Universidad Autónoma de Madrid

Escuela Politécnica Superior

24|25

Trabajo fin de grado

Scouting y predicción de resultados en ligas semiprofesionales de baloncesto

Álvaro Simón del Corral

Escuela Politécnica Superior
Universidad Autónoma de Madrid
C\Francisco Tomás y Valiente nº 11

Campus Internacional excelencia UAM CSIC+

www.uam.es

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Scouting y predicción de resultados en ligas semiprofesionales de baloncesto

Autor: Álvaro Simón del Corral
Tutor: Alejandro Bellogín Kouki

junio 2025

Álvaro Simón del Corral
Scouting y predicción de resultados en ligas
semiprofesionales de baloncesto

Álvaro Simón del Corral
C\ Francisco Tomás y Valiente Nº 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

A mi familia

No puedes descubrir nuevos océanos si no
tienes el coraje de perder de vista la orilla.

André Paul Guillaume Gide

Resumen

El baloncesto semiprofesional enfrenta limitaciones en cuanto al acceso a datos estructurados y herramientas tecnológicas avanzadas, lo que dificulta el análisis profundo y sistemático del rendimiento deportivo. Este proyecto responde a esta necesidad desarrollando un sistema integral de *scouting* y predicción de resultados orientado específicamente a este tipo de competiciones.

Para ello, se emplean técnicas de *scraping* para extraer datos de la `Federación Española de Baloncesto` y construir una base de datos adaptada a las necesidades del análisis técnico y estratégico. Sobre esta infraestructura, se ha desarrollado una aplicación web que permite visualizar información clave de equipos y jugadores, y realizar predicciones de resultados mediante una fórmula fija diseñada especialmente para este contexto.

Los resultados obtenidos demuestran que la integración de estas herramientas puede suponer un soporte útil y valioso para la toma de decisiones. Entrenadores y analistas de estas categorías podrían tener acceso de forma clara y estructurada a representaciones de información altamente relevantes. Esto supone un avance significativo en la profesionalización del análisis deportivo y en la optimización del rendimiento de jugadores y equipos a estos niveles.

Como resultado, se presenta una plataforma tecnológica funcional que no solo mejora las capacidades de análisis y predicción, sino que también establece una base sólida para futuras mejoras y ampliaciones en el ámbito semiprofesional de este deporte. Su diseño modular y su enfoque adaptable abren una amplia gama de posibilidades de evolución.

Palabras clave

Scouting, predicción, aprendizaje automático, aplicación web, scraping

Abstract

Semi-professional basketball challenges obstacles associated with the access to structured data of the games and other advanced technological tools, making extremely complicated the extended analysis of the performance of athletes or the team. This project answers the necessity of developing an integrated system for scouting and prediction focusing only on these competitions.

For this purpose, we operate with scraping techniques in order to obtain data from the `Federación Española de Baloncesto` and build a database adjusted to the necessities of the technical and strategic analysis. A web-based application originated from this architecture allows visualizing important information for the teams and its members, creating predictions based on the results of a standardized formula designed specifically for this context.

The obtained results proved that integrating these tools can be a valuable and useful mechanism in the decision-making process. Coaches and analysts could have access to a structured and simplified kind of data with beneficial and helpful information, making this research being significant for the development in the professionalization of sports analysis and optimizing the efficiency of players and teams for this specific level.

As a result, this project presents a functional platform that not only enhances the capacities of the analysis of data and prediction, but also establishes a solid foundation for the future improvement regarding the semi-professional scope of this sport. Its modular design and adaptable approach opens a spectrum full of possibilities for evolution.

Keywords

Scouting, prediction, machine learning, web application, scraping

Índice

Listas

Lista de ecuaciones

Lista de figuras

Lista de tablas

1

Introducción

La aplicación de la ciencia de datos en el deporte ha supuesto un antes y un después, transformando radicalmente la manera en la que este se concibe. La significativa mejora del juego arraigada a estos cambios es innegable, y su enorme y creciente influencia social y económica en la población global la sitúa entre una de las principales áreas de interés a explotar. Este campo, actualmente en auge, está alcanzando metas y objetivos hasta hace un tiempo impensables, lo que permite vislumbrar un futuro prometedor [2].

En los deportes de equipo no solo las estrategias tienen un papel fundamental en el juego; de igual forma, la dinámica de cada uno de los jugadores, junto a su conexión y compenetración, sumado a la dirección del cuerpo técnico conforman las bases para lograr el éxito. Por este motivo, es necesario que todos los componentes de este sistema se coordinen y trabajen adecuadamente entre sí para obtener el mejor resultado posible.

En concreto, esta optimización del rendimiento y de las tácticas del juego se aprecia notablemente en el baloncesto, un deporte que precisa de muchas decisiones en tiempo real durante intervalos muy breves. La rapidez, la exigencia y la precisión constituyen su principal desafío. A través de herramientas de estadística avanzada, y el extenso conocimiento que pueden proporcionar, se pueden tomar mejores decisiones que permiten interpretar y predecir el baloncesto de una manera nunca antes contemplada [3].

1.1. Motivación

El uso de la estadística avanzada en el baloncesto ha implicado una revolución en todos los sentidos; empleándose en el rendimiento individual y colectivo, la negociación de contratos, las planificaciones estratégicas, la reducción de lesiones, etc. Un ejemplo de ello es la obtención de datos sobre incidencias y factores de riesgo de lesiones, cargas de juego y predicción del rendimiento futuro de jugadores [4].

En este contexto, Estados Unidos se posiciona a la vanguardia gracias a la incorporación de estas tecnologías en la NBA desde hace varias décadas [5]. Unas tecnologías que han sido posteriormente

adoptadas por el resto del mundo, en donde ya prácticamente se han digitalizado todos los datos, incluidos los de las ligas no profesionales y las categorías base. La magnitud del avance es tal que ha derivado en el crecimiento exponencial de usuarios que consumen este tipo de información de multitud de plataformas destinadas a la muestra de datos y estadísticas baloncestísticas.

