



Universidad de Jaén

Escuela Politécnica Superior de Jaén

Asistente para la Selección de Mejores Jugadores en FIFA Ultimate Team

Autor: José Manuel Gamarra Espinar

Grado: Ingeniería informática

Director: Francisco José Quesada Real
Álvaro Labella Romero
Departamento del director: Informática

Fecha: 02/07/2024

Licencia CC



CREEA

Agradecimientos

A lo largo de este camino que ha sido mi trabajo de fin de grado, me gustaría expresar mi profundo agradecimiento a todas las personas que han contribuido y apoyado en este proceso. Sin su ayuda y aliento, este proyecto no habría sido posible.

Quiero comenzar expresando mi más sincero agradecimiento a mis tutores, Francisco José Quesada Real y Álvaro Labella Romero, por su orientación experta, su paciencia y su invaluable apoyo durante todo el desarrollo de este trabajo. Gracias a su conocimiento y dedicación, he podido aprender y crecer profesionalmente al final de la carrera de Ingeniería Informática.

Agradezco también a mis padres y a mi hermana Matilde por su amor incondicional, su constante apoyo y por ser mi fuente de inspiración. Sus palabras de aliento y su confianza en mí han sido fundamentales para enfrentar cada desafío con determinación y perseverancia.

Además, quiero expresar mi gratitud a la Tuna Universitaria de Distrito de Jaén y a mi padrino en la tuna Rafael Romero Cantero. No solo me han enseñado las tradiciones y el espíritu de Jaén, sino que también han sido una parte importante de mi experiencia universitaria, brindándome amistad y momentos inolvidables que han enriquecido mi vida personal y académica.

Finalmente, agradezco a todos aquellos amigos, compañeros y profesionales que de alguna manera contribuyeron con sus consejos, sugerencias y palabras de ánimo a lo largo de este proyecto.

Tabla de contenidos

1. INTRODUCCIÓN	1
1.1. Introducción al Proyecto	1
1.2. Propósito	2
1.3. Objetivos	3
1.4. Metodología de desarrollo	3
1.5. Planificación	4
1.6. Presupuesto	5
1.7. Estructura de la memoria	7
2. ANTECEDENTES	9
2.1. Videojuegos	9
2.2. FIFA	11
2.3. Recuperación de Información Web	14
2.3.1. Web Scrapping	15
2.3.2. XPATH	16
2.4. Toma de Decisión	18
3. DESARROLLO DEL PROYECTO	23
3.1. Descripción del Proyecto	23
3.2. Análisis del sistema	23
3.2.1. Especificación de Requerimientos	24

3.2.2. Modelado del sistema	25
3.3. Diseño del sistema	28
3.3.1. Diseño de los objetos	28
3.3.2. Diseño de la interfaz	31
3.3.3. Diseño de la base de datos	33
3.4. Implementación del sistema	36
3.4.1. Tecnologías usadas	36
3.4.2. Frontend	40
4. CONCLUSIONES	45
A. Manual del Usuario	47
A.1. Guía de Usuario	47
A.1.1. Interfaz	47
A.1.2. Instrucciones detalladas para realizar tareas específicas	47
A.1.3. Ejemplos prácticos	48
B. Despliegue del Sistema	51
B.1. Requisitos del Sistema	51
B.1.1. Hardware necesario	51
B.1.2. Software necesario	51
B.2. Instalación y configuración	51
B.3. Despliegue en producción	52
B.4. Mantenimiento y actualización	52
Bibliografía	II

Lista de figuras

1.1. Logo del videojuego	1
1.2. Diagrama de Gantt Fase 1	4
1.3. Diagrama de Gantt Fase 2	5
1.4. Diagrama de Gantt Fase 3	6
1.5. Diagrama de Gantt Fase 4	6
2.1. Logro del videojuego	10
2.2. Publicidad de una marca en FIFA	12
2.3. Modos de juego en FIFA	12
2.4. Ejemplo de sobres FIFA	13
2.5. Cartas originales de Ultimate Team	13
2.6. Alineación de cartas oro	14
2.7. Estructura del web scraping	15
2.8. Pasos clave en la toma de decisiones	19
3.1. Casos de uso Global	25
3.2. Caso de uso CUU1	26
3.3. Caso de uso CUU2	26
3.4. Caso de uso CUU3	26
3.5. Caso de uso CUU4	27
3.6. Caso de uso CUU5	27
3.7. Caso de uso CUU6	28

3.8. Diagrama de actividades	29
3.9. Diagrama de clases	30
3.10. Diagrama de secuencia filtrar jugador	30
3.11. Diagrama de secuencia TOPSIS	31
3.12. Vista Inicio	32
3.13. Vista Ultimate Team	33
3.14. Vista Alineaciones	33
3.15. Filtrar Jugador	34
3.16. Error Sugerir Jugador	34
3.17. Vista TOPSIS	35
3.18. Vista TOPSIS Realizado	35
3.19. Modelo Entidad-Relación	36
3.20. Hook de estado	37
3.21. Hook de efecto	38
3.22. Elaboración de elementos a obtener en el CSV	39
3.23. Creación Spider	39
3.24. Obtención de raspado	40
A.1. Búsqueda de estadística 1	48
A.2. Búsqueda de estadística 1.1	49
A.3. Búsqueda de estadística 1.2	49
A.4. Realización TOPSIS 1	49
A.5. Realización TOPSIS 1.1	50
A.6. Realización TOPSIS 1.2	50

Lista de tablas

- 1.1. Tabla de estimación de costes 7
- 2.1. Matriz de decisión 21
- 3.1. Tabla de vistas principal 32

Lista de listados de código

- 2.1. Hello World en distintos lenguajes 15
- 2.2. Extrayendo datos 17
- 2.3. Extrayendo datos utilizando Selenium 17
- 2.4. Extrayendo datos usando variables 17

Capítulo 1

INTRODUCCIÓN

1.1. Introducción al Proyecto

Desde hace años, FIFA ha sido uno de los videojuegos deportivos más populares en todo el mundo. FIFA 23 (ver Figura 1.1), vendió más de 10,3 millones de copias en su primera semana, tanto en formato físico como digital. Uno de los principales factores que contribuyen al éxito de FIFA es su modo FIFA Ultimate Team (FUT), que permite a los usuarios crear su propio equipo desde cero utilizando artículos de jugador coleccionables. Con miles de futbolistas disponibles en Ultimate Team, las posibilidades de formar equipos son prácticamente infinitas.



Figura 1.1: Logo del videojuego

Para fichar futbolistas en FUT, los usuarios pueden conseguir monedas de forma gratuita al ganar partidos contra otros usuarios. También pueden adquirir FIFA Points, una moneda virtual que se puede comprar con dinero real para fichar jugadores. Por lo tanto, FUT proporciona un mercado de futbolistas transferibles, que es la base económica de la comunidad de jugadores que apoya este modo competitivo y que es conocido como “tradeo”. Los usuarios pueden comprar y vender futbolistas y otros artículos en el mercado para ganar monedas y obtener nuevos futbolistas para sus equipos.

El mercado de FUT ofrece una vasta cantidad de futbolistas, cada uno con múltiples atributos y distintos precios de mercado. Además, la variedad de combinaciones posibles para formar un equipo ideal es prácticamente infinita. Esta complejidad hace

que la selección de futbolistas que mejor se adapten a una plantilla concreta sea un desafío considerable para los usuarios.

Los futbolistas tienen una variedad de atributos, tales como *velocidad*, *ritmo*, *regate*, entre otros, que determinan su rendimiento en el campo. Sin embargo, una alta puntuación media en estos atributos no garantiza necesariamente que un jugador sea el más adecuado para todas las situaciones de juego. La importancia de ciertos atributos puede variar dependiendo del contexto, lo que complica la tarea de identificar al jugador óptimo para cada equipo y situación específica.

Dado el gran número de futbolistas disponibles en el mercado de FUT, la diversidad de atributos y los precios fluctuantes, los usuarios de FUT enfrentan una tarea abrumadora al intentar identificar los jugadores que mejor se ajustan a sus necesidades específicas. Los métodos manuales para evaluar jugadores pueden ser ineficientes y propensos a errores, ya que no tienen en cuenta la importancia relativa de diferentes atributos en diversas situaciones de juego.

Un asistente web automatizado y basado en algoritmos de toma de decisión multi-criterio puede analizar estos atributos de manera objetiva, proporcionando recomendaciones precisas y personalizadas. Esto no solo ahorra tiempo a los usuarios, sino que también aumenta la eficacia en la toma de decisiones, permitiendo formar equipos más competitivos y optimizados.

1.2. Propósito

Este Trabajo de Fin de Grado (TFG) se centra en el diseño y desarrollo de un asistente web que permite a los usuarios identificar a los mejores jugadores de FUT basándose en diferentes atributos de los jugadores. Esta propuesta surge como respuesta a la creciente demanda de herramientas que faciliten la optimización de equipos en el juego FUT, y a la vez, como una oportunidad para explorar las posibilidades que ofrece la tecnología web en el ámbito de los videojuegos.

El objetivo principal del proyecto es implementar una solución que permita a los usuarios de FUT identificar a los mejores jugadores en función de diferentes atributos de manera rápida y eficiente a través de una interfaz web amigable e intuitiva. Esto permitirá a los usuarios tomar decisiones informadas en el momento de construir sus equipos, optimizando así su rendimiento en el juego.

Para alcanzar este objetivo, se empleará React [12] como biblioteca de desarrollo web y se utilizarán tecnologías de análisis de datos para procesar y mostrar la información relevante de los jugadores. Además, se integrarán herramientas de visualización de datos que facilitarán la comprensión de los atributos de los jugadores y su impacto en el rendimiento del equipo.

Mi motivación para elegir esta temática surge de mi pasión por los videojuegos y mi experiencia personal jugando a FUT, donde a menudo encontré la necesidad de una herramienta que me ayudara a identificar rápidamente a los mejores jugadores

para mis equipos. Esta necesidad, compartida por muchos otros usuarios, me impulsó a desarrollar una plataforma que pudiera llenar este vacío y mejorar la experiencia de juego para la comunidad. Se llevará a cabo una evaluación del asistente web en términos de rendimiento y experiencia de usuario, reflexionando sobre el proceso de desarrollo y los resultados obtenidos. Además, se identificarán posibles limitaciones y se propondrán líneas de investigación futuras para continuar mejorando la herramienta.

1.3. Objetivos

Los objetivos de este TFG se enumeran a continuación:

- Revisar los conceptos relacionados con la toma de decisiones inteligentes para seleccionar jugadores y con FUT.
- Identificar los atributos de los jugadores en FUT y analizar su importancia en el juego.
- Extraer información de los jugadores incluidos en FUT.
- Diseñar e implementar un asistente que permita definir y resolver la selección de jugadores adecuados.
- Analizar los resultados obtenidos.
- Redactar una memoria que recoja todo el trabajo desarrollado.

1.4. Metodología de desarrollo

Para el desarrollo de este proyecto software, se ha seguido un enfoque de metodología ágil [3]. La metodología ágil se basa en una serie de técnicas aplicadas a ciclos de trabajo cortos, otorgando una mayor optimización del flujo de trabajo y una mayor satisfacción del cliente.

Para este proyecto se ha seleccionado concretamente la metodología ágil SCRUM, basada en iteraciones cortas, donde se entregan pequeñas partes o incrementos del producto. El proceso consiste en plantear una idea, diseñarla, implementarla y, finalmente, probarla. De esta manera, cada iteración presenta versiones del software visibles y ejecutables [22].

Estas iteraciones son llamadas sprints y suelen durar entre dos o cuatro semanas. En este proyecto en concreto, se han llevado a cabo reuniones con los tutores cada semana o cada dos semanas en las que se exponían los avances y dudas en las tareas y se planifica el siguiente sprint.

1.5. Planificación

La planificación del TFG se describe mediante varios sprints recogidos en distintas fases del proyecto. Para ello, se utilizará un diagrama de Gantt, una herramienta visual que representa el avance de un proyecto a lo largo del tiempo, muestra las tareas programadas, su duración y secuencia, así como los recursos asignados a cada una de ellas.

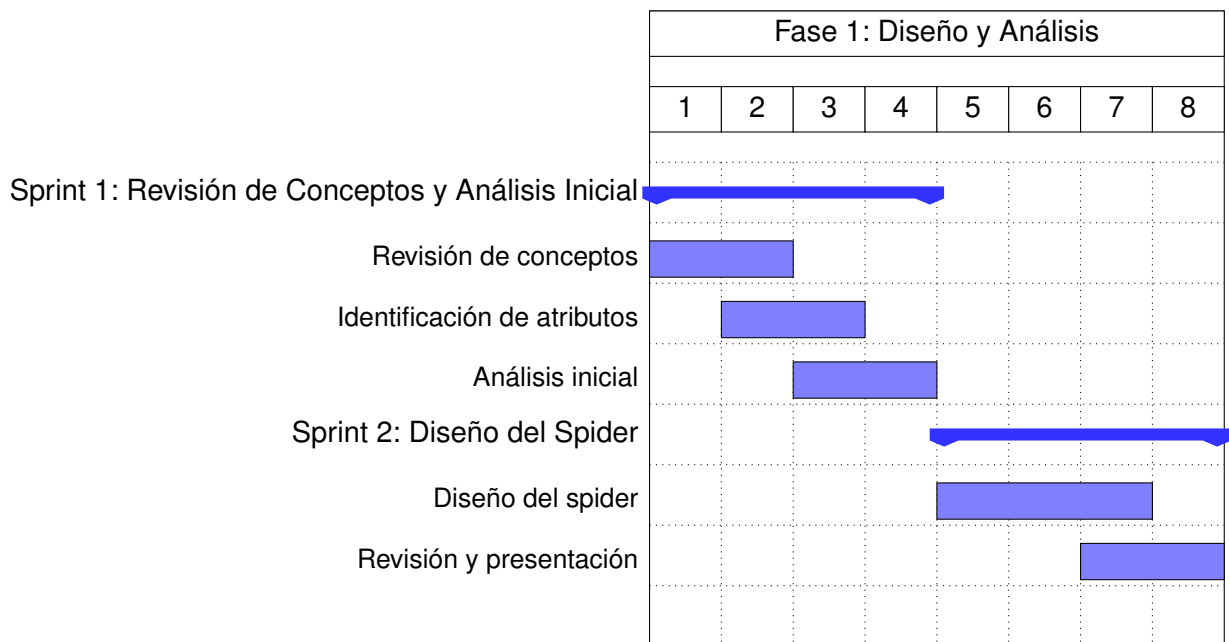


Figura 1.2: Diagrama de Gantt Fase 1

Esta primera fase (ver Figura 1.2), se desarrolla en 2 sprints, uno centrado en la revisión de conceptos y análisis inicial en las que se observan los conceptos relacionados con la toma de decisiones inteligentes y los conceptos relacionados con FUT, documentando los atributos clave.

En el segundo sprint, se diseñará el spider con el que logrará hacer la recuperación de datos, así como su revisión y presentación.

La segunda fase (ver Figura 1.3), se centra en la extracción de información y desarrollo inicial, el objetivo del sprint 3 es extraer información de los jugadores incluidos en FUT. Para ello, se realizará el web scraping con el spider diseñado anteriormente, verificando que todos los datos se vean correctos e íntegros. Posteriormente, en el sprint 4 se realizará la implementación de la interfaz de la página web, así como las pruebas para comprobar que no haya errores en el desarrollo en React.

