud_frac,den,"Fracción del total de sección por categorí
a pista despegue configuración Norte por categoría")

```

```
-----Caso: una aeronave por categoría en configuración Norte-----
```

Costes totales secciones configuración Norte						
A	B	C	D	E	F	
6000000	2000000	1200000	800000	1200000	7240000	

Costes pista despegue configuración Norte por categoría						
	A	B	C	D	E	F
A	1000000.0	0.0	0.0	0.0	0.0	0.0
B	1000000.0	400000.0	0.0	0.0	0.0	0.0
C	1000000.0	400000.0	300000.0	0.0	0.0	0.0
D	1000000.0	400000.0	300000.0	266666.6666666667	0.0	0.0
E	1000000.0	400000.0	300000.0	266666.6666666667	600000.0	0.0
F	1000000.0	400000.0	300000.0	266666.6666666667	600000.0	7240000.0

Costes totales categorías configuración Norte						
A	B	C	D	E	F	
1000000	1400000	1700000	1966666	2566666	9806666	

Fracción del total de sección por categoría pista despegue configuración Norte por categoría						
	A	B	C	D	E	F
A	0.16666666666666666	0.0	0.0	0.0	0.0	0.0
B	0.16666666666666666	0.2	0.0	0.0	0.0	0.0
C	0.16666666666666666	0.2	0.25	0.0	0.0	0.0
D	0.16666666666666666	0.2	0.25	0.3333333333333337	0.0	0.0
E	0.16666666666666666	0.2	0.25	0.3333333333333337	0.5	0.0
F	0.16666666666666666	0.2	0.25	0.3333333333333337	0.5	1.0

Configuración Sur

#Coste de las secciones de la Pista Sur

```
Coste_sec_DS_A=TORmax_A*C
Coste_sec_DS_B=(TORmax_B-TORmax_A)*C
Coste_sec_DS_C=(TORmax_C-TORmax_B)*C
Coste_sec_DS_D=(TORmax_D-TORmax_C)*C
Coste_sec_DS_E=(TORmax_E-TORmax_D)*C
Coste_sec_DS_F=(DS-TORmin_F)*C
Coste_sec_DS=[Coste_sec_DS_A, Coste_sec_DS_B, Coste_sec_DS_C, Coste_sec_DS_D,
Coste_sec_DS_E, Coste_sec_DS_F]
Coste_sec_DS=np.array(Coste_sec_DS)
```

```
print("-----
Caso: una aeronave por categoría en configuración Sur-----
---")
MDSud =np.ones((ctgs_aero,sec_pista)) # matriz DS
MDSud_frac=np.ones((ctgs_aero,sec_pista))
MDSud_frac_total=np.ones((ctgs_aero,sec_pista))
Totales_sec_DS=np.arange(sec_pista)*0
Totales_ctgs_DS=np.arange(sec_pista)*0
for j in range (0,sec_pista):
```

```

for i in range (0,ctgs_aero):
    if i<j or Coste_sec_DS[i]==0:
        MDSud[i,j]=0
    else:
        MDSud[i,j]=Coste_sec_DS[j]/(ctgs_aero-j-
len(Coste_sec_DS[np.where(Coste_sec_DS == 0)]))
#Comprobación
Totales_sec_DS[j]=sum(MDSud[:,j])

for i in range (0,ctgs_aero):
    Totales_ctgs_DS[i]=sum(MDSud[i,:])

imprimir_vector(Totales_sec_DS,den,"Costes totales secciones configuración
Sur")
imprimir_matriz(MDSud,den,"Costes pista despegue configuración Sur por cat
egoría")
imprimir_vector(Totales_ctgs_DS,den,"Costes totales categorías configuraci
ón Sur")

for j in range (0,sec_pista):
    for i in range (0,ctgs_aero):
        MDSud_frac[i,j]=(MDSud[i,j]/Totales_sec_DS[j])
        """MDSud_frac[i,j]=round(MDSud_frac[i,j],2)""" #Redondeo
        """print(Fraction(str(MDSud_frac[i,j])))"""

imprimir_matriz(MDSud_frac,den,"Fracción del total de sección por categorí
a pista despegue configuración Sur por categoría")

```