La mayoría de estos datos son únicamente accesibles para clubes de ligas profesionales. Frente a la alta demanda exigida en competiciones semiprofesionales surge la necesidad de trasladar los recursos estadísticos a dichas ligas para permitir que jugadores, entrenadores y aficionados puedan tener a su disposición todo este contenido para el scouting, las predicciones o el entretenimiento [6].

1.2. Objetivos

El objetivo principal de este trabajo es desarrollar un sistema que ofrezca predicciones de resultados de futuros partidos y que permita analizar el rendimiento de los jugadores y los equipos en los diversos aspectos del juego para la preparación de partidos y sus respectivas estrategias, futuros fichajes, o simplemente la obtención de información. De esta forma se pretende facilitar el acceso a estadísticas avanzadas de ligas semiprofesionales de baloncesto centralizándolas en una única plataforma web, para que equipos y seguidores puedan explotar su potencial.

A través de la recopilación de datos, y su respectivo análisis descriptivo, se pretende poder generar gráficos, estadísticas, rankings en base a estadísticas y comparaciones de jugadores y equipos, buscar perfiles entre los jugadores de la competición, obtener la clasificación de la liga, extraer información completa de partidos disputados o realizar pronósticos de futuros enfrentamientos.

1.3. Estructura

Este documento se compone de cinco capítulos:

Capítulo 1. Introducción: Se presenta el tema del proyecto, la motivación que lo impulsa, los objetivos que se procuran cumplir y la estructura del presente documento.

Capítulo 2. Estado del Arte: Se detalla el contexto tecnológico en el que se lleva a cabo el sistema.

Capítulo 3. Diseño e Implementación: Se esclarecen las decisiones técnicas de diseño y los aspectos clave de desarrollo e implementación para la elaboración del sistema.

Capítulo 4. Pruebas y Resultados: Se detallan las pruebas realizadas y se analizan sus resultados en base a los objetivos definidos.

Capítulo 5. Conclusiones y Trabajo futuro: Se exponen las conclusiones extraídas del proyecto, así como posibles futuras líneas de trabajo para mejoras o ampliaciones.

2

Estado del arte

Este capítulo presenta una revisión de los principales procesos y herramientas tecnológicas empleados en el proyecto, con el fin de justificar las decisiones adoptadas en el mismo. Se describe el procedimiento de recopilación de datos, así como las tecnologías web y los submódulos correspondientes al scouting y a la predicción utilizados en la aplicación web del sistema.

2.1. Recopilación de datos

La recopilación de datos integra las actividades de recolección de toda la información y las estadísticas necesarias para el posterior procesamiento, análisis, interpretación y representación de los mismos. Para ello, primero se ha de escoger la plataforma cuyos datos se adecuen mejor a la idea concebida del sistema a desarrollar. Seguidamente se procede con la obtención de dichos datos de forma automatizada empleando técnicas de scraping.

2.1.1. Selección de la plataforma

Dado que la NBA es pionera también a nivel estadístico en el mundo baloncestístico, inicialmente se utilizaron sus datos como referencia para ampliar los conocimientos en este ámbito y observar el funcionamiento en un entorno tan sumamente profesionalizado. Toda esta información oficial se recoge en su página web [7], pero también puede accederse a gran parte de la misma desde una librería Python, desarrollada hace unos años, llamada nba_api [8]. No obstante, tal y como se esperaba, pese a la gran variedad de estadísticas que proporcionan, existen muchas otras que no se encuentran disponibles públicamente en los recursos mencionados. Esta situación se produce también en el resto de ligas profesionales del mundo, cuyas estadísticas cada vez se asemejan más a las de la renombrada NBA [9, 10].

A continuación, se procedió a investigar las diferentes plataformas disponibles de seguimiento de estadísticas de baloncesto en tiempo real en España. Entre ellas cabe destacar dos de las más conocidas y utilizadas:

- **Swish:** Una aplicación que fue creada en el 2019 por la empresa tecnológica española NBN23 y que cubre las estadísticas de categorías inferiores, no profesionales y algunas semiprofesionales [11].
- **Página web de la Federación Española de Baloncesto:** Un sitio web cuyas estadísticas, manejadas también por NBN23, pertenecen única y exclusivamente a competiciones nacionales como son las ligas profesionales y semiprofesionales o los campeonatos de España de ligas inferiores [12].

Swish cuenta con una mejor y mayor representación de datos y con funcionalidades adicionales, como las crónicas generadas por Inteligencia Artificial o las previas de los partidos. Sin embargo, al ser una aplicación que actualmente ofrece mucha funcionalidad bajo suscripción o pago, resulta inviable como medio de extracción de datos. En consecuencia, la plataforma de la que se han recabado los datos de ligas semiprofesionales ha sido la de la Federación Española de Baloncesto, que además cuenta con una API.

2.1.2. Scraping

El scraping consiste en la obtención automatizada de datos de plataformas [13]. En este proyecto, dicha tarea se lleva a cabo mediante un script desarrollado en Python que recoge la información imprescindible para alcanzar los objetivos del sistema. El script hace uso de cinco librerías:

- **requests:** Sirve para realizar peticiones HTTP [14].
- **pandas:** Permite la manipulación de datos estructurados [15].
- **pdfplumber:** Se emplea para la extracción del contenido de los archivos PDF [16]. Al principio se probaron otras alternativas como PyPDF2 [17] o camelot [18], pero la única capaz de procesar correctamente tablas complejas y extensas tras ajustar sus parámetros de lectura fue esta librería.
- **pdfminer:** Se incluye para la lectura de la información de una tabla con columnas combinadas de los PDF que pdfplumber no es capaz de recuperar con precisión [19].
- **pickle:** Se utiliza para serializar los objetos en los que se integran los datos leídos, facilitando su almacenamiento y su rápida reutilización en un futuro [20].

La recopilación de datos de la página de la Federación Española de Baloncesto se hace sobre la categoría semiprofesional más antigua a nivel nacional, la ahora denominada Tercera FEB [1].