En la tercera fase (ver Figura 1.4), se continúa desarrollando el asistente implementando el método de toma de decisión multi-criterio, haciendo correcciones y mejoras. En el sprint de optimización se mejora los resultados del modelo, así como la página web, incorporando una mejor usabilidad y diseño.

Finalmente, en la fase 4 (ver Figura 1.5), se evalúan los resultados obtenidos y se

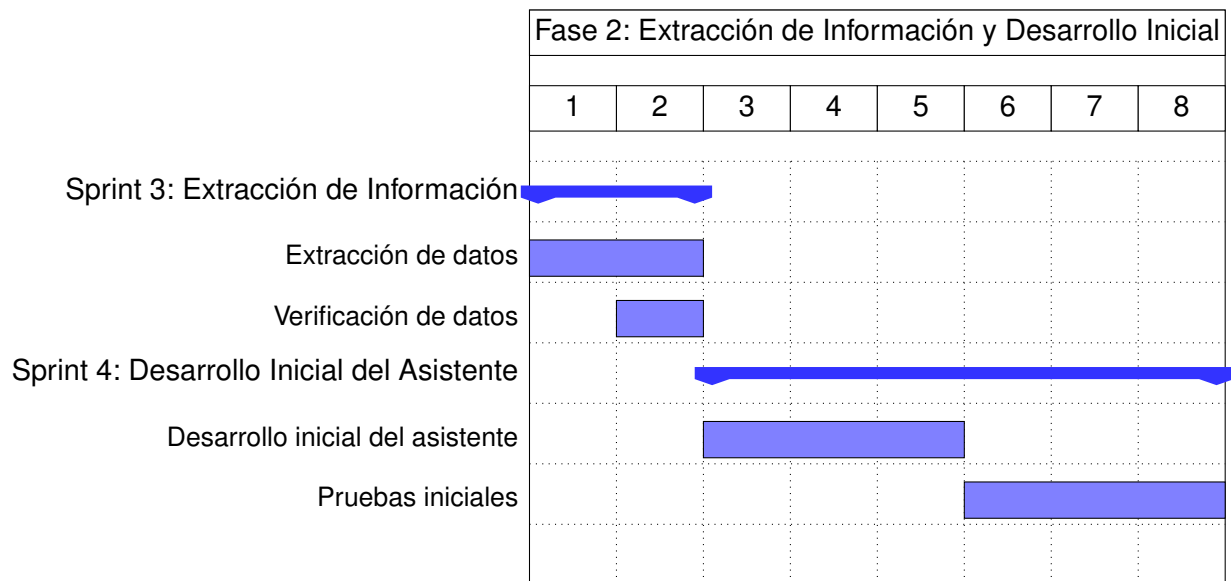


Figura 1.3: Diagrama de Gantt Fase 2

documentan, con el fin de poder redactar la memoria. La revisión y presentación final representan la última semana del TFG.

1.6. Presupuesto

El objetivo de esta sección es representar de una forma lo más aproximada posible el coste del proyecto. Para realizar el presupuesto se tienen en cuenta los gastos relativos al hardware y al software, así como los roles de las personas que deberían ser contratadas para el desarrollo del proyecto.

En la parte del software, se han utilizado varias librerías y aplicaciones gratuitas, tales como Visual Studio Code, Meet para realizar reuniones o la librería TOPSIS entre otras. Sin embargo, se ha utilizado una versión de Windows 10 Pro, con un coste aproximado de 60€.

Por otro lado, con relación al hardware, el desarrollo se ha llevado a cabo en un ordenador con las siguientes características:

- **Portátil:** HP Pavilion
- **Procesador:** Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz (hasta 1.99 GHz)
- **RAM instalada:** 16,0 GB
- **Tipo de sistema:** Sistema operativo de 64 bits, procesador basado en x64
- **Adaptador gráfico:** Nvidia geforce mx250
- **Pantalla:** 1920 x 1080 pixels

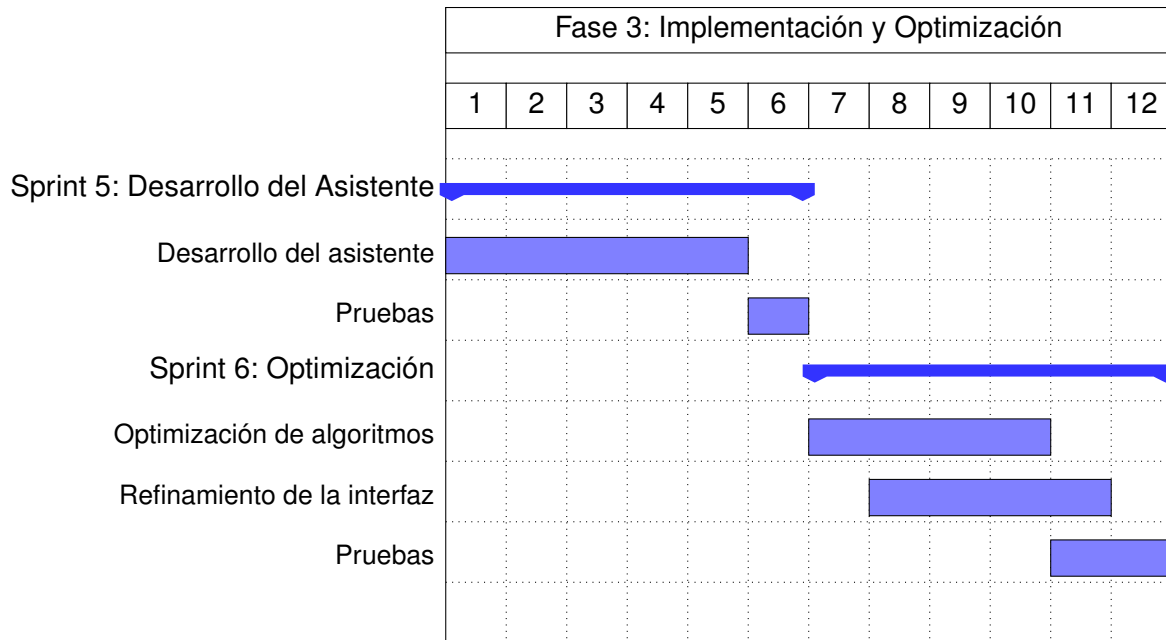


Figura 1.4: Diagrama de Gantt Fase 3

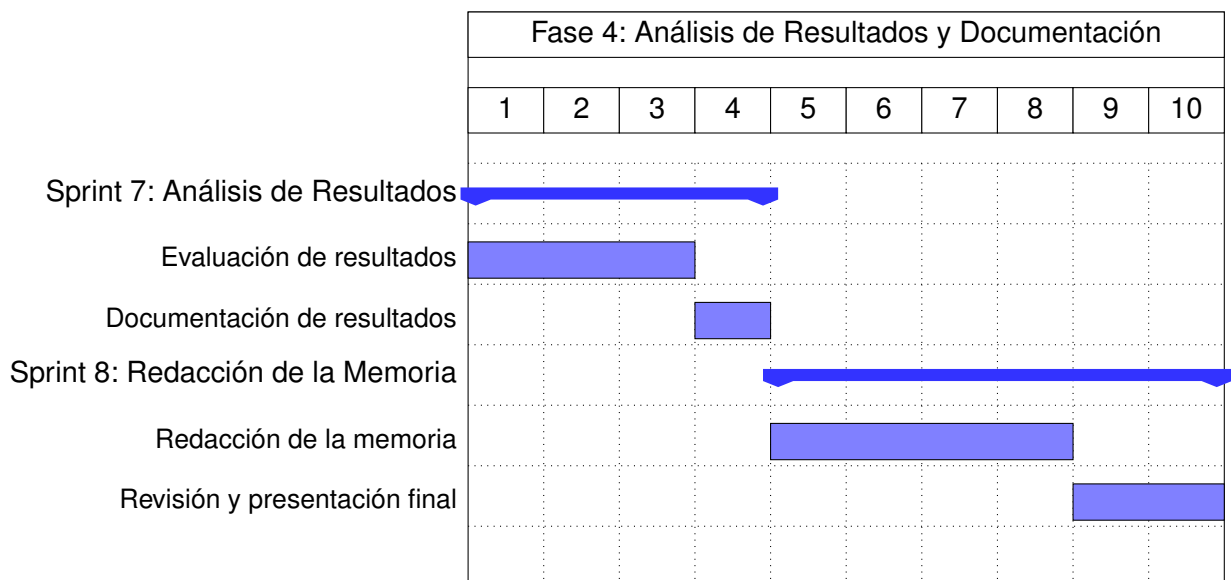


Figura 1.5: Diagrama de Gantt Fase 4

Como el precio del portátil costó 900 euros, se presupone un tiempo de amortización de 5 años, la parte proporcional del tiempo del proyecto sería un precio total de 180€.

Con una aproximación de 9 meses de proyecto y a media jornada, se presupone unas 760 horas totales. Aunque este TFG ha sido desarrollado por una persona nada más, se han considerado los siguientes roles:

- Jefe de proyecto: Con un 10 por ciento de las horas totales, lo que equivaldría a 76 horas.
- Programador junior: 90 por ciento de las horas totales, 684 horas de trabajo.

El salario jefe de proyecto promedio en España es de 38.000€ al año o 19,49€ por hora. Esto puede variar según el nivel de experiencia. Por lo tanto, el sueldo total del jefe sería de 1.481,24€. Mientras que para un programador junior promedio en España es de 21.000€ al año o 10,77€ por hora. Concluyendo en un salario total para el programador junior de 7.366,68€.

En la Tabla 1.1 se puede observar un resumen de la estimación de costes del proyecto, el cual tendría un costo total de 9.087,92€.

Sección	Coste
Software	60€
Hardware	180€
Personal	8.847,92€
Total	9.087,92€

Tabla. 1.1: Tabla de estimación de costes

1.7. Estructura de la memoria

Una vez introducido el proyecto en el Capítulo 1, a continuación, se realizará una breve descripción de los siguientes capítulos que describen el desarrollo del proyecto.

En el Capítulo 2 se realiza la introducción y análisis de los conceptos aplicados del TFG, abordando temas como el concepto de gamificación, la influencia de la psicología en los videojuegos, recuperación de información web y estrategias utilizadas.

El Capítulo 3 se centrará en el análisis y diseño del sistema, así como su implementación.

Finalmente, el Capítulo 4, abordará una reflexión sobre el desarrollo y resultados obtenidos, discutiendo las limitaciones y planteando líneas de exploración para estudios futuros, a su vez finalizando en la presentación de las conclusiones.

Por otra parte, el Apéndice A incluye un manual de usuario, explicando cómo utilizar la aplicación y arrancar el proyecto. El Apéndice B describe cómo desplegar del

sistema mediante un manual diseñado para guiar a los administradores del sistema o desarrolladores en el proceso de configurar, instalar y poner en funcionamiento la herramienta en un entorno de producción.

Capítulo 2

ANTECEDENTES

Este capítulo se centra en introducir y analizar los conceptos aplicados en la elaboración de este TFG. En primer lugar, se abordará el concepto de gamificación y se examinará la influencia de la psicología en los videojuegos, profundizando en los aspectos cognitivos y en los comportamientos de los jugadores.

Seguidamente, se introducirán algunas nociones sobre la recuperación de información web, examinando cómo los usuarios acceden, buscan y recuperan información. Se describirán las técnicas de recuperación de información utilizadas como web scraping y Xpath.

Por último, se describirán algunos de los conceptos básicos relacionados con la toma de decisiones, centrándonos en su definición, esquema de resolución, modelos de decisión y toma de decisiones multi-criterio.

En conjunto, este capítulo de antecedentes proporcionará una base sólida para comprender de forma general los conceptos básicos en los que se sustenta el TFG.

2.1. Videojuegos

Los videojuegos han dejado de ser simplemente una forma de entretenimiento para convertirse en una parte integral de la cultura contemporánea. Con su creciente popularidad y avances tecnológicos, los videojuegos han logrado cautivar a millones de personas en todo el mundo, transportándolas a mundos virtuales llenos de emociones, desafíos y experiencias interactivas [9].

La gamificación es un término implícito en los videojuegos, basado en el enfoque de usar el videojuego para alcanzar el aprendizaje de ciertos temas por parte de los jugadores, dicho de otro modo, se basa en el uso de estrategias y mecánicas para comunicar un mensaje o cambiar la actitud del jugador a través de algo placentero que plantee motivación e implicación. De acuerdo a lo que expone Espinosa en [7], *“gamificar un proceso es la respuesta a una necesidad donde se busca trabajar unos*

contenidos educativos proporcionando experiencias. De manera general, se define una necesidad, una dinámica, mecánicas, etc. pero esto también puede variar.”.

La gamificación debe estar centrada en la creación de experiencias de juego que tengan un valor real. Por otro lado, el juego en cuestión tiene que tener carácter voluntario y una motivación, sin intentar dirigir la toma de decisiones del usuario, reduciendo así su voluntariedad, y alejándose del objetivo de la experiencia [7].

A menudo, dicha motivación se logra mediante la consecución de recompensas o insignias por realizar una determinada acción dentro del videojuego. Un ejemplo pueden ser los logros de PlayStation, que recompensan al jugador cuando cumple un objetivo dentro del videojuego. Estos logros influyen en la motivación, proporcionando avances satisfactorios al jugador. Estos elementos deben estar diseñados para ser atractivos, de manera que sigan cautivando al jugador. Como se puede ver en la Figura 2.1, el jugador logra batir a un jefe de un nivel y gana por esto un logro en el videojuego que aparece en la parte superior izquierda de la imagen [17].



Figura 2.1: Logro del videojuego

Sin duda, el interés y la motivación y, en consecuencia, los logros, serán diferentes para cada tipo de videojuego. En el caso de los videojuegos deportivos, la motivación puede estar más relacionada, por ejemplo, con la creación del mejor equipo o en la mejora de los atributos de los deportistas para que tenga un mejor desempeño.

A lo largo de los últimos años también se ha observado los beneficios de jugar online, entre los que destacan la oportunidad de compartir experiencias y su función social. De acuerdo con Martínez, *“Los videojuegos online multijugador, al ser ambientes lúdicos incorporados a redes de internet, se han convertido en un medio social alternativo a las redes sociales. Los jugadores hacen uso de estos no solo por cuestiones de ocio, sino como medio de socialización. Gracias a ellos, mantienen sus relaciones interpersonales. Esto se puede evidenciar en los reportes anuales de la Entertainment Software Association de 2015 y 2017. Un porcentaje mayor al 53 por ciento de jugadores suelen jugar con alguien más. Incluso, entre ambos informes se contabiliza un promedio de 53.5 por ciento de jugadores que señalan que jugar estos videojuegos les ayuda*

a estar en mayor contacto con sus amistades. Así mismo, un promedio de 43.5 por ciento de estos jugadores manifestaron que hacer uso de estos videojuegos les ayudó a pasar más tiempo en familia [5]".

Otro aspecto importante a considerar dentro del mundo de los videojuegos es la psicología. Diversos estudios realizados han dado como conclusión que jugar a videojuegos mejora la satisfacción y comunicación familiar. Además, aportan una mejora en la toma de decisiones, sus reflexiones y colaboraciones y también en el aprendizaje perceptivo, que permite al jugador mejorar destrezas individuales a lo largo del tiempo y que le permite adaptarse a distintos desafíos [23]. También presentan ciertos inconvenientes, poniendo énfasis en los menores de edad, donde la limitación de tiempo de pantalla es clave. Se recomienda un equilibrio de días y horas de uso para cuidar la salud cognitiva del usuario.