```
-----Caso: una aeronave por categoría en configuración Sur-----
```

Costes totales secciones configuración Sur						
A	B	C	D	E	F	
6000000	2000000	1200000	800000	1200000	5600000	

Costes pista despegue configuración Sur por categoría						
	A	B	C	D	E	F
A	1000000.0	0.0	0.0	0.0	0.0	0.0
B	1000000.0	400000.0	0.0	0.0	0.0	0.0
C	1000000.0	400000.0	300000.0	0.0	0.0	0.0
D	1000000.0	400000.0	300000.0	266666.6666666667	0.0	0.0
E	1000000.0	400000.0	300000.0	266666.6666666667	600000.0	0.0
F	1000000.0	400000.0	300000.0	266666.6666666667	600000.0	5600000.0

Costes totales categorías configuración Sur						
A	B	C	D	E	F	
1000000	1400000	1700000	1966666	2566666	8166666	

Fracción del total de sección por categoría pista despegue configuración Sur por categoría						
	A	B	C	D	E	F
A	0.16666666666666666	0.0	0.0	0.0	0.0	0.0
B	0.16666666666666666	0.2	0.0	0.0	0.0	0.0
C	0.16666666666666666	0.2	0.25	0.0	0.0	0.0
D	0.16666666666666666	0.2	0.25	0.33333333333333337	0.0	0.0
E	0.16666666666666666	0.2	0.25	0.33333333333333337	0.5	0.0
F	0.16666666666666666	0.2	0.25	0.33333333333333337	0.5	1.0

Valores según Shapley

```
print("-----
Caso: una aeronave por categoría en configuración Norte (costes en tanto p
or uno)-----")
Cjugador_DNud_frac=np.arange(ctgs_aero)*0.0
for i in range(0,ctgs_aero):
    Cjugador_DNud_frac[i]=sum(redondeo_M(MDNud_frac,2)[i,:])
imprimir_vector(Cjugador_DNud_frac,den,"Costes según Shapley por categoría
de avión")
```

```
-----Caso: una aeronave por categoría en configuración Norte (costes en tanto por uno)-----
```

Costes según Shapley por categoría de avión						
A	B	C	D	E	F	
0.17	0.37	0.62	0.95	1.45	2.45	

```
print("-----
Caso: una aeronave por categoría en configuración Sur (costes en tanto por
uno)-----")
Cjugador_DSud_frac=np.arange(ctgs_aero)*0.0
for i in range(0,ctgs_aero):
```

```

Cjugador_DSud_frac[i]=sum(redondeo_M(MDSud_frac,2)[i,:])
imprimir_vector(Cjugador_DSud_frac,den,"Costes según Shapley por categoría
de avión")

```

```

-----Caso: una aeronave por categoría en configuración Sur (costes en tanto por uno)-----
+-----+
| Costes según Shapley por categoría de avión |
+-----+-----+-----+-----+-----+-----+
| A | B | C | D | E | F |
+-----+-----+-----+-----+-----+-----+
| 0.17 | 0.37 | 0.62 | 0.95 | 1.45 | 2.45 |
+-----+-----+-----+-----+-----+-----+

```

Juegos asociados