2.2. Tecnologías web

La arquitectura del sistema para el scouting y la predicción de resultados puede dividirse en dos capas principales: backend y frontend. Ambos componentes se comunican mediante peticiones HTTP a una API REST implementada con Django REST Framework [21], lo que garantiza el desacoplamiento entre servicios, una mayor flexibilidad y una mejor escalabilidad.

2.2.1. Backend

El backend, responsable de la lógica de negocio del sistema, se ha desarrollado con Django [22], un framework de alto nivel para Python. Basado en el patrón Modelo-Vista-Controlador, Django, proporciona herramientas integradas para la gestión de bases de datos, la autenticación, el enrutamiento y la creación de APIs RESTful. Su amplia flexibilidad para trabajar con distintos sistemas de gestión de bases de datos, como PostgreSQL [23], MySQL [24] o SQLite [25], permite adaptar la implementación a diferentes entornos y necesidades del proyecto. En conclusión, Django constituye una opción robusta, rápida, segura y escalable para el desarrollo de aplicaciones web modernas.

2.2.2. Frontend

El frontend, encargado de gestionar la interfaz del sistema, se ha implementado con Angular [26], un framework especializado en el desarrollo y el mantenimiento de aplicaciones web. Aunque anteriormente se había trabajado con Node.js [27], finalmente se optó por Angular con el objetivo de explorar y aprender un nuevo entorno de programación, asumiendo el reto que ello suponía. La interfaz permite el acceso visual e interactivo a las estadísticas del proceso de scouting y a la funcionalidad de predicción de resultados.

2.3. Scouting

El scouting en baloncesto es el proceso sistemático de análisis y evaluación de datos y estadísticas de jugadores y equipos para la toma de decisiones técnicas y tácticas [28]. Su propósito principal es adquirir información precisa y relevante que contribuya en:

- **La preparación de los partidos:** Antes de cada encuentro los equipos estudian a sus rivales e identifican sus fortalezas y debilidades, a nivel individual y colectivo, para diseñar las estrategias o el llamado *planning* de partido.
- **La mejora de los jugadores y del equipo:** Los cuerpos técnicos detectan los puntos fuertes y débiles de sus propios jugadores para potenciarlos o corregirlos.

- **La búsqueda de jugadores:** Los equipos buscan nuevos jugadores seleccionando aquellos perfiles que mejor se ajustan a sus necesidades específicas y a su sistema de juego. Por este motivo, se basan en sus estadísticas, su estilo de juego y su comportamiento en diferentes situaciones.

Para adentrarse en el mundo del scouting es importante conocer el significado de algunos términos fundamentales:

- **Racha:** Es el número de partidos consecutivos que un equipo ha ganado o perdido.
- **+/- o balance:** Mide el impacto de un jugador calculando la diferencia entre los puntos anotados y los recibidos por el equipo cuando él está jugando.
- **Valoración:** Evalúa el rendimiento de un jugador sumando sus estadísticas positivas y restando sus negativas. Las estadísticas positivas son puntos, rebotes, asistencias, robos, faltas recibidas y tapones cometidos; mientras que las negativas son tiros fallados, pérdidas de balón, faltas cometidas y tapones recibidos.

La representación del scouting se puede efectuar a través de gráficos, tablas, comparaciones y mapas de calor. Este tipo de recursos visuales facilitan la interpretación de datos complejos y contribuyen a una mejor toma de decisiones.

2.4. Predicción

En esta sección se profundiza en sistemas de predicción y en métricas utilizadas para evaluar su rendimiento.

2.4.1. Sistemas de predicción

Los sistemas de predicción constituyen un conjunto de técnicas y metodologías orientadas a la estimación de comportamientos o eventos futuros mediante el análisis de datos [29]. Estos sistemas se sustentan en el aprendizaje automático o *machine learning*, una rama de la inteligencia artificial que posibilita la construcción de modelos capaces de identificar patrones, aprender de los datos y realizar inferencias de forma autónoma [30]. Dentro del aprendizaje automático, se pueden diferenciar tres tipos:

- **Aprendizaje supervisado:** El modelo se entrena con un conjunto de datos etiquetados, es decir, con ejemplos que incluyen los datos de entrada y su salida correcta. Su objetivo es poder predecir la salida correcta de nuevos datos de entrada en base a patrones aprendidos durante el entrenamiento.

- **Aprendizaje no supervisado:** En este enfoque el modelo se entrena sin datos etiquetados, lo que le permite encontrar patrones y relaciones implícitas en los datos sin ninguna clase de intervención humana directa.
- **Aprendizaje por refuerzo:** El modelo se entrena para tomar decisiones en función de la retroalimentación recibida por medio de la interacción con el entorno.

Para predecir el ganador de partidos de baloncesto utilizando datos de encuentros anteriores y sus resultados —es decir, datos etiquetados— se emplea aprendizaje supervisado. Determinar el ganador es una variable categórica con dos posibles valores; por ello, los algoritmos supervisados seleccionados para evaluar su efectividad en estas predicciones son concretamente de clasificación binaria. Estos modelos de aprendizaje automático se implementan con la librería scikit-learn [31] de Python.

A continuación, se detallan los algoritmos escogidos para este propósito:

- **Regresión logística:** Calcula la probabilidad de que un evento pertenezca a una clase específica usando la función sigmoide como base del cálculo [32].
- **K-vecinos más cercanos o KNN:** Asigna una clase a un dato nuevo basándose en la clase mayoritaria de sus k vecinos más próximos en el espacio de características [33].
- **Árbol de decisión:** Crea un modelo con estructura de árbol que divide los datos en subconjuntos a partir de preguntas binarias sucesivas sobre sus características. Al alcanzar un nodo hoja, se asigna una clase final al dato [34].
- **Bosques aleatorios:** Se utilizan múltiples árboles de decisión entrenados con distintos subconjuntos de datos para combinar sus resultados, mejorando la precisión y reduciendo el sobreajuste [35].
- **Refuerzo de gradiente:** Genera una secuencia de modelos en la que cada modelo nuevo corrige los errores de los anteriores para mejorar progresivamente la precisión del conjunto [36].
- **Redes neuronales:** Son estructuras formadas por capas de nodos interconectados capaces de aprender patrones complejos y no lineales. Su diseño se inspira en la estructura del cerebro humano [37].