2.2. FIFA

Entre los videojuegos deportivos, destaca FIFA, una serie de videojuegos de fútbol desarrollada por EA Sports. El nombre del videojuego tiene su origen en el nombre de la Federación Internacional de Fútbol Asociación. FIFA ha logrado establecerse como un referente en el género, brindando a los jugadores la oportunidad de experimentar la emoción del fútbol desde la comodidad de sus hogares [16].

FIFA ha evolucionado a lo largo de los años, presentando mejoras gráficas, jugabilidad realista y una amplia gama de características y modos de juego. Los usuarios pueden asumir el papel de sus equipos y jugadores favoritos, participar en torneos, gestionar clubes, enfrentarse a desafíos en línea y mucho más.

Una de las características más destacadas de FIFA es su enfoque en la simulación realista del fútbol profesional. Los jugadores pueden disfrutar de la emoción de competir en estadios emblemáticos, usar uniformes oficiales, escuchar comentarios realistas y disfrutar de una experiencia visualmente impresionante.

Además, las marcas han visto una buena oportunidad de hacer promoción de sus productos a través de carteles dentro de los partidos en este entorno digital, como se puede observar en la Figura 2.2, tal como sucede en el mundo real [27].

FIFA, que posee varios modos de juego tales como: partido, modo carrera o temporadas, tal como se observa en la Figura 2.3, sin embargo, el que ha causado mayor impacto en la sociedad ha sido FUT. En este modo de juego, el usuario forma un equipo de futbolistas ganando a otros usuarios, mediante recompensas, jugando desafíos o comprando sobres virtuales. Estos sobres virtuales son los denominados "loot boxes". Los loot boxes son cajas de recompensas que compras para obtener un artículo del juego. Simulan los tradicionales sobres de cromos en los que se conoce el número de cartas que hay dentro, pero no se sabe cuáles. De hecho, las cartas mejor valoradas son las que menos probabilidad tienen de salir, por lo que la apertura de loot boxes no siempre tiene un resultado satisfactorio.



Figura 2.2: Publicidad de una marca en FIFA



Figura 2.3: Modos de juego en FIFA

En FUT, se pueden crear equipos gracias a la obtención de jugadores mediante sobres como los que se pueden observar en la Figura 2.4 o retos. Estas monedas se obtienen jugando partidos o comprándolas con dinero real. Los jugadores se representan en cartas y estas tienen atributos que determinan como de bueno es el jugador. Hay varios tipos de carta, algunas especiales, como jugador de la semana o jugador del año, pero las básicas son bronce, plata y oro, siendo la carta oro la mejor de las tres.

Tal y como vemos en la Figura 2.5, estas cartas tienen distintos atributos: ritmo (PAC), regate (DRI), tiro (SHO), defensa (DEF), pase (PAS) y físico (PHY). El número que aparece en la esquina superior izquierda es la valoración media del futbolista. Todas estas cartas contienen también una foto del jugador, su club, su nacionalidad y su posición. Gracias a estas cartas podremos formar alineaciones como la de la figura 2.6. Cuanto más alta sea la valoración de las cartas, más competitivo será nuestro



Figura 2.4: Ejemplo de sobres FIFA



Figura 2.5: Cartas originales de Ultimate Team

equipo y, a priori, más fácil será ganar los partidos.

También es importante destacar que, para tener más posibilidades de victoria, se tiene que realizar un equipo con *buena química*, es decir, que los futbolistas se complementen bien entre ellos. Tener el máximo de química, permitirá que se incremente el valor de algunos atributos de los futbolistas, de manera que el desempeño general del equipo también aumentará cuantos más jugadores consigan ese incremento adicional de química. El nivel de química se representa en los pequeños rombos que aparecen debajo de la carta en la alineación, como se observa en la Figura 2.6. Los aspectos que se consideran para aumentar la química del equipo son tener varios jugadores de un mismo club o tener jugadores de un mismo país o liga. Cuantos más jugadores en tu equipo cumplan estas características, mayor será su nivel de química y, por lo tanto, mejor su rendimiento.



Figura 2.6: Alineación de cartas oro

2.3. Recuperación de Información Web

La recuperación de información web es el proceso de búsqueda, recuperación y presentación de información útil a partir de fuentes de datos en la web. Como la web contiene una cantidad masiva de información, el objetivo de la recuperación de información web es ayudar al usuario a encontrar la información deseada de manera eficiente, mediante la aplicación de distintas técnicas o algoritmos [25].

Ejemplos de sistemas de recuperación de información web serían los motores de búsqueda como Bing o Google, que indexan el contenido de las páginas y proporcionan resultados relevantes de búsqueda cuando el usuario ingresa su consulta. Aspectos como la relevancia a la hora de dar resultados, la clasificación de la información y su personalización son consideraciones bastante importantes en la recuperación de información web.

A lo largo de esta sección se explicarán diversas técnicas y herramientas de recuperación de información como Web Scraping o Xpath.

2.3.1. Web Scrapping

La recopilación automatizada de datos de Internet es una práctica tan antigua como Internet. El término *web scraping*, también llamado recolección web, es definido como el proceso automatizado de extracción de datos e información desde páginas web de manera sistemática.

En lugar de utilizar una API para obtener datos estructurados, el *web scraping* implica el uso de programas diseñados para navegar por múltiples páginas web, descargar su contenido y analizarlo en busca de información específica. Normalmente, se consigue un programa que consulte un servidor web, solicite datos, analice esos datos y extraiga la información necesaria [19]. La estructura básica del web scraping es la vista en la Figura 2.7.

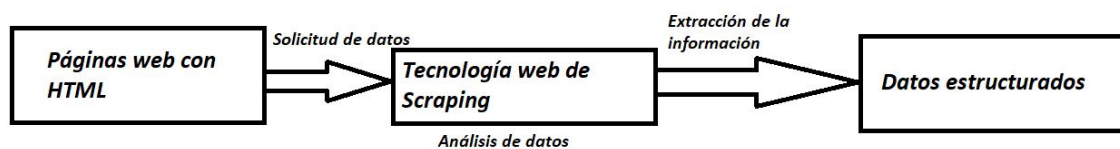


Figura 2.7: Estructura del web scraping

“El web scraping es una de las herramientas más valiosas disponibles para los científicos de datos, ya que permite la extracción de una gran cantidad de datos que se generan constantemente en línea con un costo relativamente bajo ” [28].

Python es el lenguaje por excelencia para hacer scraping, ya que no requiere de gran experiencia para usarlo. Además, tiene una gran comunidad, lo que facilita la resolución de fallos ante posibles problemas que surjan. Por último, su accesibilidad es otro punto clave para que sea el lenguaje más popular para realizar web scraping [13]. Como podemos observar en el Listado 2.1 el lenguaje Python es sencillo e intuitivo para el usuario y no requiere de un alto grado de conocimiento en programación.

```

1 //Java
2 class TextApp{
3 public static void main (String[] args){
4 System.out.println("Hello World");
5 }
6 }
7 //C++
8 #include <iostream>
9 int main(){
10 std::cout<<"Hello World";
11 }
12 //Python
13 Print "Hello World";
  
```

Listado 2.1: Hello World en distintos lenguajes

Es importante destacar que pueden existir varios enfoques para web scrapping. Algunos de los más interesantes son [13]:

1. **Expresiones regulares:** Las expresiones regulares funcionan marcando un modelo que queremos encontrar en una cadena y luego busca cualquier similitud de esa cadena de texto
2. **Beautiful Soup:** Es un paquete de Python que deja recuperar datos estructurados de una página web. Sobre todo es usado para analizar XML y HTML. En comparación con las expresiones regulares, son mucho más fáciles de usar, ya que tiene menos pasos para navegar, examinar y actualizar un árbol de análisis. Otra característica importante es la capacidad de convertir de forma automática el documento saliente en UTF-8 y el entrante en Unicode, por lo que no necesita realizar monitoreo de las codificaciones.
3. **Lxml:** Es una biblioteca de Python eficiente y rápida en funciones para XML y procesamiento HTML. Se puede extraer el contenido en una lista usando XPath.

Aunque ha habido muchos avances en web scraping, su legalidad sigue siendo algo discutible. La prohibición de scraping no puede impedir que alguien pueda rastrear un sitio web desde un punto de vista legal. Sin embargo, su uso fraudulento está penado por varias leyes, por ejemplo, un usuario que acceda a datos de la web a sabiendas de que son confidenciales puede ser penado bajo la ley de abuso y fraude informático. Si el usuario utilizando web scraping sobrecarga o daña el servidor web, entonces esta persona puede ser acusada por la ley [14].

2.3.2. XPATH

XPath es un lenguaje para hallar de forma única partes de un documento XML. Es utilizado en el ámbito de la extracción de datos web, particularmente en herramientas como Scrapy [18].

XPath permite navegar a través de la estructura jerárquica de un documento XML o HTML mediante la utilización de rutas de acceso. Estas rutas de acceso, también conocidas como expresiones XPath, describen la ubicación de los elementos o nodos dentro del documento.

Los conceptos básicos del XPath incluyen:

1. **Nodos:** El documento XML o HTML se organiza en una estructura jerárquica compuesta por nodos. Los nodos pueden ser elementos, atributos, texto u otros tipos de información presente en el documento.
2. **Expresiones de ruta:** Las expresiones XPath se utilizan para definir una ruta de acceso a un nodo o conjunto de nodos en el documento. Estas expresiones pueden ser absolutas, comenzando desde la raíz del documento, o relativas, comenzando desde un nodo de referencia.
3. **Selección de nodos:** Con XPath, es posible seleccionar nodos utilizando diferentes criterios, como el nombre del elemento, el contenido del texto, los

atributos o su posición relativa en la estructura del documento. Esto permite una gran flexibilidad en la extracción de información específica.

4. Ejes: Los ejes en XPath permiten especificar la dirección de navegación y el tipo de relación entre los nodos. Algunos ejes comunes incluyen el eje descendente (child), el eje padre (parent), el eje siguiente (following), entre otros.
5. Predicados: Los predicados se utilizan para filtrar los nodos seleccionados según ciertas condiciones. Estas condiciones pueden incluir comparaciones, evaluaciones lógicas y otras expresiones (ver Listado 2.2).

```
1 sel = Selector(response)
2 response.xpath('//span[@id="ps-lowest-1"]/text()').get()
```

Listado 2.2: Extrayendo datos

En el Listado 2.2 podemos observar como se utiliza el comando Selector, que es un mecanismo que se utiliza para seleccionar partes del documento HTML especificándolo con XPath.

Normalmente, no es necesario hacer manualmente el selector de Scrapy, el objeto *response* está disponible en el Spider, de tal forma que si se usa este acceso directo podemos afianzar que el cuerpo de la respuesta se analice una vez solamente.

Este código busca un elemento HTML ** con el atributo *id* igual a *ps-lowest-1* en el contenido HTML de la página web. Luego, extrae el texto contenido dentro de ese elemento y lo devuelve. *.get()* se utiliza para obtener el primer valor de una lista de elementos extraídos.

```
1 data_price = response.meta['driver'].find_element_by_xpath('//span[@id="ps-lowest-1"]').text
```

Listado 2.3: Extrayendo datos utilizando Selenium

Sin embargo, también se puede utilizar el XPath con la biblioteca Selenium, En el Listado 2.3 se utiliza Selenium para extraer el texto de un elemento span, guardando su valor en una variable llamada *dataprince*. El *response.meta[driver]* hace referencia al controlador del navegador web utilizado por Selenium para interactuar con la página web, el *find element* es otro método de Selenium que busca un elemento en la página web utilizando una expresión XPath. En este caso se extrae el texto de un elemento span con el atributo *id* igual a *ps-lowest-1*.

```
1 class Campos(Item):
2     price = Field ()
3 item = ItemLoader(item=Campos(), response=response)
4 item.add_xpath('price', '//span[@id="pc-lowest-1"]/text()')
```

Listado 2.4: Extrayendo datos usando variables

Se puede interactuar de diversas formas, como en el Listado 2.4. Se crea una clase llamada *Campos* donde se irán guardando los campos que serán las estadísticas de cada jugador.

Esta clase campos hereda de Item, que es una clase base de Scrapy utilizada para definir campos de un elemento que se extrae en una página web. Posteriormente, se crea una instancia de ItemLoader llamada item pasando como argumento una instancia de Campos() y como respuesta response, obtenida de la página web. Finalmente, se añade con XPath al campo price del item el texto del elemento span con id pc-lowest-1.

2.4. Toma de Decisión

La toma de decisión es algo cotidiano en seres humanos. Abarca decisiones fáciles o difíciles, como qué ropa ponerse hoy o qué puedo comer. La toma de decisiones es un método que recaba información y evalúa las posibles alternativas para concluir en una decisión final.

La toma de decisiones es un proceso a veces complejo que hay que llevar a cabo de forma correcta, evitando problemas como la disonancia cognitiva, es decir, cuando lo que se quiere hacer y lo que se realiza no coincide, o el pensamiento en grupo y el consenso, es decir, cuando un grupo de personas decide por otros individuos, a pesar de que exista desacuerdo [21].

Para entender mejor la toma de decisiones, debemos entender los pasos clave del proceso [15]:

1. **Identificación del problema y decisión:** Para saber qué decisión tomar, es necesario saber a qué tipo de problema se enfrenta el individuo y cuáles son sus objetivos. Una vez definidas estas premisas, se puede abordar mejores decisiones.
2. **Recolección de información:** Para tomar la decisión, es fundamental reunir información relacionada con el problema, esto nos permite enfocarlo de una manera más eficiente, ya sea buscando en fuentes internas o externas, dando así una variedad de alternativas de soluciones.
3. **Valoración de soluciones distintas:** En un problema pueden surgir varios inconvenientes, valorando situaciones distintas alcanzaremos diferentes soluciones que nos permitirán tener mayor flexibilidad a la hora de abordar el problema.
4. **Análisis de soluciones:** Este paso consiste en analizar todas las soluciones distintas halladas y observarlas desde un punto de vista de pros y contras. Esto nos permite eliminar las alternativas que no son tan eficientes.
5. **Elección de la decisión:** Una vez analizadas todas las alternativas, se escoge una como solución del problema, que debería ser la más óptima. A veces no tiene por qué ser una solución única, también puede darse la hibridación de estas para obtener un mejor resultado.
6. **Implementación de la decisión:** Cuando se elige la decisión, toca el turno de su implementación alineándose con los objetivos preestablecidos.

7. Análisis e impacto de la decisión tomada: Finalmente se evalúa el impacto de la decisión, si se resolvió el problema planteado, si se obtuvo beneficio o fue una decisión desacertada. Si el resultado no fuese satisfactorio, se pueden aplicar técnicas de gestión como ayuda.



Figura 2.8: Pasos clave en la toma de decisiones

Existen varios tipos de modelos de toma de decisiones, siendo los racionales los que se rigen por la experiencia e instinto de decisores y los creativos cuyo análisis de la decisión no se identifica por las ventajas y desventajas, sino que se sigue un proceso iterativo con el que se prueba distintas soluciones y se observa cuál se adapta mejor al problema dado.

Sin embargo, uno se puede preguntar que decisión es mejor basándonos en unos criterios, por ejemplo, si uno quiere saber que ropa ponerse y es verano, un criterio podría ser el color de la ropa o el tipo de corte. A la decisión basándonos en valoraciones de criterios distintos, se le llama decisión multi-criterio [1]. La decisión multi-criterio es una optimización con varias funciones objetivo y un agente que decide:

$$\max F(x), x \in X \quad (2.1)$$

La Ecuación 2.1 describe un proceso de toma de decisión multi-criteria de manera que:

- x : Es el vector de variables de decisión. Hay que asignar los mejores valores.
- X : Es la región factible del problema, es decir, el conjunto de posibles valores que pueden tomar las variables

- $F(x)$: Es el vector de las funciones objetivo que reúnen los criterios del problema. Dado que pueden existir criterios de beneficio (cuanto más mejor) o coste (cuanto más peor), la maximización de las funciones no es restrictiva, por lo que se puede adaptar dependiendo del tipo de criterio mediante un cambio de signo.