```

den_coa=["A", "B", "C", "D", "E", "F", "v(S)"]
print("-----")
Caso: una aeronave por categoría en configuración Norte (costes en tanto p
or uno)-----")
MJ_DNud_frac=juegos_asociados(MDNud_frac,den,den_coa,sec_pista,ctgs_aero)

```

Coaliciones Juego A						
A	B	C	D	E	F	v(S)
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.2
0.0	0.0	0.0	0.0	1.0	0.0	0.2
0.0	0.0	1.0	0.0	0.0	0.0	0.2
0.0	1.0	0.0	0.0	0.0	0.0	0.2
1.0	0.0	0.0	0.0	0.0	0.0	0.2
1.0	0.0	0.0	0.0	0.0	1.0	0.3
1.0	0.0	0.0	0.0	1.0	0.0	0.3
0.0	1.0	1.0	0.0	0.0	0.0	0.3
1.0	0.0	0.0	1.0	0.0	0.0	0.3
0.0	1.0	0.0	0.0	1.0	0.0	0.3
0.0	1.0	0.0	0.0	0.0	1.0	0.3
1.0	0.0	1.0	0.0	0.0	0.0	0.3
0.0	0.0	1.0	1.0	0.0	0.0	0.3
0.0	0.0	1.0	0.0	0.0	1.0	0.3
1.0	1.0	0.0	0.0	0.0	0.0	0.3
0.0	0.0	0.0	1.0	1.0	0.0	0.3
0.0	0.0	0.0	1.0	0.0	1.0	0.3
1.0	1.0	0.0	0.0	0.0	1.0	0.3
0.0	0.0	0.0	0.0	1.0	1.0	0.3
1.0	1.0	0.0	0.0	0.0	1.0	0.5
1.0	0.0	1.0	0.0	1.0	0.0	0.5
1.0	0.0	0.0	1.0	0.0	1.0	0.5
1.0	0.0	0.0	0.0	1.0	1.0	0.5
1.0	0.0	1.0	0.0	0.0	1.0	0.5
1.0	1.0	0.0	0.0	1.0	0.0	0.5
0.0	1.0	0.0	1.0	1.0	0.0	0.5
0.0	1.0	1.0	1.0	0.0	0.0	0.5
0.0	1.0	1.0	0.0	1.0	0.0	0.5
0.0	1.0	0.0	1.0	0.0	1.0	0.5
1.0	1.0	1.0	0.0	0.0	0.0	0.5
0.0	0.0	1.0	1.0	1.0	0.0	0.5
0.0	0.0	1.0	1.0	0.0	1.0	0.5
0.0	0.0	1.0	0.0	1.0	1.0	0.5
0.0	0.0	0.0	1.0	1.0	1.0	0.5
1.0	1.0	0.0	1.0	0.0	0.0	0.5
1.0	1.0	0.0	0.0	1.0	1.0	0.7
1.0	1.0	0.0	1.0	0.0	1.0	0.7
1.0	1.0	0.0	1.0	1.0	0.0	0.7
1.0	1.0	1.0	1.0	0.0	0.0	0.7
0.0	1.0	1.0	1.0	0.0	1.0	0.7
1.0	0.0	1.0	1.0	0.0	1.0	0.7
1.0	0.0	0.0	1.0	1.0	1.0	0.7
0.0	1.0	1.0	1.0	1.0	0.0	0.7
1.0	1.0	1.0	0.0	0.0	1.0	0.7
0.0	1.0	0.0	1.0	1.0	1.0	0.7
0.0	0.0	1.0	1.0	1.0	1.0	0.7
1.0	0.0	1.0	1.0	1.0	0.0	0.7
1.0	1.0	1.0	0.0	1.0	0.0	0.7
1.0	1.0	1.0	1.0	0.0	1.0	0.8
1.0	1.0	1.0	0.0	1.0	1.0	0.8
0.0	1.0	1.0	1.0	1.0	1.0	0.8
1.0	0.0	1.0	1.0	1.0	1.0	0.8
1.0	1.0	1.0	1.0	1.0	0.0	0.8
1.0	1.0	0.0	1.0	1.0	1.0	0.8
1.0	1.0	1.0	1.0	1.0	1.0	1.0

Coaliciones Juego B						
A	B	C	D	E	F	v(S)