2.4.2. Métricas de predicción

Las métricas de predicción permiten cuantificar la calidad de las predicciones en función de los aciertos y los errores cometidos por un modelo [38]. En problemas de clasificación binaria, se suelen utilizar las siguientes:

- **Confusion matrix:** Es una tabla que compara las predicciones del modelo con los valores reales. Está compuesta por:

- **TP:** Casos positivos correctamente clasificados como tal.
- **TN:** Casos negativos correctamente clasificados como tal.
- **FP:** Casos negativos clasificados erróneamente como positivos.
- **FN:** Casos positivos clasificados erróneamente como negativos.

- **Accuracy:** Mide la proporción de predicciones correctas, tanto positivas como negativas, sobre el total.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

- **Precision:** Mide cuántos de los ejemplos predichos como positivos realmente lo son.

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

- **Recall:** Mide cuántos de los ejemplos positivos reales fueron correctamente identificados.

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

- **F1 score:** Es la media armónica de precision y recall.

$$F1score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.4}$$

3

Diseño e Implementación

A lo largo de este capítulo se abordan las decisiones tomadas a nivel de diseño del sistema, incluyendo diagramas que ilustran la arquitectura general y requisitos funcionales y no funcionales que definen el comportamiento y las características del proyecto. Asimismo, se describen las principales estrategias de implementación asumidas durante el desarrollo. De este modo, se ofrece una visión global y detallada de la aplicación web que abarca desde su concepción teórica hasta su realización práctica.

3.1. Diseño

En la Figura 3.1 se presenta el diagrama de arquitectura del sistema, una representación visual de sus componentes y de las relaciones entre ellos. Este diagrama facilita la comprensión de la organización interna del proyecto y muestra cómo interactúan los módulos garantizando escalabilidad, mantenibilidad y un bajo nivel de acoplamiento. Como se puede apreciar, el sistema se estructura en tres elementos principales:

- **Scraping:** Es el responsable de la extracción automatizada de los datos y de suministrar dicha información a la aplicación web.
- **Aplicación web:** Es el núcleo central del sistema y está compuesto por el frontend y el backend. Dentro del frontend se integran las funcionalidades de scouting y predicción de resultados.
- **Usuario:** Interactúa directamente con la aplicación web.

Cabe destacar que en cada componente se especifican las herramientas tecnológicas empleadas en sus respectivas implementaciones, proporcionando un esquema visual que refleja las decisiones técnicas adoptadas.

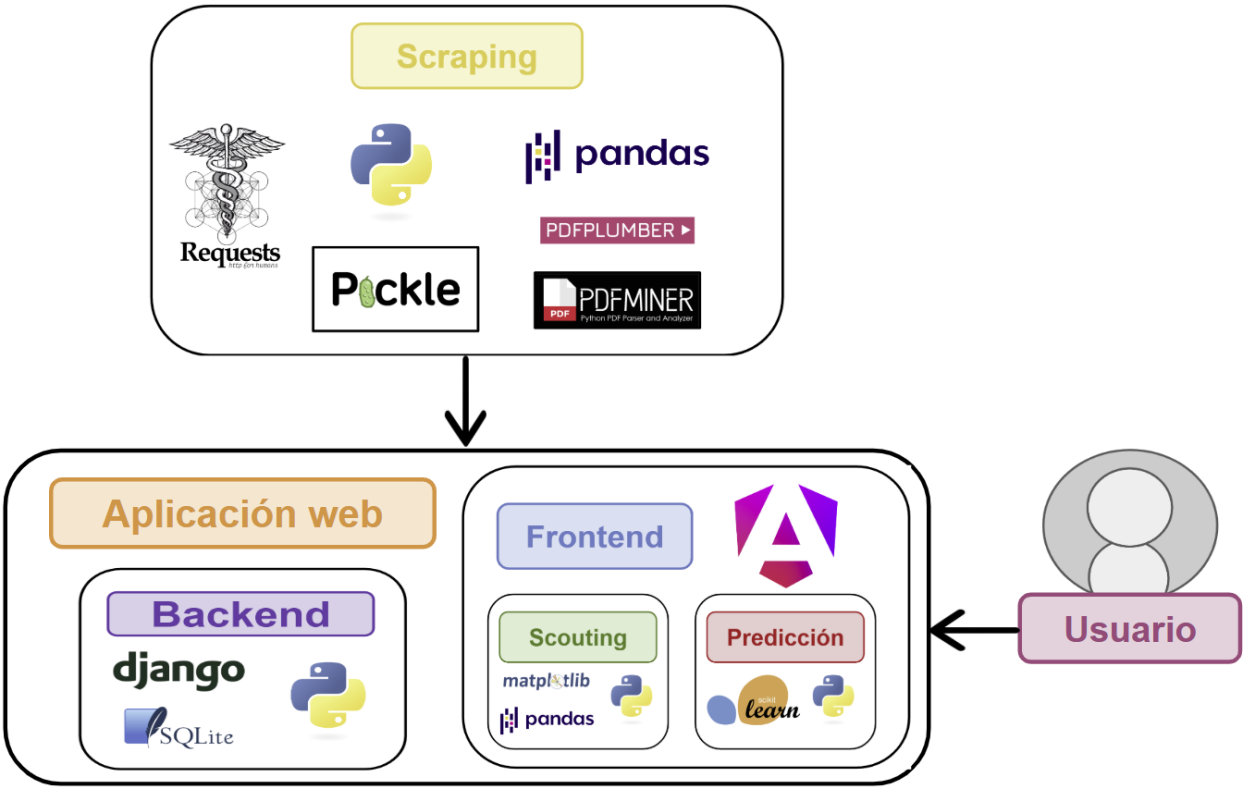

Figura 3.1: Diagrama de arquitectura del proyecto

3.1.1. Requisitos funcionales

Los requisitos funcionales se clasifican en cuatro categorías: los relacionados con el conjunto de datos, el submódulo de scouting, el submódulo de predicción y la interfaz de usuario.