En este TFG es fundamental usar un modelo que permita resolver problemas multi-criterio, puesto que consideramos la elección de un jugador con diferentes atributos (ritmo, físico,...) a los que se les podrán asignar distintos pesos dependiendo del objetivo del usuario. A la hora de seleccionar los criterios en este tipo de problemas, se deben fijar en las siguientes propiedades [11]:

- Exhaustividad: Asegurarse que no se olvida ningún criterio que permita discernir alternativas. Es la más importante porque muestra todos los aspectos relevantes al problema.
- Coherencia: El usuario que toma la decisión tiene unas preferencias globales que deben tener sentido con las preferencias de cada criterio
- No redundante: El conjunto de criterios seleccionado no debe ser sobrante e inútil, ya que no aportaría nada al problema. Esta propiedad no es igual a independencia de criterios.

Para evaluar cada criterio, se puede usar distintas escalas de medida. Estas pueden ser cualitativas, es decir, se evalúa en torno a calificaciones tipo “malo”, “regular”, “bueno” o pueden ser cuantitativas que se rigen por números.

En los problemas multi-criterio, a menudo los diferentes criterios tienen diferente importancia en la resolución del problema. Por ejemplo, si nos centramos en nuestro problema, podríamos querer seleccionar un jugador para ocupar una posición de ataque en nuestro equipo. En ese caso, atributos como la defensa, o la habilidad como portero, no son tan importantes como otros como la velocidad, la agilidad o la estatura. Existen muchos enfoques para asignar pesos a los criterios, tanto objetivos como subjetivos. En este TFG, la ponderación de los criterios seguirá un enfoque subjetivo, donde será el usuario el que decida la ponderación de cada uno de los criterios.

Uno de los métodos multi-criterio existentes más usados en la literatura es el Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), propuesto por Hwang y Yoon en 1981. Este método ayuda a seleccionar la mejor opción entre un conjunto de alternativas, considerando múltiples criterios que pueden tener diferentes importancias [10].

Siendo las alternativas $A_i, i = 1, 2, \dots, m$, los criterios $C_j, j = 1, 2, \dots, n$, los pesos de los criterios w_j y una matriz de decisión con $x_{ij} = U_j(A_i), \forall i, j$, donde U es la función de utilidad del tomador de decisiones, operando bajo la premisa de que todos los criterios deben maximizarse o minimizarse dependiendo de si son de beneficio o coste, respectivamente. El TOPSIS se basa en la idea de considerar la solución ideal positiva en \mathbb{R}^n al punto A^+ , siendo la mejor opción concebible, y A^- , la solución ideal negativa en \mathbb{R}^n , siendo la peor alternativa posible. Un decisor racional siempre debe

seleccionar A^+ , y si este valor no coincidiera con alguna de las alternativas, elegiría aquella que estuviera más próxima a ella [10].

Los pasos a seguir en el TOPSIS serían:

1. Elaboración de la matriz de decisión: Teniendo m alternativas A_i , $i = 1, 2, \dots, m$ que serán evaluadas a partir de los criterios C_j , $j = 1, 2, \dots, n$ se alcanza la siguiente matriz de decisión:

	w_1	w_2	\dots	w_n
C_1	x_{11}	x_{12}	\dots	x_{1n}
C_2	x_{21}	x_{22}	\dots	x_{2n}
\vdots	\vdots	\vdots	\ddots	\vdots
C_n	x_{n1}	x_{n2}	\dots	x_{nn}

Tabla. 2.1: Matriz de decisión

La x_{ij} simboliza la valoración de la alternativa A_i con respecto al criterio C_j y $W = [w_1, w_2, \dots, w_n]$ es el vector de pesos asociado a los criterios.

2. Normalizar la matriz: Los elementos pueden no estar en un mismo dominio y tienen que normalizarse. La norma es:

$$n_{ij} = \frac{x_{ij}}{\sqrt{\sum_{k=1}^m x_{kj}^2}}, \quad j = 1, \dots, n, \quad i = 1, \dots, m \quad (2.2)$$

3. Ponderación de la matriz normalizada: Estos elementos se calcularán con esta expresión:

$$v_{ij} = w_j \times n_{ij}, \quad j = 1, \dots, n, \quad i = 1, \dots, m \quad (2.3)$$

Teniendo en cuenta que la w_j es el peso al criterio relacionado.

4. Obtención de soluciones ideales: Tanto positivas como negativas, los valores ideales se obtienen con las siguientes expresiones:

$$\{A^+ = (v_1^+, v_2^+, \dots, v_n^+)\} = \left\{ (v_j^+) \mid v_j^+ = \max_i(v_{ij}), \forall j \in J \text{ o } v_j^+ = \min_i(v_{ij}), \forall j \in J' \right\} \quad (2.4)$$

$$\{A^- = (v_1^-, v_2^-, \dots, v_n^-)\} = \left\{ (v_j^-) \mid v_j^- = \max_i(v_{ij}), \forall j \in J \text{ o } v_j^- = \min_i(v_{ij}), \forall j \in J' \right\} \quad (2.5)$$

5. Cálculo de distancias: Las distancias tanto a la solución ideal como a la anti-ideal:

$$d_i^+ = \left(\sum_{j=1}^n (v_{ij} - v_j^+)^2 \right)^{\frac{1}{2}}, \quad i = 1, \dots, m \quad (2.6)$$

$$d_i^- = \left(\sum_{j=1}^n (v_{ij} - v_j^-)^2 \right)^{\frac{1}{2}}, \quad i = 1, \dots, m \quad (2.7)$$

6. Cálculo del coeficiente de cercanía: Esta ecuación nos indica que cuanto más cercano a 1 el valor de R más cerca está la alternativa i -ésima a la solución ideal.

$$R_i = \frac{d_i^-}{d_i^+ + d_i^-}, \quad i = 1, \dots, m \quad (2.8)$$

7. Clasificación de prioridades: Las opciones serán ordenadas de manera descendente, iniciando con la que esté más cercana a la solución ideal, es decir, máxima proximidad relativa.

Capítulo 3

DESARROLLO DEL PROYECTO

En este capítulo, se llevará a cabo una descripción en detalle del desarrollo del proyecto. Se describirá el sistema de una forma completa, así como la implementación del proyecto.

3.1. Descripción del Proyecto

El objetivo principal de este proyecto es diseñar e implementar un asistente web que facilite a los usuarios de FUT la identificación de los mejores jugadores para sus equipos, basándose en diversos atributos y criterios específicos. Este asistente proporcionará una solución inteligente y personalizada para la selección de jugadores, optimizando así la experiencia de los usuarios en el juego.

El desarrollo implicará la extracción y gestión de una gran cantidad de datos sobre los jugadores disponibles en FUT. Esto incluirá información detallada sobre sus atributos, precios en el mercado de transferencias, y estadísticas de rendimiento. Se utilizarán técnicas avanzadas de scraping y bases de datos para manejar y procesar esta información de manera eficiente.

3.2. Análisis del sistema

Esta sección llevará a cabo un análisis del sistema, incluyendo una especificación de requerimientos y su modelado.

3.2.1. Especificación de Requerimientos

En esta subsección definiremos los requisitos funcionales y no funcionales del proyecto, pero antes, se va a definir qué son los requisitos funcionales y no funcionales:

- **Requisitos funcionales:** Especifican las propiedades y capacidades que exigen tener una entidad, como un sistema, es decir, definen lo que hace el sistema.
- **Requisitos no funcionales:** Especifican restricciones de calidad, tales como usabilidad, mantenibilidad o rendimiento, sobre esas propiedades y capacidades.

La única diferencia entre ambos radica en que el sistema no puede operar a menos que cumpla con todos los requisitos funcionales. Además, el sistema entregará el resultado esperado aunque no cumpla con los requisitos no funcionales.

Ambos tipos de requisitos son complementarios y deben ser elaborados simultáneamente para asegurar una comprensión completa del alcance y los objetivos del proyecto.

Requerimientos funcionales

1. REQF01: Búsqueda y filtrado de jugadores. El asistente debe permitir a los usuarios buscar jugadores por nombre.
2. REQF02: Visualización de Atributos y Estadísticas. Los usuarios deben poder ver una lista detallada de atributos y estadísticas de cada jugador.
3. REQF03: La plataforma debe tener una interfaz amigable y fácil de usar, con menús y botones claramente identificados.
4. REQF04: La plataforma tiene que dar al mejor jugador basado en las estadísticas requeridas.
5. REQF05: El sistema debe ser capaz de encontrar un listado recomendado de jugadores.

Requerimientos no funcionales

1. REQNF01: Las recomendaciones de jugadores deben generarse en un tiempo razonable (menos de 3 segundos).
2. REQNF02: El código debe estar bien documentado y estructurado para facilitar el mantenimiento y las futuras actualizaciones.
3. REQNF03: Mostrar la información lo más clara posible para que resulte fácil de entender para el usuario.

3.2.2. Modelado del sistema

Este sistema va dirigido a los siguientes tipos de usuario:

- Usuarios:

1. Jugadores de FUT:

- a) Usuarios regulares: Jugadores ocasionales que buscan mejorar su equipo de manera eficiente sin invertir demasiado tiempo en análisis detallados.
- b) Usuarios avanzados: Jugadores competitivos que participan en torneos y ligas, y requieren optimizar sus plantillas al máximo.
- c) Usuarios profesionales: Jugadores de eSports y streamers que necesitan seleccionar los mejores jugadores para mantener su rendimiento en competiciones y para su audiencia.

2. Analistas del juego: Individuos o equipos que analizan las estadísticas de los jugadores para proporcionar consejos estratégicos a otros jugadores o equipos.

Basándose en estos tipos de usuario, un diagrama de casos de uso ayudará a visualizar las interacciones entre los usuarios y el sistema. En este contexto, nuestros actores principales son los usuarios, que pueden ser regulares, avanzados, o profesionales.

Casos de uso para el Usuario

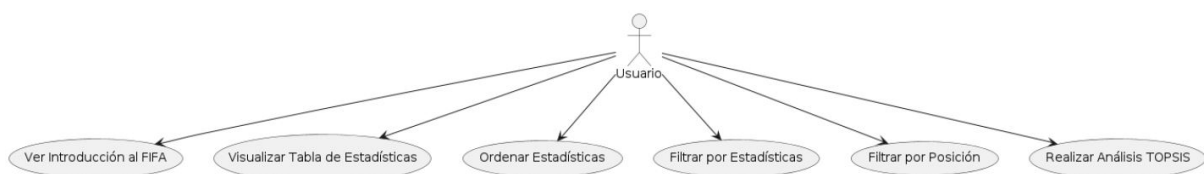


Figura 3.1: Casos de uso Global

En el caso de uso mostrado en Figura 3.2 los usuarios pueden acceder a una página que proporciona una introducción al juego y sus mecanismos. El usuario accede a la página principal y selecciona la opción “*Introducción al FIFA*”.

En el caso de uso mostrado en Figura 3.3 llamado “*Visualizar Tabla de Estadísticas*”, los usuarios pueden ver una tabla que muestra las estadísticas de los jugadores. El usuario accede a la página de estadísticas y visualiza la tabla con los datos de los jugadores.

En el caso de uso mostrado en Figura 3.4 llamado “*Ordenar Estadísticas*”, los usuarios pueden ordenar la tabla de estadísticas por diferentes atributos. El usuario



Figura 3.2: Caso de uso CUU1

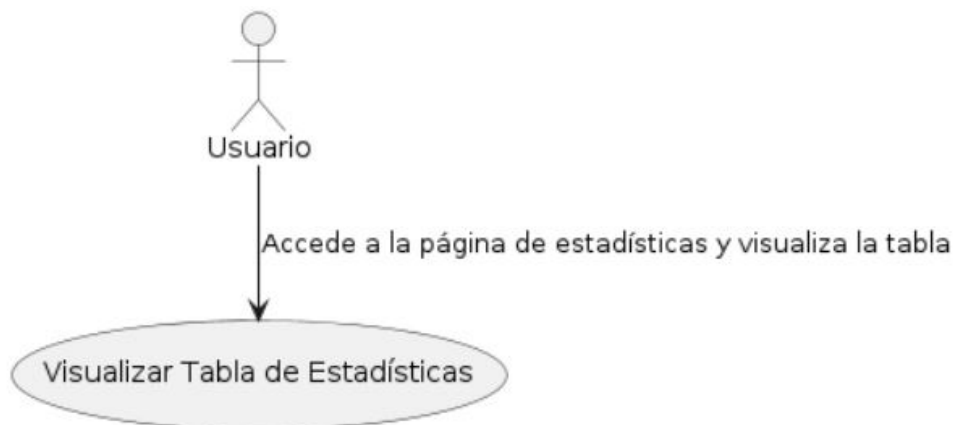


Figura 3.3: Caso de uso CUU2

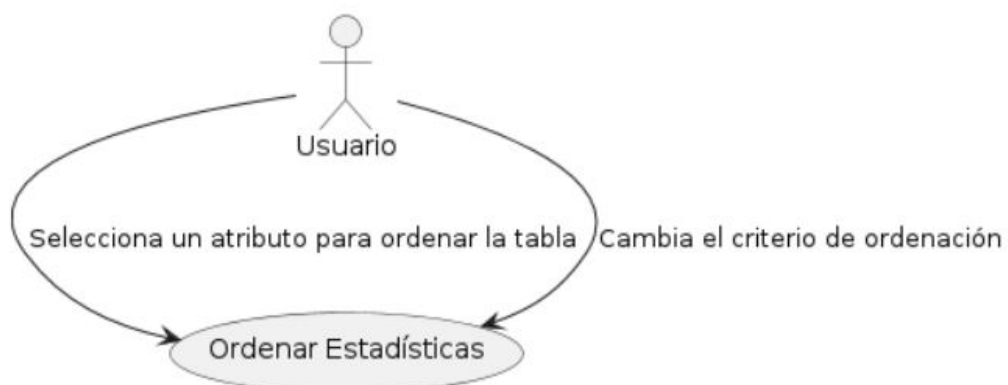


Figura 3.4: Caso de uso CUU3

selecciona un atributo por el cual ordenar la tabla y los datos se reordenan en consecuencia. Surgiría un flujo secundario en el que el usuario puede cambiar el

criterio de ordenación seleccionando otro atributo.



Figura 3.5: Caso de uso CUU4

En el caso de uso mostrado en Figura 3.5 llamado “*Filtrar por Estadísticas*”, los usuarios pueden aplicar filtros para ver jugadores que cumplen con ciertos criterios estadísticos. El usuario selecciona los filtros deseados y la tabla se actualiza para mostrar solo los jugadores que cumplen con esos criterios. Surgiría un flujo secundario donde el usuario puede modificar los filtros para ajustar los resultados.



Figura 3.6: Caso de uso CUU5

En el caso de uso mostrado en Figura 3.6 llamado “*Filtrar por Posición*”, los usuarios pueden filtrar los jugadores por su posición en el campo. El usuario selecciona una posición específica y la tabla se actualiza para mostrar solo los jugadores que juegan en esa posición. El flujo secundario sería que el usuario puede cambiar la posición seleccionada para ver jugadores en otras posiciones.