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.2
0.0	0.0	0.0	0.0	1.0	0.0	0.2
0.0	1.0	0.0	0.0	0.0	0.0	0.2
0.0	0.0	0.0	1.0	0.0	0.0	0.2
0.0	0.0	1.0	0.0	0.0	0.0	0.2
0.0	1.0	0.0	1.0	0.0	0.0	0.4
0.0	0.0	1.0	1.0	0.0	0.0	0.4
0.0	0.0	1.0	0.0	1.0	0.0	0.4
0.0	0.0	1.0	0.0	0.0	1.0	0.4
0.0	1.0	1.0	0.0	0.0	0.0	0.4
0.0	0.0	0.0	1.0	1.0	0.0	0.4
0.0	0.0	0.0	0.0	1.0	1.0	0.4
0.0	1.0	0.0	0.0	0.0	1.0	0.4
0.0	1.0	0.0	0.0	1.0	0.0	0.4
0.0	1.0	1.0	1.0	0.0	0.0	0.6
0.0	1.0	0.0	1.0	1.0	0.0	0.6
0.0	1.0	0.0	1.0	0.0	1.0	0.6
0.0	0.0	1.0	1.0	0.0	1.0	0.6
0.0	0.0	1.0	1.0	1.0	0.0	0.6
0.0	1.0	1.0	0.0	0.0	1.0	0.6
0.0	0.0	1.0	0.0	1.0	1.0	0.6
0.0	0.0	0.0	1.0	1.0	1.0	0.6
0.0	1.0	0.0	0.0	1.0	1.0	0.6
0.0	1.0	1.0	0.0	1.0	0.0	0.6
0.0	1.0	1.0	1.0	0.0	1.0	0.8
0.0	0.0	1.0	1.0	1.0	1.0	0.8
0.0	1.0	0.0	1.0	1.0	1.0	0.8
0.0	1.0	1.0	1.0	1.0	0.0	0.8
0.0	1.0	1.0	0.0	1.0	1.0	0.8
0.0	1.0	1.0	1.0	1.0	1.0	1.0

Coaliciones Juego C						
A	B	C	D	E	F	v(S)
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.2
0.0	0.0	0.0	0.0	1.0	0.0	0.2
0.0	0.0	0.0	1.0	0.0	0.0	0.2
0.0	0.0	1.0	0.0	0.0	0.0	0.2
0.0	0.0	0.0	0.0	1.0	1.0	0.5
0.0	0.0	0.0	1.0	0.0	1.0	0.5
0.0	0.0	0.0	1.0	1.0	0.0	0.5
0.0	0.0	1.0	0.0	0.0	1.0	0.5
0.0	0.0	1.0	0.0	1.0	0.0	0.5
0.0	0.0	1.0	1.0	0.0	0.0	0.5
0.0	0.0	0.0	1.0	1.0	1.0	0.8
0.0	0.0	1.0	0.0	1.0	1.0	0.8
0.0	0.0	1.0	1.0	0.0	1.0	0.8
0.0	0.0	1.0	1.0	1.0	0.0	0.8
0.0	0.0	1.0	1.0	1.0	1.0	1.0

Coaliciones Juego D							
A	B	C	D	E	F	v(S)	
0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.0	1.0	0.3	
0.0	0.0	0.0	0.0	1.0	0.0	0.3	
0.0	0.0	0.0	1.0	0.0	0.0	0.3	
0.0	0.0	0.0	0.0	1.0	1.0	0.7	
0.0	0.0	0.0	1.0	0.0	1.0	0.7	
0.0	0.0	0.0	1.0	1.0	0.0	0.7	
0.0	0.0	0.0	1.0	1.0	1.0	1.0	

Coaliciones Juego E							
A	B	C	D	E	F	v(S)	
0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.0	1.0	0.5	
0.0	0.0	0.0	0.0	1.0	0.0	0.5	
0.0	0.0	0.0	0.0	1.0	1.0	1.0	

Coaliciones Juego F							
A	B	C	D	E	F	v(S)	
0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.0	1.0	1.0	