Requisitos funcionales del conjunto de datos

RF-1.- El conjunto de datos se debe obtener de la página web de la Federación Española de Baloncesto mediante técnicas de scraping.

RF-2.- Los datos extraídos deben incluir información detallada sobre jugadores, partidos y equipos pertenecientes a una misma competición, garantizando la integridad y la coherencia de la información recopilada.

RF-3.- Debe poder realizarse extracción de contenido desde documentos PDF.

RF-4.- El procesamiento y manipulación de los datos se tiene que llevar a cabo con estructuras DataFrame de la librería pandas.

RF-5.- El sistema almacenará los datos procesados en archivos Pickle, para permitir un acceso rápido y eficiente en operaciones futuras.

RF-6.- El sistema permitirá la actualización periódica de los datos a través de la incorporación de los

identificadores de los partidos nuevos y la modificación del bearer token en el fichero de configuración del proyecto.

RF-7.- El sistema debe validar los datos recogidos antes de su almacenamiento para evitar errores o datos incompletos.

RF-8.- El sistema debe manejar adecuadamente errores y excepciones durante la extracción.

RF-9.- El sistema almacenará logs que permitan seguir el proceso de scraping para depurar posibles problemas y comprobar que se obtienen correctamente todos los partidos.

Requisitos funcionales del submódulo de scouting

RF-10.- El contenido del scouting se generará después del scraping con los datos guardados en el archivo Pickle.

RF-11.- Toda la información del scouting, a excepción de la comparación entre jugadores, se debe crear antes de su uso por la aplicación (no será generado dinámicamente).

RF-12.- Los gráficos, tablas, comparaciones y mapas de calor del scouting se deben elaborar con las librerías pandas y matplotlib.

RF-13.- El sistema debe permitir visualizar la clasificación actualizada de la liga.

RF-14.- El usuario podrá consultar los rankings de equipos y de jugadores filtrándolos según la estadística deseada.

RF-15.- El sistema permitirá ver estadísticas generales de equipos y jugadores.

RF-16.- El usuario podrá comparar estadísticas entre equipos o entre jugadores de forma clara y diferenciada.

RF-17.- El sistema permitirá la búsqueda de jugadores aplicando filtros para obtener el perfil que mejor se adecúe a unas estadísticas concretas.

RF-18.- El usuario podrá acceder a información completa y detallada de los partidos que ya se han disputado.

Requisitos funcionales del submódulo de predicción

RF-19.- El sistema ofrecerá una predicción sobre el ganador de un partido aplicando una fórmula fija basada en estadísticas acumuladas de los equipos implicados.

RF-20.- El sistema mostrará los resultados de la fórmula para cada equipo.

RF-21.- El sistema indicará el nivel de probabilidad de acierto de la predicción realizada, clasificándolo en baja, media o alta.

RF-22.- Toda la información de las predicciones se debe crear antes de su uso por la aplicación (no

será generado dinámicamente).

Requisitos funcionales de la interfaz de usuario

RF-23.- La aplicación web debe disponer de una página principal desde la cual el usuario podrá acceder a todas las funcionalidades del sistema mediante dos botones principales: uno para el submódulo de scouting y otro para el submódulo de predicción.

RF-24.- El usuario podrá seleccionar jugadores, equipos o partidos a evaluar a través de selectores desplegables en la interfaz, de manera que constituya una navegación intuitiva y personalizada de los datos.

RF-25.- La navegación por el sistema se realizará con botones distribuidos de forma coherente y visualmente clara.

3.1.2. Requisitos no funcionales

RNF-1.- El proyecto se ejecutará en un entorno virtual para gestionar las dependencias y contribuir a la portabilidad y consistencia del mismo.

RNF-2.- El sistema está compuesto por un backend desarrollado en Django siguiendo el patrón MVT y un frontend desacoplado implementado con Angular. La comunicación entre ambos se debe realizar a través de una API REST.

RNF-3.- El sistema debe ser modular y fácil de mantener, facilitando la actualización o sustitución de sus componentes sin afectar al resto del sistema.

RNF-4.- La interfaz será usable y sencilla. Para ello, en las tablas complejas se tendrá que incluir una ayuda de usuario que explique su funcionamiento.

RNF-5.- El diseño del frontend debe ser *responsive*, garantizando una correcta visualización y funcionamiento en pantallas con tamaños distintos.

RNF-6.- El tiempo de respuesta de la aplicación debe ser inferior a 3 segundos en las operaciones habituales, asegurando una experiencia de usuario fluida y eficiente.

RNF-7.- El desarrollo y las pruebas de la aplicación se podrán realizar en el sistema operativo Windows, aunque el sistema está diseñado para ser portable y funcionar correctamente en otros entornos compatibles con Django y Angular.

RNF-8.- El sistema debe ser capaz de manejar grandes volúmenes de datos y archivos, manteniendo un rendimiento adecuado que no afecte negativamente a la experiencia del usuario.

RNF-9.- Se deben realizar *backups* o copias de seguridad periódicas de la base de datos y del sistema para prevenir la pérdida de información relevante.

3.1.3. Estructura de la aplicación

Tal y como se explica al comienzo del apartado de diseño del presente documento, la aplicación ha sido concebida con una arquitectura modular, en la que cada componente se encuentra claramente delimitado y cumple una función específica dentro del conjunto del sistema. Esta organización estructural facilita la incorporación de nuevas funcionalidades, así como la sustitución o modificación de las ya existentes, sin afectar negativamente al resto del proyecto. Además, este enfoque contribuye significativamente a la realización de pruebas independientes y exhaustivas para la detección temprana y aislada de errores.

A continuación, se detallan los principales aspectos de diseño de los módulos de scraping y de la aplicación web, siguiendo el orden en el que fueron creados. Es importante aclarar que el módulo de usuario no se considera un módulo de software con diseño propio dentro del sistema, sino que representa a un actor externo que interactúa con la aplicación web. Por esa razón, no se tiene en cuenta en las decisiones de este ni de los demás capítulos.