En el caso de uso 3.7 llamado “*Realizar Análisis TOPSIS*”, los usuarios pueden utilizar la técnica TOPSIS para identificar al mejor jugador según los resultados deseados. El usuario accede a la herramienta TOPSIS, define los criterios y pesos, y ejecuta el análisis para obtener una recomendación del mejor jugador. El flujo secundario estaría en que el usuario puede ajustar los criterios y pesos y ejecutar nuevamente el análisis para refinar los resultados.

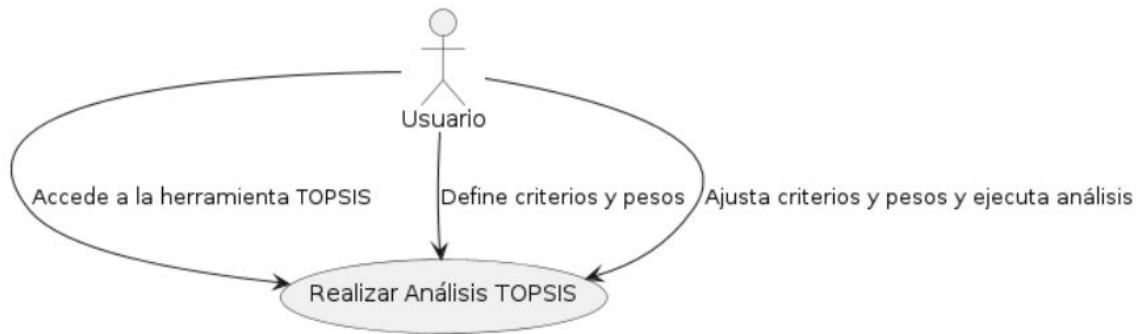


Figura 3.7: Caso de uso CUU6

Diagrama de actividades

Un diagrama de actividades (ver Figura 3.8) es una representación gráfica de los flujos de trabajo del sistema y se utiliza para modelar la lógica del flujo de control de un sistema, describiendo la secuencia de actividades y decisiones que ocurren dentro de un proceso. Un diagrama de actividades ayuda a entender el flujo de trabajo entre los distintos casos de uso.

3.3. Diseño del sistema

Esta sección pretende abarcar todo el diseño de la aplicación, para ello se centrará en el diseño que tienen sus objetos, con diagrama de clase y de secuencia, el diseño que tiene la interfaz, abarcando todas las vistas, y un diseño de la base de datos, realizado con un modelo Entidad-Relación.

3.3.1. Diseño de los objetos

Para entender bien un diseño del sistema es necesario realizar un diagrama de objetos, que para ello se nutrirá de un diagrama de clases y de secuencia.

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema enseñando sus clases, atributos y sus conexiones entre sí. En un diagrama de clases se pueden distinguir principalmente dos elementos: clases y sus interrelaciones [6].

En el diagrama de clases mostrado en la Figura 3.9 se indica que un club puede tener varios jugadores, pero solo puede pertenecer a 1 liga, mientras que 1 jugador solo tendrá 1 nación. Aunque existen varios jugadores que poseen varias nacionalidades, FIFA solo reconoce la nacionalidad con la que juegan en la selección, por lo tanto, en el diagrama de clases esa relación se verá como 1 a 1.

Un diagrama de secuencias modela la interacción entre los objetos de un sistema de

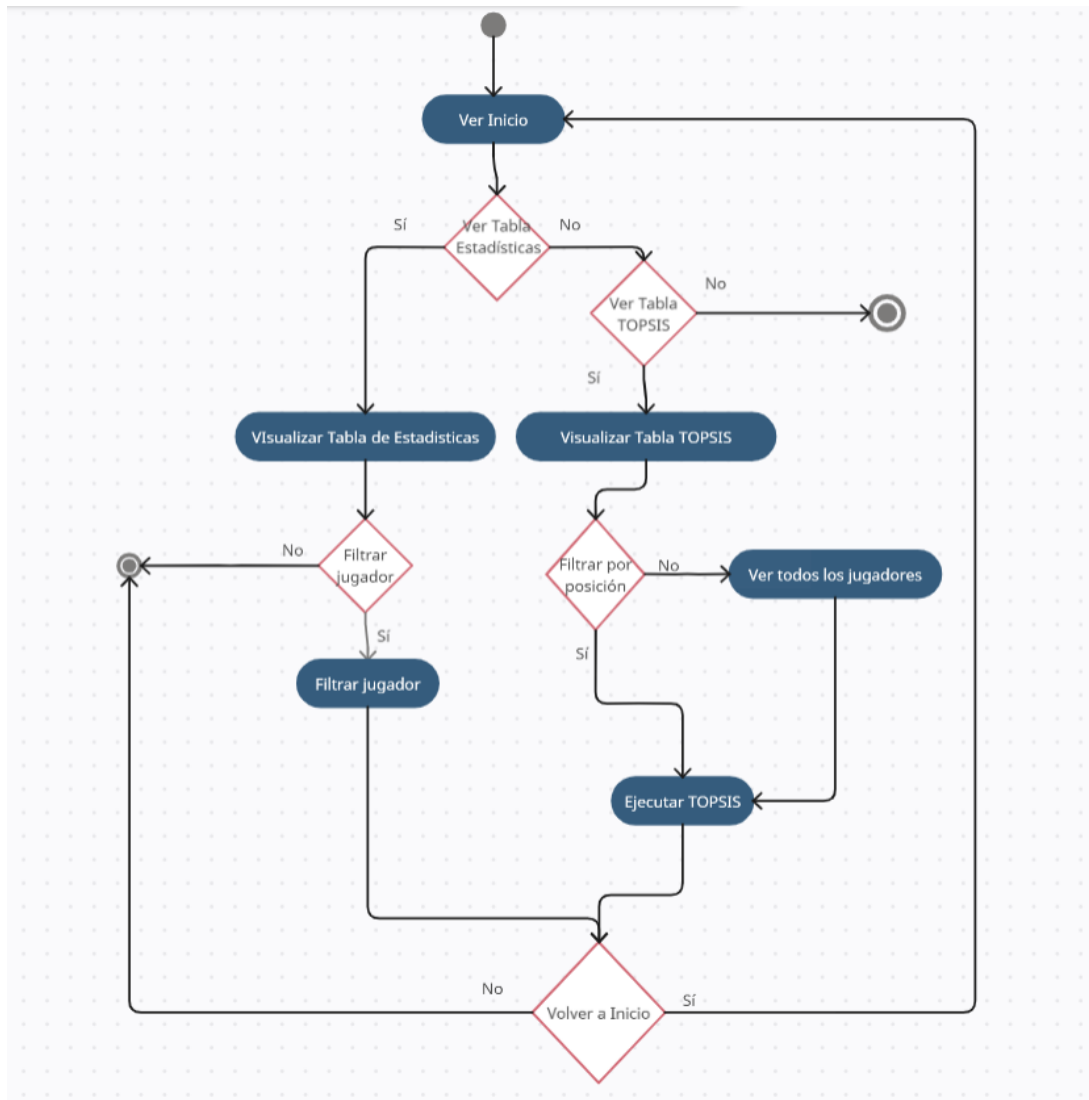


Figura 3.8: Diagrama de actividades

software, con ello podemos saber cómo funcionan las interacciones que tenemos con el sistema. En nuestra aplicación se muestran dos funcionalidades muy importantes: la de filtrar jugadores, representado en el diagrama de secuencia de la Figura 3.10, y el de ejecutar TOPSIS, mostrado en el diagrama de secuencia de la Figura 3.11.

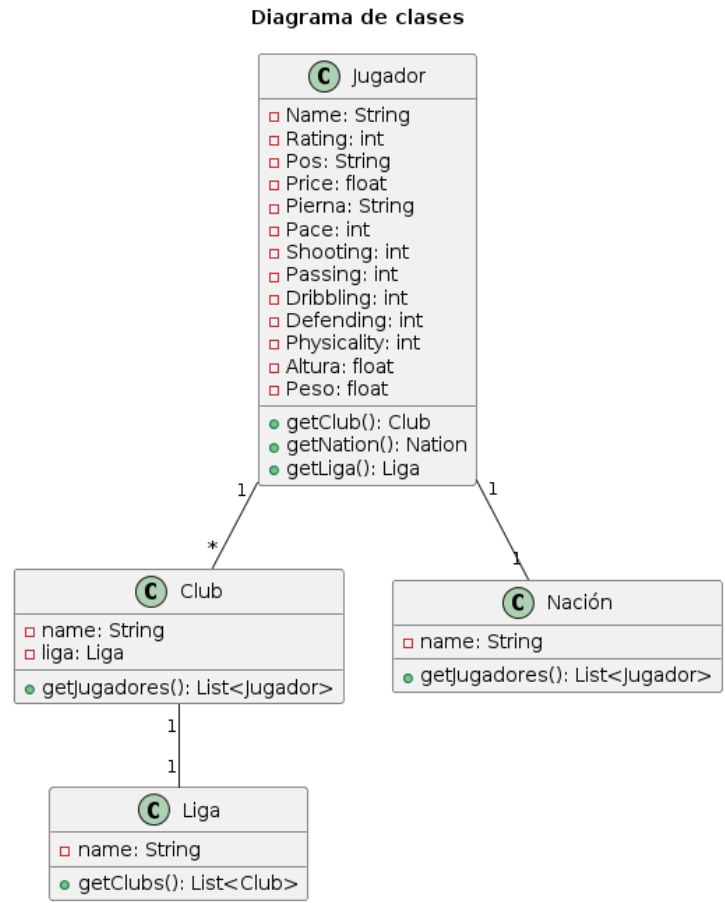


Figura 3.9: Diagrama de clases

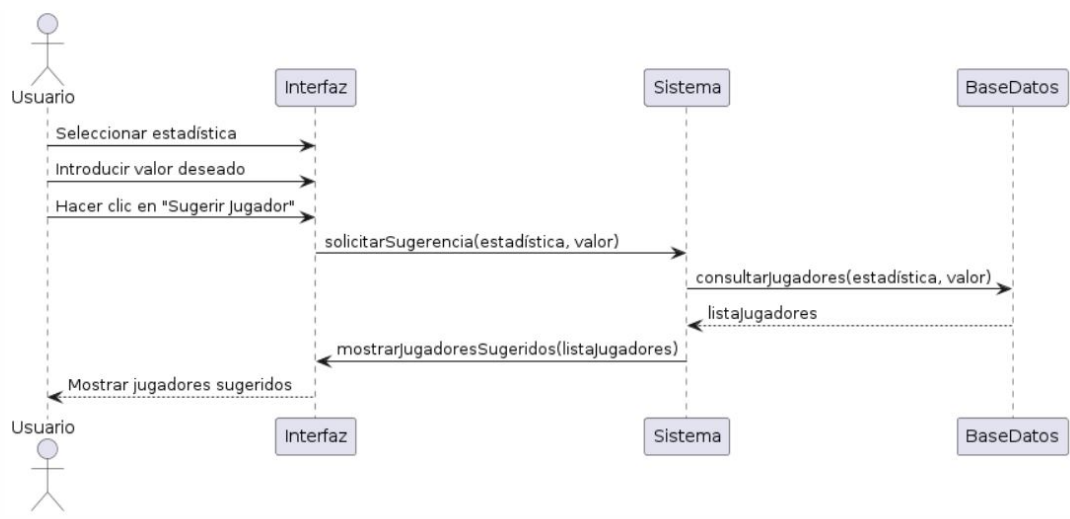


Figura 3.10: Diagrama de secuencia filtrar jugador

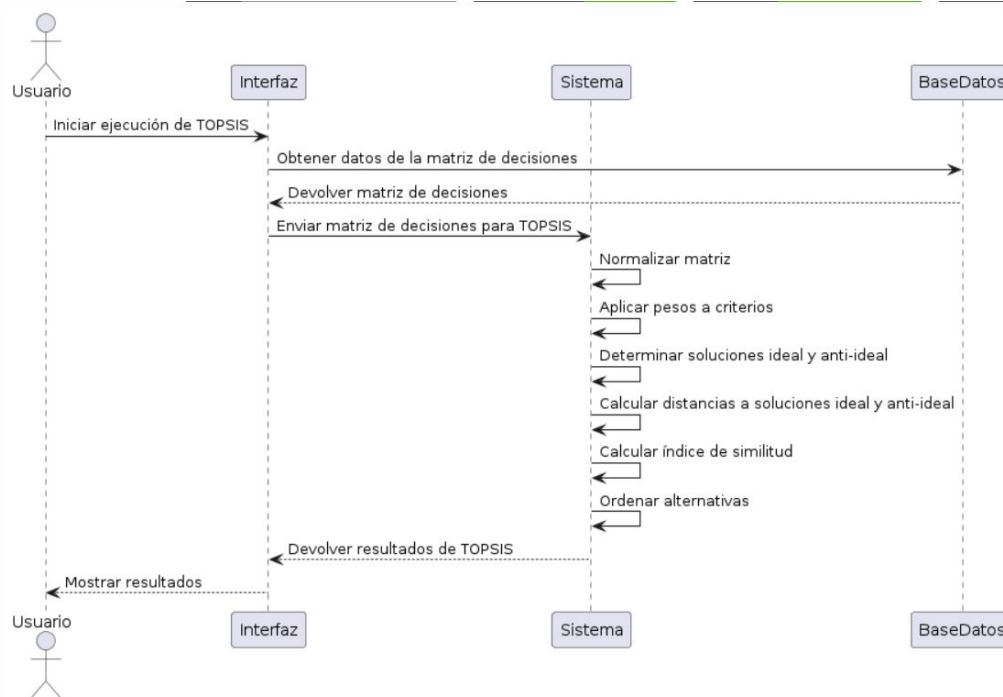


Figura 3.11: Diagrama de secuencia Topsis

3.3.2. Diseño de la interfaz

En esta subsección se mostrarán las vistas principales de la plataforma que tienen que estar incluidas en la aplicación para asegurar que se cumplen los requisitos especificados respetando los criterios de usabilidad. Además, se describirá el storyboard, fundamental para poder visualizar cómo se van a conectar las diferentes vistas y cómo se va a desarrollar la navegación dentro de la aplicación.

Storyboard

El storyboard, que es definido como guion gráfico secuencial que muestra el contenido y el flujo de nuestra página web, nos ayudará a presentar todas las vistas desde un punto de vista más sencillo, centrándose en el diseño [24].

Las vistas de inicio (Figura 3.12) y FUT (Figura 3.13) presentan características similares, dado que son una introducción a lo que es el videojuego FIFA y a la modalidad del juego FUT. Están representadas en una temática de fondo azul con un fondo blanco para el texto, facilitando así la lectura y siendo cómodo de entender.

En la vista de alineaciones (Figura 3.14), podemos observar que se nos mostrará la tabla de jugadores entera. Habrá un selector para elegir la estadística que queramos buscar y el valor deseado que queremos obtener. Tras ello y tras haber seleccionado el botón sugerir jugador, se nos mostrará la vista de filtrar jugador (Figura 3.15), si el atributo no existe se nos indicará en la vista mostrada en Figura 3.16. A la derecha se

Id Vista	Vista	Descripción
1	Inicio	Te permite ver la introducción al FIFA
2	Ultimate Team	Te permite ver la introducción al FUT
3	Alineaciones	Te permite ver la tabla con todos los jugadores
3.1	Seleccionar estadística	Selector para ver que estadística quieres ver
3.2	Sugerir Jugador	Una vez seleccionado la estadística y el valor deseado, puedes ver los jugadores sugeridos
4	TOPSIS	Tabla con todos los jugadores
4.1	Seleccionar posición	Te permite ver todos los jugadores de la posición filtrada
4.2	Run TOPSIS	Te permite ver todos los jugadores de la posición filtrada con el TOPSIS realizado y con la opción de mostrar la tabla original

Tabla. 3.1: Tabla de vistas principal



Figura 3.12: Vista Inicio

muestra un botón transparente que nos permite subir arriba de la tabla para que cuando bajemos mucho no sea incómodo tener que subir de nuevo, haciendo la vista más dinámica y sencilla.