```
print("-----
```

```
Caso: una aeronave por categoría en configuración Sur (costes en tanto por uno)-----")
```

```
MJ_DSud_frac=juegos_asociados(MDSud_frac,den,den_coa,sec_pista,ctgs_aero)
```

Coaliciones Juego A						
A	B	C	D	E	F	v(S)
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.2
0.0	0.0	0.0	0.0	1.0	0.0	0.2
0.0	0.0	1.0	0.0	0.0	0.0	0.2
0.0	1.0	0.0	0.0	0.0	0.0	0.2
1.0	0.0	0.0	0.0	0.0	0.0	0.2
1.0	0.0	0.0	0.0	0.0	1.0	0.3
1.0	0.0	0.0	0.0	1.0	0.0	0.3
0.0	1.0	1.0	0.0	0.0	0.0	0.3
1.0	0.0	0.0	1.0	0.0	0.0	0.3
0.0	1.0	0.0	0.0	1.0	0.0	0.3
0.0	1.0	0.0	0.0	0.0	1.0	0.3
1.0	0.0	1.0	0.0	0.0	0.0	0.3
0.0	0.0	1.0	1.0	0.0	0.0	0.3
0.0	0.0	1.0	0.0	0.0	1.0	0.3
1.0	1.0	0.0	0.0	0.0	0.0	0.3
0.0	0.0	0.0	1.0	1.0	0.0	0.3
0.0	0.0	0.0	1.0	0.0	1.0	0.3
0.0	0.0	0.0	0.0	1.0	1.0	0.3
1.0	1.0	0.0	0.0	0.0	1.0	0.5
1.0	0.0	1.0	0.0	1.0	0.0	0.5
1.0	0.0	0.0	1.0	0.0	1.0	0.5
1.0	0.0	0.0	0.0	1.0	1.0	0.5
1.0	0.0	1.0	0.0	0.0	1.0	0.5
1.0	1.0	0.0	0.0	1.0	0.0	0.5
0.0	1.0	0.0	1.0	1.0	0.0	0.5
0.0	1.0	1.0	1.0	0.0	0.0	0.5
0.0	1.0	1.0	0.0	1.0	0.0	0.5
0.0	1.0	0.0	1.0	0.0	1.0	0.5
1.0	1.0	1.0	0.0	0.0	0.0	0.5
0.0	0.0	1.0	1.0	1.0	0.0	0.5
0.0	0.0	1.0	1.0	0.0	1.0	0.5
0.0	0.0	1.0	0.0	1.0	1.0	0.5
0.0	0.0	0.0	1.0	1.0	1.0	0.5
1.0	1.0	0.0	1.0	0.0	0.0	0.5
1.0	1.0	0.0	0.0	1.0	1.0	0.7
1.0	1.0	0.0	1.0	0.0	1.0	0.7
1.0	1.0	0.0	1.0	1.0	0.0	0.7
1.0	1.0	1.0	1.0	0.0	0.0	0.7
0.0	1.0	1.0	1.0	0.0	1.0	0.7
1.0	0.0	1.0	1.0	0.0	1.0	0.7
1.0	0.0	0.0	1.0	1.0	1.0	0.7
0.0	1.0	1.0	1.0	1.0	0.0	0.7
1.0	1.0	1.0	0.0	0.0	1.0	0.7
0.0	1.0	0.0	1.0	1.0	1.0	0.7
0.0	0.0	1.0	1.0	1.0	1.0	0.7
1.0	0.0	1.0	1.0	1.0	0.0	0.7
1.0	1.0	1.0	0.0	1.0	0.0	0.7
1.0	1.0	1.0	1.0	0.0	1.0	0.8
1.0	1.0	1.0	0.0	1.0	1.0	0.8
0.0	1.0	1.0	1.0	1.0	1.0	0.8
1.0	0.0	1.0	1.0	1.0	1.0	0.8
1.0	1.0	1.0	1.0	1.0	0.0	0.8
1.0	1.0	0.0	1.0	1.0	1.0	0.8
1.0	1.0	1.0	1.0	1.0	1.0	1.0

Coaliciones Juego B						
A	B	C	D	E	F	v(S)
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.2
0.0	0.0	0.0	0.0	1.0	0.0	0.2
0.0	1.0	0.0	0.0	0.0	0.0	0.2
0.0	0.0	0.0	1.0	0.0	0.0	0.2
0.0	0.0	1.0	0.0	0.0	0.0	0.2
0.0	1.0	0.0	1.0	0.0	0.0	0.4
0.0	0.0	1.0	1.0	0.0	0.0	0.4
0.0	0.0	1.0	0.0	1.0	0.0	0.4
0.0	0.0	1.0	0.0	0.0	1.0	0.4
0.0	1.0	1.0	0.0	0.0	0.0	0.4
0.0	0.0	0.0	1.0	1.0	0.0	0.4
0.0	0.0	0.0	0.0	1.0	1.0	0.4
0.0	1.0	0.0	0.0	0.0	1.0	0.4
0.0	1.0	0.0	0.0	1.0	0.0	0.4
0.0	1.0	1.0	1.0	0.0	0.0	0.6
0.0	1.0	0.0	1.0	1.0	0.0	0.6
0.0	1.0	0.0	1.0	0.0	1.0	0.6
0.0	0.0	1.0	1.0	0.0	1.0	0.6
0.0	0.0	1.0	1.0	1.0	0.0	0.6
0.0	1.0	1.0	0.0	0.0	1.0	0.6
0.0	0.0	1.0	0.0	1.0	1.0	0.6
0.0	0.0	0.0	1.0	1.0	1.0	0.6
0.0	1.0	0.0	0.0	1.0	1.0	0.6
0.0	1.0	1.0	0.0	1.0	0.0	0.6
0.0	1.0	1.0	1.0	0.0	1.0	0.8
0.0	0.0	1.0	1.0	1.0	1.0	0.8
0.0	1.0	0.0	1.0	1.0	1.0	0.8
0.0	1.0	1.0	1.0	1.0	0.0	0.8
0.0	1.0	1.0	0.0	1.0	1.0	0.8
0.0	1.0	1.0	1.0	1.0	1.0	1.0

Coaliciones Juego C						
A	B	C	D	E	F	v(S)
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.2
0.0	0.0	0.0	0.0	1.0	0.0	0.2
0.0	0.0	0.0	1.0	0.0	0.0	0.2
0.0	0.0	1.0	0.0	0.0	0.0	0.2
0.0	0.0	0.0	0.0	1.0	1.0	0.5
0.0	0.0	0.0	1.0	0.0	1.0	0.5
0.0	0.0	0.0	1.0	1.0	0.0	0.5
0.0	0.0	1.0	0.0	0.0	1.0	0.5
0.0	0.0	1.0	0.0	1.0	0.0	0.5
0.0	0.0	1.0	1.0	0.0	0.0	0.5
0.0	0.0	0.0	1.0	1.0	1.0	0.8
0.0	0.0	1.0	0.0	1.0	1.0	0.8
0.0	0.0	1.0	1.0	0.0	1.0	0.8
0.0	0.0	1.0	1.0	1.0	0.0	0.8
0.0	0.0	1.0	1.0	1.0	1.0	1.0

```

+-----+
|           Coaliciones Juego D           |
+-----+-----+-----+-----+-----+
| A | B | C | D | E | F | v(S) |
+-----+-----+-----+-----+-----+
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.3 |
| 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.3 |
| 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.3 |
| 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 0.7 |
| 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.7 |
| 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.7 |
| 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 1.0 |
+-----+-----+-----+-----+-----+
    
```

```

+-----+
|           Coaliciones Juego E           |
+-----+-----+-----+-----+-----+
| A | B | C | D | E | F | v(S) |
+-----+-----+-----+-----+-----+
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.5 |
| 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.5 |
| 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 |
+-----+-----+-----+-----+-----+
    
```

```

+-----+
|           Coaliciones Juego F           |
+-----+-----+-----+-----+-----+
| A | B | C | D | E | F | v(S) |
+-----+-----+-----+-----+-----+
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 |
+-----+-----+-----+-----+-----+
    