- **Módulo de scraping:** Este componente es el encargado de obtener toda la información necesaria de la página web de la Federación Española de Baloncesto mediante técnicas de scraping. El contenido extraído se estructura en objetos definidos y se almacena de forma persistente en un archivo serializado utilizando la librería pickle, lo que permite su posterior reutilización sin necesidad de repetir el proceso de recolección. En la Figura 3.2 se muestra un diagrama de clases simplificado que ilustra estos objetos y las relaciones existentes entre ellos.

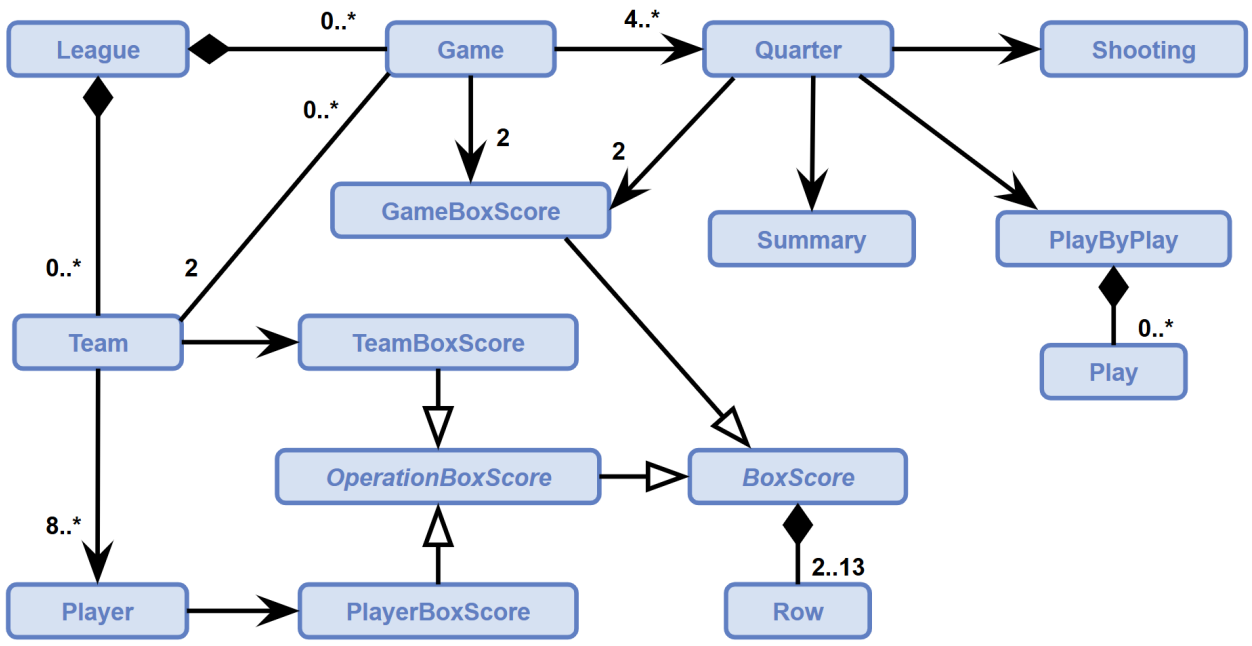

Figura 3.2: Diagrama de clases simplificado del contenido del scraping

Una League está compuesta por varios Teams y Games. Por un lado, cada Team participa en varios Games y está formado por una plantilla de entre 8 y 12 Players. Tanto los Teams como los Players cuentan con 2 Rows de BoxScores; una que refleja los totales acumulados y otra los promedios. Por otro lado, cada Game involucra a 2 Teams, se divide en 4 o más Quarters y tiene 2 BoxScores asociados, correspondientes a los de los equipos local y visitante. Cada Quarter cuenta con 2 BoxScores, un Summary o resumen, un Shooting o datos de tiros y un PlayByPlay o jugada a jugada asociados. Finalmente, dado que un BoxScore presenta distintos atributos en sus Rows dependiendo de si se trata de un Game, un Team o un Player, se define como una clase abstracta base de la que heredan GameBoxScore y otra clase abstracta intermedia denominada OperationBoxScore. Como TeamBoxScore y PlayerBoxScore comparten más atributos entre sí que con GameBoxScore, ambas extienden directamente de OperationBoxScore, potenciando así una mayor abstracción y reutilización de código.

- **Módulo de la aplicación web:** Constituye la capa principal de interacción del sistema con el usuario. Este componente proporciona representaciones visuales de los datos estructurados creados en el módulo anterior, tales como tablas, diversos tipos de gráficos, comparaciones, mapas de calor y otros recursos interactivos. A través de su interfaz, el usuario puede acceder a todas estas posibilidades de manera sencilla e intuitiva. Con el fin de separar las dos funcionalidades principales que ofrece este sistema, la aplicación web se divide en dos submódulos:
  - **Submódulo de scouting:** Permite observar y analizar detalladamente los datos recogidos sobre la competición. Entre las funcionalidades que el sistema pone a disposición del usuario a través de este submódulo resaltan la clasificación actualizada de la liga, la visualización de rankings en función de las estadísticas de equipos y jugadores, el acceso a estadísticas generales y comparativas tanto de jugadores como de equipos, o la información completa y estructurada de los partidos que ya han sido disputados. Igualmente, es posible realizar búsquedas aplicando filtros avanzados para identificar jugadores que se ajusten a un perfil estadístico determinado.
  - **Submódulo de predicción:** Es el responsable de gestionar la generación y representación de las predicciones sobre los equipos ganadores de los partidos. El sistema calcula un valor numérico para cada equipo involucrado aplicando una fórmula fija diseñada con diversos parámetros estadísticos cuyo valor, mostrado en la aplicación, varía en función de los equipos enfrentados. Junto al equipo ganador de la predicción se indica también cuál es la probabilidad de acierto de la misma según cuánto difieran las predicciones calculadas para ambos equipos: baja, media o alta.