Finalmente, la vista representada en la Figura 3.17 muestra la tabla original con todos los jugadores y pudiéndose filtrar por jugador, en este caso se ha filtrado por jugadores cuya posición en el campo es LWB (carrilero izquierdo) que se muestran en la Figura 3.17, y una vez ejecutado el TOPSIS, pulsando el botón correspondiente

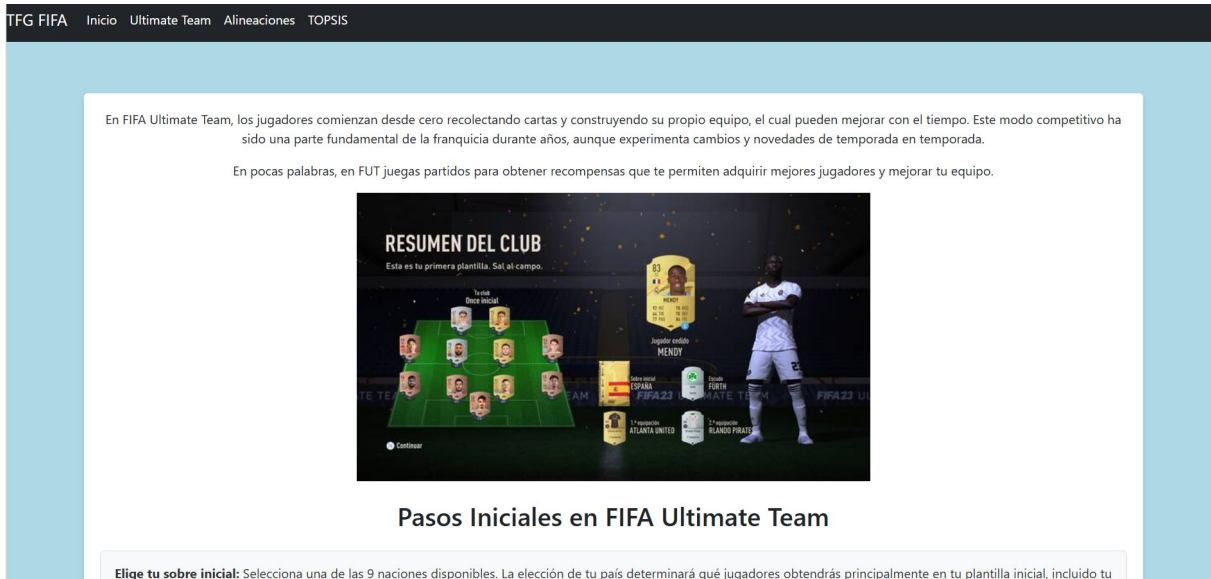


Figura 3.13: Vista Ultimate Team

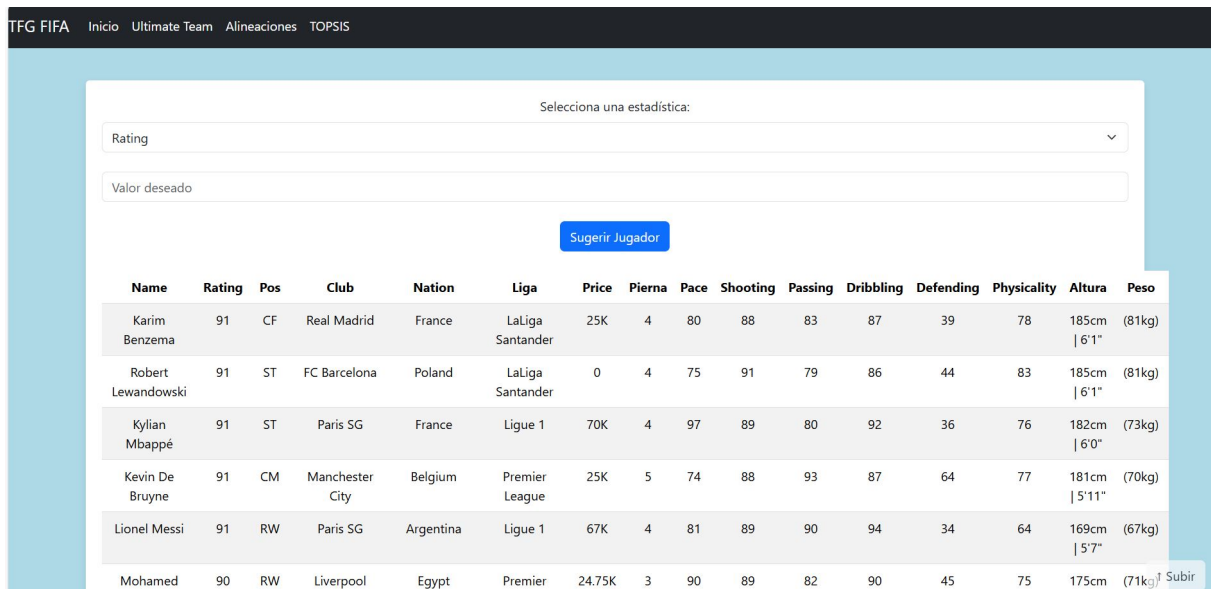


Figura 3.14: Vista Alineaciones

(ver Figura 3.18), se obtienen los jugadores más rápidos, dado que en ese momento el TOPSIS tiene asignado peso 1 para el criterio velocidad.

3.3.3. Diseño de la base de datos

En esta subsección entenderemos como funciona la base de datos para tener más claro el funcionamiento de la aplicación, para ello se aplicará un modelo Entidad-Relación.

El modelo Entidad-Relación (MER) es uno de los más utilizados y fue introducido

TFG FIFA Inicio Ultimate Team Alineaciones TOPSIS

Selecciona una estadística:

Pos

CF

Sugerir Jugador

Jugadores Sugeridos:

Name	Rating	Pos	Club	Nation	Liga	Price	Pierna	Pace	Shooting	Passing	Dribbling	Defending	Physicality	Altura	Peso
Karim Benzema	91	CF	Real Madrid	France	LaLiga Santander	25K	4	80	88	83	87	39	78	185cm 6'1"	(81kg)
Christopher Nkunku	86	CF	RB Leipzig	France	Bundesliga	10K	3	88	81	83	88	65	66	175cm 5'9"	(73kg)
Paulo Dybala	86	CF	Roma FC	Argentina	Serie A TIM	8.5K	3	80	85	85	90	40	59	177cm 5'10"	(75kg)
Diogo Jota	85	CF	Liverpool	Portugal	Premier League	9.7K	5	85	83	75	85	57	77	178cm 5'10"	(73kg)
Memphis Depay	85	CF	FC Barcelona	Netherlands	LaLiga Santander	20K	3	83	84	82	86	30	79	176cm 5'9"	(78kg)

1 Subir

Figura 3.15: Filtrar Jugador

TFG FIFA Inicio Ultimate Team Alineaciones TOPSIS

Selecciona una estadística:

Rating

97

Sugerir Jugador

No se encontraron jugadores que coincidan con los criterios de búsqueda.

Name	Rating	Pos	Club	Nation	Liga	Price	Pierna	Pace	Shooting	Passing	Dribbling	Defending	Physicality	Altura	Peso
Karim Benzema	91	CF	Real Madrid	France	LaLiga Santander	25K	4	80	88	83	87	39	78	185cm 6'1"	(81kg)
Robert Lewandowski	91	ST	FC Barcelona	Poland	LaLiga Santander	0	4	75	91	79	86	44	83	185cm 6'1"	(81kg)
Kylian Mbappé	91	ST	Paris SG	France	Ligue 1	70K	4	97	89	80	92	36	76	182cm 6'0"	(73kg)
Kevin De Bruyne	91	CM	Manchester City	Belgium	Premier League	25K	5	74	88	93	87	64	77	181cm 5'11"	(70kg)
Lionel Messi	91	RW	Paris SG	Argentina	Ligue 1	67K	4	81	89	90	94	34	64	169cm 5'6"	(67kg)

1 Subir

Figura 3.16: Error Sugerir Jugador

por Chen en 1976. Surge por la necesidad de contar con una herramienta para permitir el entendimiento y la comprensión entre distintos tipos de usuarios (diseñador, desarrollador y usuarios finales) con diferentes perspectivas [4].

El MER se basa en una representación del mundo real que incluye una colección de elementos fundamentales, denominados entidades, y las relaciones entre estos elementos.

A continuación se definen los elementos más relevantes de un MER:

- Entidad: Objeto que existe y es distinguible entre otros objetos.

TFG FIFA Inicio Ultimate Team Alineaciones TOPSIS

Jugadores

Defensa Lateral Izquierdo (LWB)

Run TOPSIS

Name	Rating	Pos	Club	Nation	Liga	Price	Pierna	Pace	Shooting	Passing	Dribbling	Defending	Physicality	Altura	Peso
Angeliño	83	LWB	TSG Hoffenheim	Spain	Bundesliga	1.2K	3	78	71	82	84	76	70	171cm 5'7"	(69kg)
Ben Chilwell	82	LWB	Chelsea	England	Premier League	750	3	76	60	78	78	78	75	180cm 5'11"	(77kg)
Robin Gosens	82	LWB	Inter	Germany	Serie A TIM	750	3	82	75	74	79	77	80	183cm 6'0"	(76kg)
Leonardo Spinazzola	82	LWB	Roma FC	Italy	Serie A TIM	850	4	91	64	73	80	78	69	186cm 6'1"	(75kg)

Figura 3.17: Vista TOPSIS

TFG FIFA Inicio Ultimate Team Alineaciones TOPSIS

Jugadores

Defensa Lateral Izquierdo (LWB)

Run TOPSIS Mostrar Tabla Original

Resultados TOPSIS

Name	Rating	Pos	Club	Nation	Liga	Price	Pierna	Pace	Shooting	Passing	Dribbling	Defending	Physicality	Altura	Peso
Przemyslaw Frankowski	76	LWB	RC Lens	Poland	Ligue 1	500	3	92	70	72	74	65	72	175cm 5'9"	(70kg)
Leonardo Spinazzola	82	LWB	Roma FC	Italy	Serie A TIM	850	4	91	64	73	80	78	69	186cm 6'1"	(75kg)
Rico Henry	75	LWB	Brentford	England	Premier League	650	2	90	40	59	74	70	72	170cm 5'7"	(66kg)
David Raum	81	LWB	RB Leipzig	Germany	Bundesliga	700	2	87	59	77	78	71	77	180cm 5'11"	(75kg)

Figura 3.18: Vista TOPSIS Realizado

- Entidad física: Persona, trabajador, evento, compañía.
 - Entidad lógica: Proyecto, materia, pago.
- Conjunto de entidades: Conjunto de entidades del mismo tipo que comparten atributos en común.
 - Atributo: Características que detallen las entidades. Estos pueden ser simples, derivados y monovalorados o multivalorados.
 - Conjunto de relaciones: Asociación entre entidades representadas por un rombo.

El MER tiene que disponer de restricciones, que son reglas que deben mantener los datos. Las restricciones dispondrán de cardinalidad para indicar el número de relaciones existentes entre entidades, así como claves.

Una clave es un identificador de un conjunto de atributos que permite distinguir entidades. Existen tres tipos de clave:

- **Clave primaria:** La clave primaria o candidata es la elegida para identificar a una entidad.
- **Superclave:** Son varios o un único atributo que identifican de modo único una entidad dentro del conjunto de entidades.
- **Clave candidata:** Es una superclave en la que ninguno de sus subconjuntos es superclave.

Entendiendo estos conceptos, nuestro MER sería el siguiente (ver Figura 3.19):

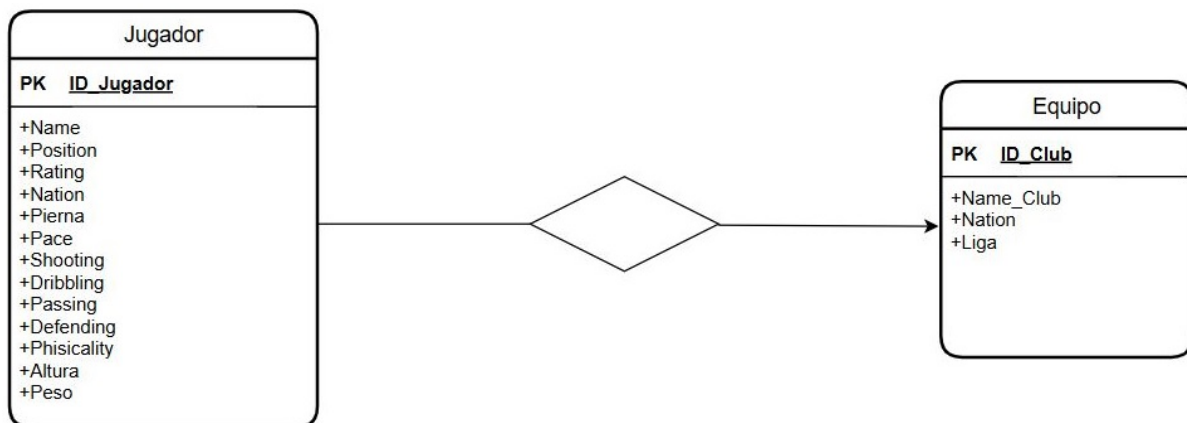


Figura 3.19: Modelo Entidad-Relación

Entendiendo que un jugador puede pertenecer a un solo club, pero un club puede tener asignados a varios jugadores.

3.4. Implementación del sistema

Una vez explicado el análisis y diseño del sistema, debemos empezar la fase de desarrollo del sistema. Para ello tenemos que abordar las tecnologías usadas, es decir, framework y lenguaje de programación, así como el desarrollo del frontend.

En el enlace <https://github.com/JoseEspinar/TFG/tree/master> puede descargarse el código de la aplicación.

3.4.1. Tecnologías usadas

En esta sección se describen las tecnologías usadas en el desarrollo del TFG.

REACT

React, también conocido como React.js o ReactJS, es una biblioteca JavaScript de código abierto diseñada para crear interfaces de usuario. Esta biblioteca se ha especializado en el desarrollo de aplicaciones de una sola página (SPA), ofreciendo un rendimiento sobresaliente y un enfoque de desarrollo similar al de los videojuegos. React es mantenida por Facebook, Instagram y una amplia comunidad de desarrolladores independientes y empresas.

La historia de React comienza en 2011, cuando un pequeño equipo en Facebook empezó a desarrollar una versión de XHP, un lenguaje derivado de PHP creado por Facebook poco antes. XHP fue diseñado para reducir los ataques XSS (Cross-Site Scripting) mediante una sintaxis XML que permitía la creación de elementos HTML personalizables y reutilizables. Sin embargo, surgió un problema con XHP: las aplicaciones SPA requerían muchas más peticiones al servidor, y XHP no podía manejarlas de manera eficiente. Ante este desafío, un grupo de ingenieros de Facebook propuso utilizar XHP en el navegador a través de JavaScript, y recibieron seis meses para experimentar con esta idea. El resultado fue ReactJS, que se lanzó como software de código abierto en 2013 [12].