```

Valores Marginales

```

print("-----
Caso: una aeronave por categoría en configuración Sur (costes en tanto por
uno)-----")
CJ_DSud_frac=C_marginales_juegos(MJ_DSud_frac,sec_pista,ctgs_aero)
imprimir_matriz(np.transpose(redondeo_M(CJ_DSud_frac,2)),den,"Valores Marg
inales")
    
```

```

-----Caso: una aeronave por categoría en configuración Sur (costes en tanto por uno)-----
+-----+
|           Valores Marginales           |
+-----+-----+-----+-----+-----+
| A | B | C | D | E | F |
+-----+-----+-----+-----+-----+
| A | 0.42 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| B | 0.42 | 0.46 | 0.0 | 0.0 | 0.0 | 0.0 |
| C | 0.42 | 0.46 | 0.5 | 0.0 | 0.0 | 0.0 |
| D | 0.42 | 0.46 | 0.5 | 0.6 | 0.0 | 0.0 |
| E | 0.42 | 0.46 | 0.5 | 0.6 | 0.75 | 0.0 |
| F | 0.42 | 0.46 | 0.5 | 0.6 | 0.75 | 1.0 |
+-----+-----+-----+-----+-----+
    
```

```

print("-----
Caso: una aeronave por categoría en configuración Norte (costes en tanto p
or uno)-----")
CJ_DNud_frac=C_marginales_juegos(MJ_DNud_frac,sec_pista,ctgs_aero)
imprimir_matriz(np.transpose(redondeo_M(CJ_DNud_frac,2)),den,"Valores Marg
inales")

-----Caso: una aeronave por categoría en configuración Norte (costes en tanto por uno)-----
+-----+
|          Valores Marginales          |
+-----+-----+-----+-----+-----+-----+
| A | B | C | D | E | F |
+-----+-----+-----+-----+-----+-----+
| A | 0.42 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| B | 0.42 | 0.46 | 0.0 | 0.0 | 0.0 | 0.0 |
| C | 0.42 | 0.46 | 0.5 | 0.0 | 0.0 | 0.0 |
| D | 0.42 | 0.46 | 0.5 | 0.6 | 0.0 | 0.0 |
| E | 0.42 | 0.46 | 0.5 | 0.6 | 0.75 | 0.0 |
| F | 0.42 | 0.46 | 0.5 | 0.6 | 0.75 | 1.0 |
+-----+-----+-----+-----+-----+-----+

```

Valores Simpliciales

```

print("-----
Caso: una aeronave por categoría en configuración Norte (costes en tanto p
or uno)-----")
Cjugador_DNud_frac=np.arange(ctgs_aero)*0.0
for i in range(0,ctgs_aero):
    Cjugador_DNud_frac[i]=sum(np.transpose(CJ_DNud_frac)[i,:])
imprimir_vector(Cjugador_DNud_frac,den,"Costes según la categoría de avión
")

-----Caso: una aeronave por categoría en configuración Norte (costes en tanto por uno)-----
+-----+
| Costes según la categoría de avión |
+-----+-----+-----+-----+-----+-----+
| A | B | C | D | E | F |
+-----+-----+-----+-----+-----+-----+
| 0.42 | 0.88 | 1.38 | 1.98 | 2.73 | 3.73 |
+-----+-----+-----+-----+-----+-----+

print("-----
Caso: una aeronave por categoría en configuración Sur (costes en tanto por
uno)-----")
Cjugador_DSud_frac=np.arange(ctgs_aero)*0.0
for i in range(0,ctgs_aero):
    Cjugador_DSud_frac[i]=sum(np.transpose(CJ_DSud_frac)[i,:])
imprimir_vector(Cjugador_DSud_frac,den,"Costes según la categoría de avión
")