Para concluir el diseño del módulo de la aplicación web, en la Figura 3.3 se presenta un diagrama de secuencia que ilustra el flujo general de la aplicación desde la perspectiva del usuario, comprendiendo la interacción con las pantallas, el procesamiento interno y la visualización de resultados. Con este diagrama se muestra un ejemplo de cómo trabajan entre sí los distintos componentes del sistema durante el tiempo de ejecución desde el momento en que el usuario accede a la página principal hasta que completa una tarea determinada.

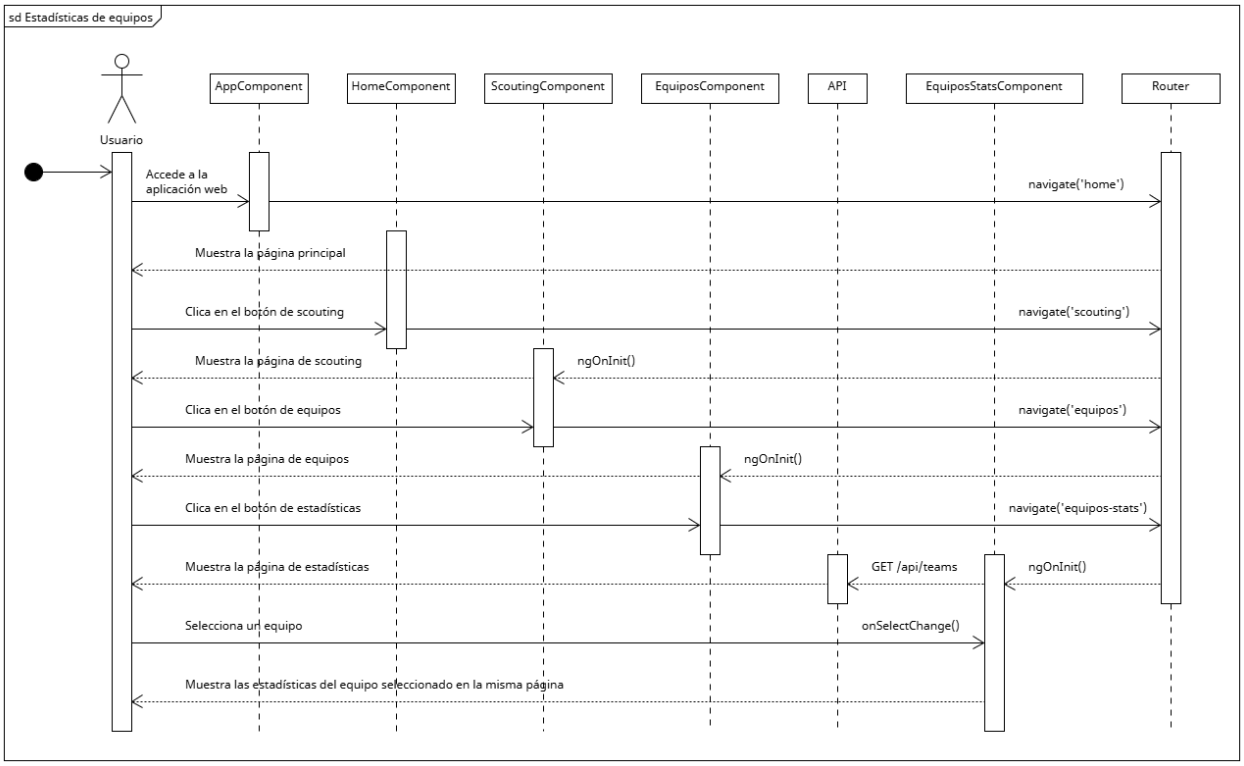

Figura 3.3: Diagrama de secuencia de las estadísticas de los equipos

El diagrama de secuencia desarrollado ilustra el proceso de visualización de las estadísticas de un equipo específico desde que el usuario accede a la aplicación. En primer lugar, al iniciar la navegación desde la vista principal, el usuario puede desplazarse por las distintas secciones funcionales del sistema a través de la interfaz. En el caso concreto de las estadísticas de equipos, la interacción sigue una secuencia que atraviesa los módulos de scouting, equipos y, finalmente, estadísticas generales de equipos. Cada uno de estos módulos depende funcionalmente del anterior, y se encuentran enlazados mediante una lógica de navegación que facilita el intercambio y la consulta de datos. Dicha dinámica refleja la estructura modular del sistema y cómo los componentes colaboran entre sí para proporcionar la información solicitada. Esta misma lógica de navegación y dependencia se aplica en el resto de funcionalidades de la aplicación web.

3.1.4. Ciclo de vida y metodologías de desarrollo

Dentro del diseño del sistema, es fundamental considerar tanto el ciclo de vida como las metodologías de desarrollo del software empleadas para asegurar la organización, la eficiencia y la escalabilidad. El ciclo de vida define las etapas que atraviesan todos los componentes del proyecto desde su creación hasta su finalización o mejora; mientras que las metodologías estructuran los procesos de desarrollo, diseño e implementación mediante la planificación y ejecución de las diferentes funcionalidades.