Las ventajas de usar React incluyen la reutilización de componentes, la separación de responsabilidades, la claridad en el origen de datos o variables, y la actualización solo del componente que ha recibido datos nuevos. Además, es fácil de adaptar a nuevas versiones, tiene un núcleo pequeño y una curva de aprendizaje corta. Sin embargo, presenta desventajas como la necesidad de mantener un orden estricto en el desarrollo para mejorar el rendimiento de la aplicación, y su enfoque contrasta con los principios de la Open Web, obligando al uso de JSX ¹ y virtual DOM [8].

Una de las propiedades más notables que incluye React y que se han utilizado en el desarrollo de la aplicación han sido los “Hooks”, una característica añadida en la versión 16.8 de React, que nos deja usar estados y otras características de React sin escribir clases y que aunque es un componente opcional, han sido usados en el desarrollo de la aplicación, ya que hacen que el código sea más sencillo y legible [8].

Principalmente, los que han sido más útiles a lo largo del desarrollo de la aplicación han sido los Hooks de estado (ver Figura 3.20) y los de efecto (ver Figura 3.21):

```
const [selectedPosition, setSelectedPosition] = useState('');
```

Figura 3.20: Hook de estado

El hook `useState` se usa para manejar el estado de las variables dentro del componente. Cada llamada a `useState` retorna una pareja de valores: el estado actual y una función para actualizarlo. En este caso, se llama al estado posición y se usa la función `setSelectedPosition` para actualizarla.

¹ JSX, no es más que una extensión de la sintaxis de JavaScript basada en etiquetas, que nos puede recordar a HTML.

```
useEffect(() => {  
  if (selectedPosition === '') {  
    setFilteredData(csvData);  
  } else {  
    setFilteredData(csvData.filter(row => row.Pos === selectedPosition));  
  }  
}, [selectedPosition, csvData]);
```

Figura 3.21: Hook de efecto

El hook `useEffect` se utiliza para manejar efectos secundarios en componentes funcionales. Este `useEffect` se ejecuta cada vez que cambian los valores de `selectedPosition` o `csvData`, y filtra los datos en consecuencia. Además, el hook `useEffect` se ejecuta automáticamente después del primer renderizado y tras cada actualización subsiguiente. Se utiliza cuando queremos que un componente realice alguna acción una vez que se ha renderizado.

Scrapy

Para tener la base de datos con la que poder analizar jugadores para poder hacerle el TOPSIS, se usa el scrapeo, que utiliza web scraping, que fue detallado en la Subsección [2.3.1](#).

El lenguaje de programación más usado para obtener datos de páginas web es Python, ya que alberga varias librerías para este cometido: Scrapy, BeautifulSoup y Selenium. Scrapy es un framework que deja administrar peticiones, preservar sesiones de usuario y seguir redirecciones [13].

Una de las mayores ventajas de Scrapy es que es muy eficiente y es capaz de obtener más cantidad de datos, más rápido y a menor coste de CPU que las otras alternativas.

Para poder obtener datos se utilizan bots, spiders o crawlers para extraer contenido y datos de un sitio web. Para obtener los datos y extraerlos en un CSV, se utiliza técnicas de raspado web como XPATH, también comentado en la Subsección [2.3.2](#).

Para que el funcionamiento sea correcto y se obtenga una tabla primeramente, hay que definirla con las columnas o items que se quiera obtener.

Como podemos observar en la Figura [3.23](#), para obtener los datos hace falta una preparación en el spider, colocando el nombre, la opción `allowed domains` que permite esos dominios web para poder scrapearlos, que `url` o `urls` vamos a scrapear, una personalización de como se va a guardar, que en este caso es en formato CSV. La opción `ROBOTSTXT OBEY` se marca a `true` porque algunas páginas web utilizan el archivo `robots.txt` para indicarles a los bots cómo pueden navegar por su sitio web. Un archivo `robots.txt` alberga una lista de agentes de usuario y rutas asociadas que los

```

class Campos(Item):
    Name = Field ()
    Rating = Field()
    Pos = Field ()
    Club = Field ()
    Nation = Field()
    Liga = Field ()
    Price = Field ()
    Pierna = Field()
    Pace = Field ()
    Shooting = Field ()
    Passing = Field ()
    Dribbling = Field ()
    Defending = Field ()
    Physicality = Field ()
    Altura = Field ()
    Peso = Field ()

```

Figura 3.22: Elaboración de elementos a obtener en el CSV

```

class Players(scrapy.Spider):
    name='players'
    allowed_domains = ['www.futbin.com', 'futbin.com']
    ...

    def start_requests(self):
        base_url = 'https://www.futbin.com/23/players?page={}&version=gold&pos_type=all'
        for page in range(1, 63):
            url = base_url.format(page)
            yield scrapy.Request(url, self.parse)
        ...

    start_urls=[
        'https://www.futbin.com/23/players?page=62&version=gold&pos_type=all'
    ]

    custom_settings={
        'FEEDS': {
            'jugadores.csv': {
                'format': 'csv',
                'encoding': 'utf-8',
            },
        },
        'ROBOTSTXT_OBEY': True,
    }

```

Figura 3.23: Creación Spider

agentes de usuario pueden o no seguir.

```
def parse(self, response):
    players = response.xpath('//tbody/tr')
    for player in players:
        item = ItemLoader(item=Campos(), selector=player)

        # Agrega los atributos del jugador al ItemLoader
        item.add_xpath('Name', './td[@class="table-row-text row"]/div[@class="d-inline pt-2 pl-3"]/div/a/text()')
        item.add_xpath('Rating', './td[3]/span/text()')
        item.add_xpath('Price', './td[6]/span/text()')
        item.add_xpath('Club', './td[@class="table-row-text row"]/div[@class="d-inline pt-2 pl-3"]/div/span/a[1]/@data-original-title')
        item.add_xpath('Nation', './td[@class="table-row-text row"]/div[@class="d-inline pt-2 pl-3"]/div/span/a[2]/@data-original-title')
        item.add_xpath('Liga', './td[@class="table-row-text row"]/div[@class="d-inline pt-2 pl-3"]/div/span/a[3]/@data-original-title')
        item.add_xpath('Pos', './td/div[@class="font-weight-bold"]/text()')
        item.add_xpath('Pierna', './td[8]/text()')
        item.add_xpath('Altura', './td[16]/div[1]/text()')
        item.add_xpath('Peso', './td/div[@style="font-size: 12px; "]/text()[2]')
        item.add_xpath('Pace', './td[10]/span/text()')
        item.add_xpath('Shooting', './td[11]/span/text()')
        item.add_xpath('Passing', './td[12]/span/text()')
        item.add_xpath('Dribbling', './td[13]/span/text()')
        item.add_xpath('Defending', './td[14]/span/text()')
        item.add_xpath('Physicality', './td[15]/span/text()')

        # Carga y emite el ítem con todos los atributos del jugador
        yield item.load_item()
```

Figura 3.24: Obtención de raspado

Por último, en la Figura 3.24 podemos observar como se utiliza XPATH para obtener los items necesarios de la tabla. Para ello, se inspecciona con el navegador la página web y se observa en elementos las partes que nos interesan, en nuestro caso son varias características de jugadores de carta de oro común.

3.4.2. Frontend

El frontend de la aplicación es crucial para la interacción del usuario, proporcionando una interfaz amigable y eficiente para acceder a los datos y funcionalidades del sistema. Esta sección describe las tecnologías utilizadas, la estructura del frontend y los componentes desarrollados para la aplicación.

Tecnologías usadas del Frontend

React

Como ya se ha explicado anteriormente, React es una biblioteca de JavaScript desarrollada por Facebook para la construcción de interfaces de usuario. La principal ventaja de React es su capacidad para crear componentes reutilizables, lo que permite una organización modular del código y facilita el mantenimiento y la escalabilidad del proyecto. React utiliza un Virtual DOM que mejora la eficiencia en las actualizaciones de la interfaz, ya que solo actualiza los elementos que han cambiado, en lugar de renderizar toda la página nuevamente. Esta característica es esencial para aplicaciones dinámicas y de alto rendimiento, como la que se desarrolla en este proyecto.

React Router DOM

React Router DOM es una biblioteca que permite la navegación dinámica en una aplicación React. A través de la definición de rutas y componentes específicos para cada ruta, se puede crear una experiencia de usuario fluida y coherente. Esta biblioteca es fundamental para la estructuración de la aplicación, permitiendo la separación clara de diferentes secciones como “Inicio”, “Ultimate Team”, “Alineaciones” y “Análisis”. El uso de React Router DOM facilita la gestión del estado de la URL y la sincronización de la interfaz con la navegación del usuario.

PapaParse

PapaParse es una biblioteca de JavaScript que facilita la manipulación y el análisis de archivos CSV. En este proyecto, PapaParse se utiliza para leer y procesar los datos de jugadores almacenados en el archivo CSV. La capacidad de convertir estos datos en objetos JavaScript permite una integración directa con el resto de la lógica de la aplicación. PapaParse es eficiente y capaz de manejar grandes conjuntos de datos, lo que es crucial para analizar los atributos y estadísticas de numerosos jugadores de fútbol en tiempo real [20].

Axios

Axios es una biblioteca para realizar solicitudes HTTP desde el navegador. Se utiliza en este proyecto para comunicar el frontend con el backend, permitiendo el envío y recepción de datos de manera asíncrona. La simplicidad y versatilidad de Axios lo hacen ideal para manejar operaciones de red, como la obtención de datos de jugadores y la actualización de información en el servidor. Además, su compatibilidad con promesas facilita el manejo de respuestas y errores de manera más limpia y estructurada [2].

Linear Algebra y ml-matrix

Linear Algebra y ml-matrix son bibliotecas utilizadas para operaciones matemáticas y de álgebra lineal. Estas bibliotecas son cruciales para implementar el método de análisis multicriterio TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution). Linear Algebra proporciona herramientas básicas para operaciones matriciales, mientras que ml-matrix ofrece funcionalidades avanzadas para manipular y calcular matrices. Estas herramientas matemáticas permiten procesar y evaluar las estadísticas de los jugadores de manera precisa y eficiente.

react-csv-reader y react-papaparse

react-csv-reader y react-papaparse son componentes específicos de React que facilitan la carga y procesamiento de archivos CSV dentro de una aplicación React. react-csv-reader proporciona una interfaz fácil de usar para seleccionar y cargar archivos CSV, mientras que react-papaparse integra PapaParse directamente en los componentes React, simplificando aún más la manipulación de datos CSV. Estas herramientas permiten una interacción fluida y eficiente con los datos de jugadores, mejorando la experiencia del usuario al analizar y visualizar la información.

Web Vitals

Web Vitals es una iniciativa de Google que proporciona métricas esenciales sobre la calidad de la experiencia del usuario en la web. En este proyecto, Web Vitals se utiliza para monitorear y mejorar el rendimiento de la aplicación React. Las métricas ofrecidas, como el Largest Contentful Paint (LCP), el First Input Delay (FID) y el Cumulative Layout Shift (CLS), ayudan a identificar áreas de mejora en la velocidad y la interactividad de la aplicación. Integrar Web Vitals asegura que la aplicación no solo funcione correctamente, sino que también ofrezca una experiencia de usuario optimizada [26].

Testing Libraries

Para asegurar la calidad y fiabilidad de la aplicación, se utilizan diversas bibliotecas de pruebas como @testing-library/jest-dom, @testing-library/react, y @testing-library/user-event. Estas herramientas permiten la creación de pruebas unitarias y de integración que validan el correcto funcionamiento de los componentes React y su interacción con el usuario. Las pruebas automatizadas son esenciales para detectar y corregir errores antes de que lleguen a producción, garantizando que la aplicación se mantenga robusta y funcional a lo largo del tiempo.

Componentes del frontend

La aplicación sigue una arquitectura de componentes, donde cada funcionalidad o parte de la interfaz se encapsula en un componente React. Esto permite una mayor modularidad y reusabilidad del código.

El componente principal sería la clase App.js donde se configuran las rutas utilizando React Router. Luego tendríamos otros componentes como Cabecera, que maneja la barra de navegación de la aplicación, Inicio, donde se mostraría la pantalla inicial con una pequeña información introductoria de FIFA, al igual que el modo de juego FUT.

Sin embargo, los componentes más importantes serían Alineaciones, donde se

muestra una tabla de jugadores, donde se puede filtrar y buscar por estadísticas de un jugador, y el componente Análisis, donde se haría el TOPSIS de los criterios a elegir por el usuario.

El componente Análisis, carga y parsea el CSV, luego filtra los datos basados en la posición seleccionada por el usuario. Posteriormente, procesa los datos filtrados usando el método TOPSIS para determinar los mejores jugadores según ciertos criterios. Finalizando mostrando los datos en una tabla y los resultados TOPSIS en otra tabla si están disponibles.

Capítulo 4

CONCLUSIONES

Llegados al final de este TFG, es inevitable mirar el paso del tiempo y mirar con agradecimiento todo el trayecto hecho. En este TFG se ha abordado el desarrollo de un sistema automatizado para la gestión y análisis de datos en el contexto de FUT. A lo largo del proyecto se han alcanzado varios objetivos específicos que han permitido obtener resultados significativos y satisfactorios.

Inicialmente, se realizó una revisión de la literatura relacionada con los videojuegos, la psicología en el entorno de los juegos digitales y las técnicas de recuperación de información web, estableciendo así una base teórica robusta para el diseño y desarrollo del sistema.

El diseño del sistema incluyó la arquitectura del software, el modelado de la base de datos y la interfaz de usuario. Estas etapas fueron fundamentales para garantizar la usabilidad, la eficiencia y la escalabilidad del sistema, asegurando que cumpliera con los requisitos establecidos inicialmente.

Durante la fase de desarrollo, se implementaron técnicas avanzadas de web scraping para la extracción de datos relevantes desde fuentes web, así como la integración de algoritmos como TOPSIS para la toma de decisiones automatizada basada en múltiples criterios.

Se elaboró un manual de usuario detallado que facilita el uso del sistema, así como uno de despliegue.

Gracias al uso de Scrapy, he aprendido a realizar una obtención de datos y me ha permitido ver de primera mano como es hacer web scraping. Personalmente, me ha gustado utilizar React, porque es un framework del que no tenía conocimientos previos y me ha parecido muy innovador e intuitivo.

La posibilidad de aplicar los conocimientos de React a otros proyectos, hace que este proyecto tenga más atractivo para mí, así como el aprendizaje del TOPSIS, que puede aplicarse en otros ámbitos.

Como posibles líneas de desarrollo futuro, se consideran las siguientes áreas:

1. Expansión a Plataformas Móviles: Implementar una versión del sistema utilizando React Native para dispositivos móviles, permitiendo a los usuarios acceder y gestionar datos desde sus teléfonos inteligentes de manera conveniente.
2. Mejora de la Seguridad: Investigar y desarrollar técnicas avanzadas para aumentar la seguridad de los spiders utilizados en el web scraping, mitigando el riesgo de ser detectado y bloqueado por los sistemas de protección de sitios web.
3. Creación de Cuentas de Usuario: Integrar funcionalidades para la creación de cuentas de usuario dentro de la aplicación, proporcionando a los usuarios la capacidad de personalizar y guardar sus configuraciones y preferencias.
4. Análisis Predictivo: Implementar capacidades de análisis predictivo utilizando técnicas de machine learning para prever tendencias futuras en el mercado de jugadores dentro de FIFA Ultimate Team, proporcionando recomendaciones más precisas y personalizadas a los usuarios.
5. Mejoras en la Interfaz de Usuario: Continuar mejorando la interfaz de usuario con características adicionales de visualización y personalización, mejorando así la experiencia del usuario final.
6. Funciones de administrador: Una vez añadida cuentas de usuarios, crear roles. Un administrador podrá cargar y actualizar la información de los jugadores en la base de datos. El administrador accede al sistema de gestión de jugadores. Carga o actualiza los datos de un jugador específico. El flujo secundario estaría en que el administrador puede revisar los cambios antes de confirmar la actualización en la base de datos. También podrían actualizar la introducción al FIFA y otros contenidos informativos. El administrador accede al sistema de gestión de contenidos. Actualiza el contenido informativo, incluyendo la introducción al FIFA.