-----Caso: una aeronave por categoría en configuración Sur (costes en tanto por uno)-----
+-----+
| Costes según la categoría de avión |
+-----+-----+-----+-----+-----+-----+
| A | B | C | D | E | F |
+-----+-----+-----+-----+-----+-----+
| 0.17 | 0.37 | 0.62 | 0.95 | 1.45 | 2.45 |
+-----+-----+-----+-----+-----+-----+

```


ANEXO B: PLANIFICACIÓN

EDT

En la siguiente jerarquía se muestra la descomposición en paquetes de trabajo de este TFM.

A cada paquete se le asignó una cantidad de horas en la planificación que posteriormente han sido comparadas con las reales en la Tabla A-1. En algunos paquetes de trabajo se han tenido desviaciones, alcanzando un total de 15 horas. Las tareas en las que se ha invertido más tiempo, como era previsible, son las relacionadas con el propio desarrollo de la simulación, seguidas de la gestión y documentación (Figura A-1).

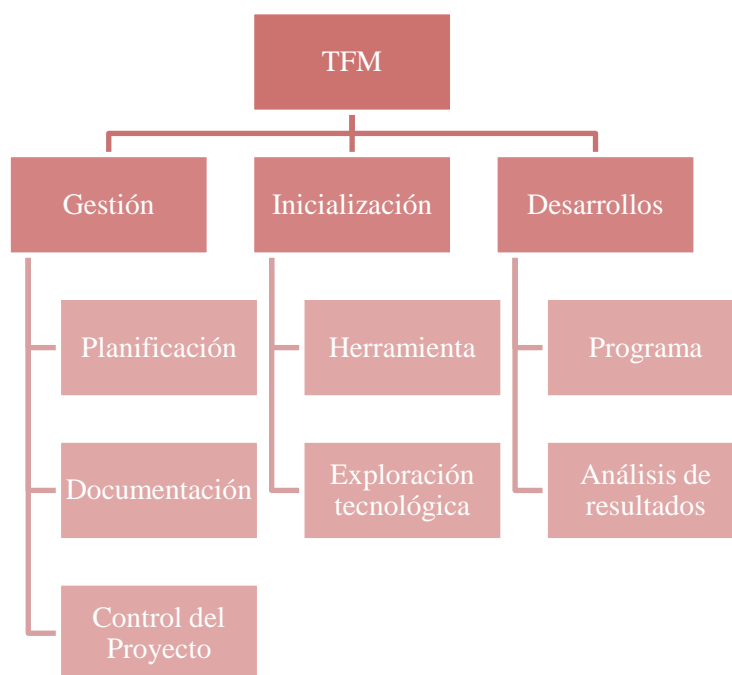


Tabla A-1. Distribución horaria de los paquetes de trabajo de este TFM.

Paquetes de Trabajo	Planificadas	Reales	Desviación
Gestión	110	120	10
Inicialización	50	50	0
Desarrollos	145	150	5
Total	305	320	15

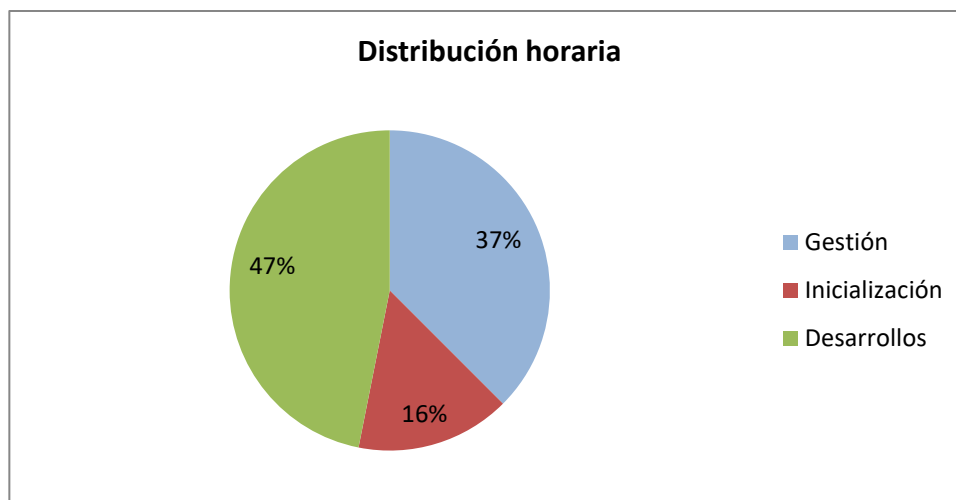


Figura A–1. Distribución porcentual de las horas reales invertidas en los diferentes paquetes de trabajo de este TFM.

Diagrama de Gantt

Seguidamente, se muestra la planificación temporal de todas las actividades del TFM. Se destaca que el trabajo se ha dividido en tres fases relacionadas con el desarrollo del mismo. Desde su inicio, se han empleado 320 h, las cuales se han extendido a lo largo de 89 días.

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	P										
						May '22	11 jul '22	22 ago '22	03 oct '22	14 nov '22						
						S	X	D	J	L	V	M	S	X	D	J
0	TFM	89 días	vie 29/07/22	jue 01/12/22												
1	Gestionar y Controlar Cambios del Proyecto	89 días	vie 29/07/22	jue 01/12/22	3CC;19FF											
2	Primera Fase: Inicialización															
3	Inicio TFM	0 días	vie 29/07/22	vie 29/07/22												
4	Escoger Lenguaje	8 días	jue 18/08/22	lun 29/08/22	3											
5	Realizar Exploración Tecnológica	21 días	mié 03/08/22	mié 31/08/22	3											
6	Segunda Fase: Desarrollos Preliminares															
7	Reunión Juego Inicial	0 días	jue 01/09/22	jue 01/09/22												
8	Desarrollar Juego Inicial	12 días	jue 01/09/22	vie 16/09/22	7											
9	Reunión Evaluación Propuestas	0 días	lun 19/09/22	lun 19/09/22												
10	Programar Índices	25 días	lun 19/09/22	vie 21/10/22	9											
11	Analizar Resultados	15 días	lun 03/10/22	lun 24/10/22	10CC+10 días;12FF											
12	Reunión Evaluación Resultados	0 días	lun 24/10/22	lun 24/10/22												
13	Modificar Programa	5 días	lun 24/10/22	vie 28/10/22	12											
14	Tercera Fase: Documentación															
15	Redactar Documento	15 días	mar 01/11/22	mar 22/11/22	16FF											

Proyecto: TFM
Fecha: jue 24/11/22

Tarea		Resumen inactivo		Tareas externas	
División		Tarea manual		Hito externo	
Hito		solo duración		Fecha límite	
Resumen		Informe de resumen manual		Progreso	
Resumen del proyecto		Resumen manual		Progreso manual	
Tarea inactiva		solo el comienzo			
Hito inactivo		solo fin			

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	P							P						
						May '22			11 jul '22			22 ago '22			03 oct '22			14 nov '22	
						S	X	D	J	L	V	M	S	X	D	J			
16	Entrega Tutores	0 días	mar 22/11/22	mar 22/11/22															
17	Revisar Documento	7 días	mar 22/11/22	jue 01/12/22	15;19FF														
18	Modificar Documento	7 días	mar 22/11/22	jue 01/12/22	15;19FF														
19	Entrega TFM	0 días	jue 01/12/22	jue 01/12/22															

Proyecto: TFM
Fecha: jue 24/11/22

Tarea		Resumen inactivo		Tareas externas	
División		Tarea manual		Hito externo	
Hito		solo duración		Fecha límite	
Resumen		Informe de resumen manual		Progreso	
Resumen del proyecto		Resumen manual		Progreso manual	
Tarea inactiva		solo el comienzo			
Hito inactivo		solo fin			

ANEXO C: PRESUPUESTO

En las siguientes tablas se desglosan todos los costes que ha involucrado este trabajo. Cabe destacar que la única partida relevante del presupuesto es la de personal, puesto que el ordenador empleado ya estaba amortizado y la plataforma de programación utilizada es gratuita.

Se ha decidido incluir esta partida ya que pesar de ser un trabajo de carácter universitario, las horas del mismo se podrían contabilizar para un presupuesto de proyecto de investigación. Ascendiendo el total a la cantidad de 13.010,53 €.

Tabla A-2. Coste de todos los conceptos de este TFM.

Concepto	Coste unitario	Unidad	Cantidad	Coste
Ingeniero	25 €	h	320	8.000 €
Ingeniero jefe proyecto	45 €	h	30	1.350 €
Subtotal personal				9.350,00 €

Tabla A-3. Presupuesto resumen de este TFM.

Concepto subtotales	Coste
Personal	9.350,00 €
Costes indirectos (15 %)	1.402,50 €
IVA (21 %)	2.258,03 €
Total	13.010,53 €

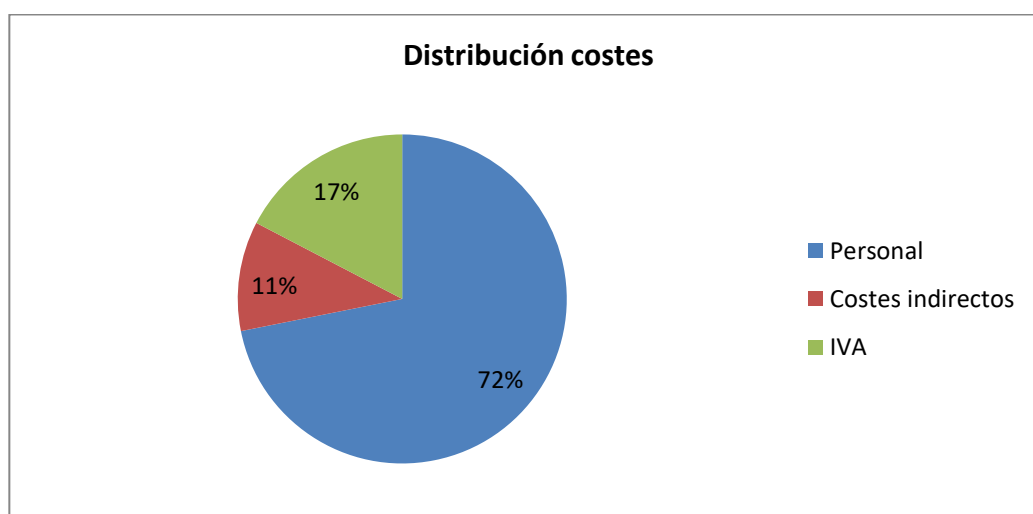


Figura A-2. Distribución porcentual de los costes de este TFM.