En conclusión, este TFG no solo ha demostrado la viabilidad de utilizar tecnologías modernas para la gestión de datos en el contexto de los videojuegos, sino que también ha establecido una plataforma robusta para posibles desarrollos futuros que podrían potenciar aún más la utilidad y la funcionalidad del sistema.

Apéndice A

Manual del Usuario

Este manual del usuario proporciona una guía completa para utilizar el Asistente para FUT. Está diseñado para ayudar a los usuarios a comprender y utilizar todas las funcionalidades de la aplicación de manera eficiente.

A.1. Guía de Usuario

A.1.1. Interfaz

La aplicación consta de:

- Pantallas Principales: La aplicación tiene varias pantallas principales, incluyendo la pantalla de inicio, pantalla de ultimate team, pantalla de búsqueda, y la pantalla de análisis.
- Menú principal: El menú principal permite a los usuarios navegar entre las diferentes secciones de la aplicación. Está situado en la parte superior de la pantalla y proporciona acceso rápido a todas las funcionalidades principales.

A.1.2. Instrucciones detalladas para realizar tareas específicas

¿Cómo visualizar la tabla de jugadores?

1. Desde inicio clic a la vista alineaciones
2. Navegue por la tabla libremente

¿Cómo visualizar un criterio de un jugador en concreto de la tabla de jugadores?

1. Desde inicio clicas a la vista alineaciones
2. Seleccione la estadística o criterio a buscar
3. Introduzca el valor deseado
4. Clicas en el botón sugerir jugador

¿Cómo realizo el TOPSIS?

1. Desde inicio clicas a la vista TOPSIS.
2. Filtra la posición que desees o puedes dejarla en todos los jugadores
3. Clicas en el botón run TOPSIS

¿Cómo leer una introducción al ultimate Team?

1. Desde inicio clicas a la vista Ultimate Team.

A.1.3. Ejemplos prácticos

Ejemplo de búsqueda de estadística

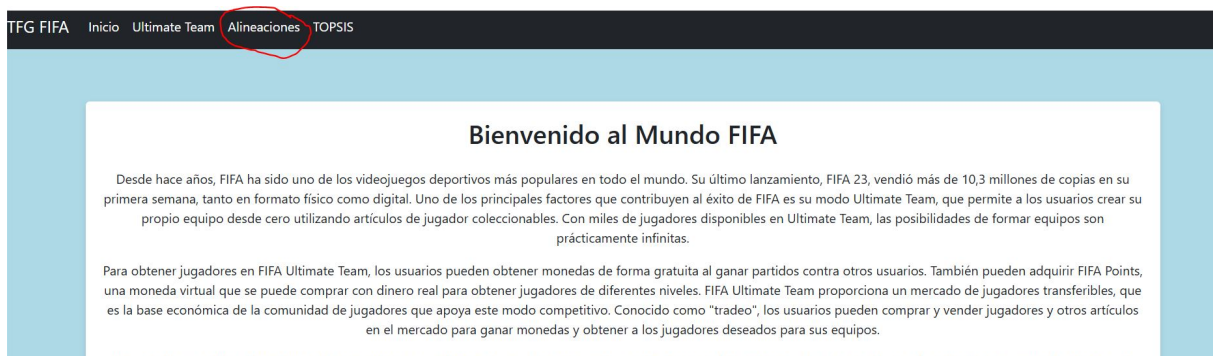


Figura A.1: Búsqueda de estadística 1

Ejemplo de realización de TOPSIS

Selecciona una estadística:

Pace

80

Sugerir Jugador

Name	Rating	Pos	Club	Nation	Liga	Price	Pierna	Pace	Shooting	Passing	Dribbling	Defending	Physicality	Altura	Peso
Karim Benzema	91	CF	Real Madrid	France	LaLiga Santander	25K	4	80	88	83	87	39	78	185cm 6'1"	(81kg)
Robert Lewandowski	91	ST	FC Barcelona	Poland	LaLiga Santander	0	4	75	91	79	86	44	83	185cm 6'1"	(81kg)
Kylian Mbappé	91	ST	Paris SG	France	Ligue 1	70K	4	97	89	80	92	36	76	182cm 6'0"	(73kg)

Figura A.2: Búsqueda de estadística 1.1

Jugadores Sugeridos:

Name	Rating	Pos	Club	Nation	Liga	Price	Pierna	Pace	Shooting	Passing	Dribbling	Defending	Physicality	Altura	Peso
Karim Benzema	91	CF	Real Madrid	France	LaLiga Santander	25K	4	80	88	83	87	39	78	185cm 6'1"	(81kg)
Andrew Robertson	87	LB	Liverpool	Scotland	Premier League	6.9K	2	80	61	81	81	82	76	178cm 5'10"	(64kg)
Riyad Mahrez	86	RW	Manchester City	Algeria	Premier League	10K	4	80	83	81	90	38	60	179cm 5'10"	(67kg)
Romelu Lukaku	86	ST	Inter	Belgium	Serie A TIM	5K	4	80	85	75	77	38	82	191cm 6'3"	(94kg)
Paulo Dybala	86	CF	Roma FC	Argentina	Serie A TIM	8.5K	3	80	85	85	90	40	59	177cm 5'10"	(75kg)

Figura A.3: Búsqueda de estadística 1.2

TFG FIFA Inicio Ultimate Team Alineaciones TOPSIS

Bienvenido al Mundo FIFA

Desde hace años, FIFA ha sido uno de los videojuegos deportivos más populares en todo el mundo. Su último lanzamiento, FIFA 23, vendió más de 10,3 millones de copias en su primera semana, tanto en formato físico como digital. Uno de los principales factores que contribuyen al éxito de FIFA es su modo Ultimate Team, que permite a los usuarios crear su propio equipo desde cero utilizando artículos de jugador coleccionables. Con miles de jugadores disponibles en Ultimate Team, las posibilidades de formar equipos son prácticamente infinitas.

Para obtener jugadores en FIFA Ultimate Team, los usuarios pueden obtener monedas de forma gratuita al ganar partidos contra otros usuarios. También pueden adquirir FIFA Points, una moneda virtual que se puede comprar con dinero real para obtener jugadores de diferentes niveles. FIFA Ultimate Team proporciona un mercado de jugadores transferibles, que es la base económica de la comunidad de jugadores que apoya este modo competitivo. Conocido como "tradeo", los usuarios pueden comprar y vender jugadores y otros artículos en el mercado para ganar monedas y obtener a los jugadores deseados para sus equipos.

Los jugadores en FIFA Ultimate Team tienen diferentes atributos, como velocidad, ritmo, regate, etc., que determinan su nivel. Sin embargo, el hecho de que un jugador tenga una puntuación media alta en sus atributos no siempre significa que sea el mejor jugador. Algunos atributos pueden tener más importancia que otros en determinados momentos, por lo que es importante tener en cuenta este aspecto al comprar un jugador.

Figura A.4: Realización TOPSIS 1

Jugadores

Defensa Lateral Izquierdo (LWB) ▼

Run TOPSIS

Name	Rating	Pos	Club	Nation	Liga	Price	Pierna	Pace	Shooting	Passing	Dribbling	Defending	Physicality	Altura	Peso
Angeliño	83	LWB	TSG Hoffenheim	Spain	Bundesliga	1.2K	3	78	71	82	84	76	70	171cm 5'7"	(69kg)
Ben Chilwell	82	LWB	Chelsea	England	Premier League	750	3	76	60	78	78	78	75	180cm 5'11"	(77kg)
Robin Gosens	82	LWB	Inter	Germany	Serie A TIM	750	3	82	75	74	79	77	80	183cm 6'0"	(76kg)

Figura A.5: Realización TOPSIS 1.1

Jugadores

Defensa Lateral Izquierdo (LWB) ▼

Run TOPSIS **Mostrar Tabla Original**

Resultados TOPSIS

Name	Rating	Pos	Club	Nation	Liga	Price	Pierna	Pace	Shooting	Passing	Dribbling	Defending	Physicality	Altura	Peso
Przemysław Frankowski	76	LWB	RC Lens	Poland	Ligue 1	500	3	92	70	72	74	65	72	175cm 5'9"	(70kg)
Leonardo Spinazzola	82	LWB	Roma FC	Italy	Serie A TIM	850	4	91	64	73	80	78	69	186cm 6'1"	(75kg)
Rico Henry	75	LWB	Brentford	England	Premier League	650	2	90	40	59	74	70	72	170cm 5'7"	(66kg)

Figura A.6: Realización TOPSIS 1.2

Apéndice B

Despliegue del Sistema

Este manual de despliegue proporciona las instrucciones necesarias para instalar, configurar y desplegar el Asistente para FIFA Ultimate Team en un entorno de desarrollo y producción.

B.1. Requisitos del Sistema

B.1.1. Hardware necesario

- Procesador: Intel i3 o superior
- Memoria RAM: 8GB
- Espacio en disco: 500MB

B.1.2. Software necesario

- Sistema Operativo: Windows 10 o superior, macOS 10.15 o superior
- Navegador: Google Chrome, Mozilla Firefox
- Node.js: Versión 14.x o superior (Se recomienda la más actualizada posible)
- npm: Versión 6.x o superior (Se recomienda la más actualizada posible)

B.2. Instalación y configuración

1. Clonar el repositorio del proyecto: Con el comando “git clone <https://github.com/JoseEspinar/TFG>”

2. Navegar al directorio del proyecto: `cd TFG`
3. Instalar las dependencias necesarias: `npm install`
4. Configurar las variables de entorno: Crear un archivo `.env` en el directorio raíz del proyecto. Las variables de entorno son útiles para gestionar configuraciones específicas de diferentes entornos (como desarrollo, pruebas y producción) sin tener que modificar el código fuente.
5. Iniciar la aplicación: `npm start`

B.3. Despliegue en producción

El despliegue en producción es el proceso de preparar y transferir una aplicación desde un entorno de desarrollo a un entorno de producción donde estará accesible para los usuarios finales. Este proceso generalmente implica varios pasos para asegurar que la aplicación funcione de manera eficiente, segura y confiable en un entorno en vivo.

1. Crear el Build de Producción: `npm build`
2. Crear un Repositorio en GitHub: Para ello hay que tener en cuenta en GitHub y hacer un nuevo proyecto.
3. Mueve tu Build a la Carpeta Correcta
4. Habilitar GitHub Pages

B.4. Mantenimiento y actualización

Para actualizar el código y las dependencias, hay que ejecutar los comandos `git pull origin main` y `npm install`. Siempre monitoreando el rendimiento del sistema y realizando optimizaciones según sea necesario.

Bibliografía

- [1] Aruldoss, M., Lakshmi, T. M., and Venkatesan, V. P. (2013). A survey on multi criteria decision making methods and its applications. *American Journal of Information Systems*, 1(1):31–43.
- [2] AXIOS (2024). AXIOS website. <https://axios-http.com/es/docs/intro>. Comprobado en 2024-07-01.
- [3] Canós, J. H., Letelier, P., and Penadés, M. C. (2003). Metodologías ágiles en el desarrollo de software. *Universidad Politécnica de Valencia, Valencia*.
- [4] Chen, P. P.-S. (1976). The entity-relationship model—toward a unified view of data. *ACM transactions on database systems (TODS)*, 1(1):9–36.
- [5] Crawford, G., Gosling, V. K., and Light, B. (2013). *Online gaming in context: The social and cultural significance of online games*. Routledge.
- [6] Cueto, M. J. J. F. and Zuñiga, C. B. (2016). Diagrama de clases en UML. <http://es.scribd.com/doc/31096724/Diagrama-de-Clases-en-UML#scribd>. [Comprobado en 2024-06-12].
- [7] Espinosa, R. S. C. (2017). Gamificación en escenarios educativos. revisando literatura para aclarar conceptos. *Experiencias de gamificación en aulas*, 15:11–17.
- [8] Gackenheim, C. (2015). *Introduction to React*. Apress.
- [9] Gómez del Castillo Segurado, M. T. (2007). Videojuegos y transmisión de valores. *Revista Iberoamericana de Educación*.
- [10] Hwang, C.-L. and Yoon, K. (1981). Methods for multiple attribute decision making. *Multiple attribute decision making: methods and applications a state-of-the-art survey*, pages 58–191.
- [11] Ishizaka, A. and Nemery, P. (2013). *Multi-criteria decision analysis: methods and software*. John Wiley & Sons.
- [12] Jimenez, J. (2015). *Descubre React*, volume 2. Lean Publishing.
- [13] Khder, M. A. (2021). Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing & Its Applications*, 13(3).
- [14] Krotov, V., Johnson, L., and Silva, L. (2020). Tutorial: Legality and ethics of web scraping.

- [15] Laoyan, S. (2022). Toma de decisiones: definición, pasos, tipos y características. <https://asana.com/es/resources/decision-making-process>. Comprobado en 2024-07-02.
- [16] Marín Montín, J. (2014). Videojuegos y televisión. influencias en el tratamiento audiovisual de contenidos deportivos. *LifePlay: revista académica internacional sobre videojuegos*, 3, 35-55.
- [17] Martín, M. and Vílchez, L. (2017). Videojuegos, gamificación y reflexiones éticas. *Cuadernos de ética en clave cotidiana*, 7.
- [18] Marx, M. (2005). Conditional xpath. *ACM Transactions on Database Systems (TODS)*, 30(4):929–959.
- [19] Mitchell, R. (2018). *Web scraping with Python: Collecting more data from the modern web*. O'Reilly Media, Inc.
- [20] PapaParse (2024). PapaParse website. <https://www.papaparse.com/>. Comprobado en 2024-07-01.
- [21] Sahoo, S. K. and Goswami, S. S. (2023). A comprehensive review of multiple criteria decision-making (mcdm) methods: advancements, applications, and future directions. *Decision Making Advances*, 1(1):25–48.
- [22] Trigás Gallego, M. (2012). Metodología SCRUM.
- [23] Vaamonde, A. G.-N., Toribio, M. J., Molero, B. T., and Suárez, A. (2018). Beneficios cognitivos, psicológicos y personales del uso de los videojuegos y esports: una revisión. *Revista de Psicología Aplicada al Deporte y al Ejercicio Físico*, 3(2):1–14.
- [24] Vilar, N. (2021). Storyboard y una narrativa gráfica y literaria. *Actas de Diseño*, (36).
- [25] Weadon, D. V., Sánchez, M. F., and Azpilicueta, P. C. (2005). Recuperación de información en la web semántica. *Novática: Revista de la Asociación de Técnicos de Informática*, (178):12–15.
- [26] Wehner, N., Seufert, M., Schatz, R., and Hoßfeld, T. (2023). Do you agree? contrasting google's core web vitals and the impact of cookie consent banners with actual web qoe. *Quality and User Experience*, 8(1):5.
- [27] Woodcock, J. and Johnson, M. R. (2021). Live streamers on twitch. tv as social media influencers: Chances and challenges for strategic communication. In *Social media influencers in strategic communication*, pages 88–102. Routledge.
- [28] Zhao, B. (2017). Web scraping. *Encyclopedia of big data*, 1.