Para este proyecto se ha seleccionado un modelo de ciclo de vida iterativo en el que se establecen ciclos recurrentes que abarcan desde la fase de pruebas hasta la de análisis. Esta estrategia permite revisar y mejorar fases previas del sistema en función de los resultados obtenidos en etapas posteriores. En la Figura 3.4 se puede apreciar una representación esquemática de este ciclo de vida:

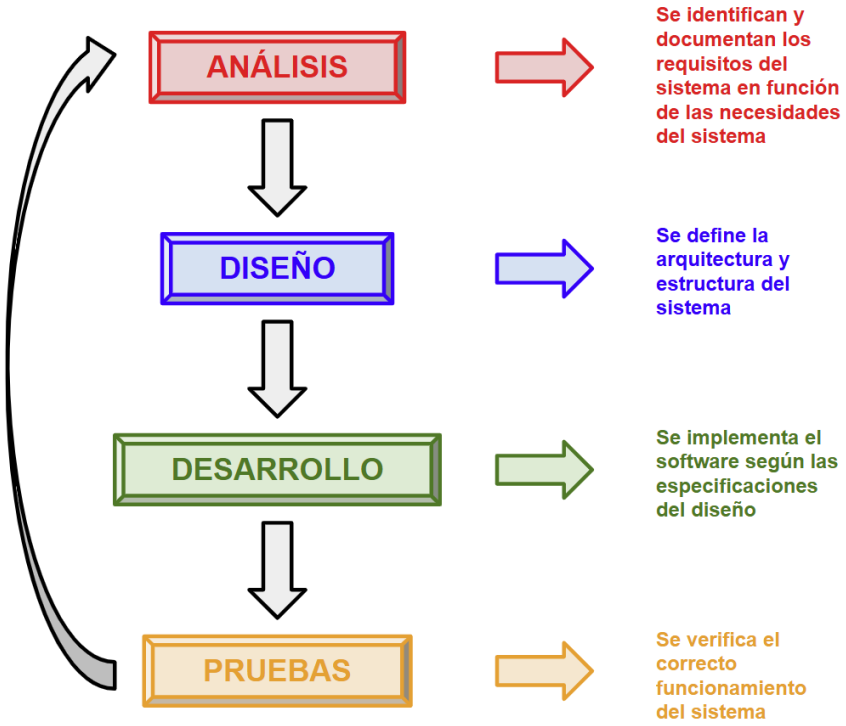

Figura 3.4: Ciclo de vida del sistema

En cuanto a las metodologías adoptadas, se ha optado por enfoques ágiles, que favorecen iteraciones cortas y una adaptación continua ante los cambios que los requisitos puedan sufrir durante el desarrollo del proyecto, garantizando así una evolución controlada y progresiva del mismo. A lo largo de todo el proceso, se han aplicado principios de diseño destinados a la modularidad, la separación de responsabilidades y la reutilización de código.

3.2. Implementación

En este apartado se describe el proceso de implementación del sistema desarrollado. En primer lugar, se detalla la lógica de extracción y procesamiento de datos mediante técnicas de scraping. A continuación, se expone la construcción de la aplicación web junto con las tecnologías y herramientas empleadas, diferenciando entre su backend y su frontend. Finalmente, se analizan en profundidad las dos funcionalidades principales del proyecto: el scouting y la predicción.

3.2.1. Scraping

Como se menciona en el capítulo anterior, la recopilación de datos se hace sobre la página de la categoría semiprofesional Tercera FEB [1] de la Federación Española de Baloncesto. Se recogen 13 jornadas de partidos, es decir, toda la fase de ida de la competición. Esta cantidad se considera adecuada para conformar una muestra representativa que permita identificar de forma realista los comportamientos y patrones de cada equipo con una consistencia estadística suficiente.

El script de Python “populate_database.py” se encarga de obtener los partidos indicados en el fichero de configuración del proyecto “configuration_file.txt” por medio de peticiones HTTP, con la librería requests [14], a la página web de la FEB [12]. A partir de las respuestas recibidas se recolectan una serie de elementos clave para cada partido de la competición. En base a estos, se organiza y construye el resto de la información relevante necesaria para el funcionamiento del sistema:

- A través de su API se obtiene:
  - Box Score de cada equipo: Es una tabla que contiene las estadísticas completas de cada jugador y del equipo tras finalizar el partido. Un partido tiene dos Box Scores; uno local y otro visitante.

									REBOTES						TAPONES			FALTAS			
I	D	JUGADOR	MIN	PT	T2	T3	TC	TL	RO	RD	RT	AS	BR	BP	TF	TC	MT	FC	FR	VA	+/-
	3	J. ATIENZA PEREA	06:50	0	0/0	0/2	0/2	0/0	1	1	2	0	0	1	0	0	0	1	0	-2	-3
	5	D. FERNANDEZ CAÑAS	11:17	4	2/3	0/0	2/3	0/0	0	1	1	0	0	1	0	0	0	2	1	2	-15
*	6	J. DOMINGUEZ LARRE	29:24	11	4/5	1/10	5/15	0/0	0	4	4	1	1	1	0	0	0	3	1	4	12
*	9	F. GOMEZ DE ENTERRIA LOPEZ	22:17	6	0/2	2/5	2/7	0/0	1	3	4	0	1	2	0	0	0	0	1	5	16
*	10	J. DE DOMINGO GARAY	12:46	5	1/2	1/3	2/5	0/0	0	1	1	1	0	0	0	0	0	1	3	6	2
	14	G. DIAZ MONTERO	25:57	16	2/2	3/9	5/11	3/3	2	2	4	6	2	1	0	0	0	3	2	20	-1
	15	A. SANCHO PEREZ	19:40	5	1/4	0/1	1/5	3/6	0	2	2	0	0	0	0	0	0	2	7	5	-18
*	25	G. LARDIES GUILLEN	22:01	11	4/9	0/0	4/9	3/3	5	4	9	2	1	1	0	1	0	4	2	15	27
*	27	A. SIMON DEL CORRAL	32:11	8	4/6	0/0	4/6	0/0	1	5	6	4	4	1	0	0	0	0	0	19	8
	29	J. MOLINA MELENDEZ	08:38	8	0/0	2/3	2/3	2/2	0	0	0	0	0	0	0	0	0	1	1	7	-5
	34	N. MAIGA	08:59	0	0/1	0/0	0/1	0/0	0	1	1	0	0	0	0	0	0	1	1	0	-13
	99	C. BARADJI	00:00	0	0/0	0/0	0/0	0/0	0	0	0	0	0	0	0	0	0	0	0	0	0
			200:00	74	18/34	9/33	27/67	11/14	10	24	34	14	9	8	0	1	0	18	19	81	

Figura 3.5: Ejemplo de un Box Score extraído de [1]

- Play by play: Es un registro detallado de las acciones que transcurren a lo largo de un partido. Entre las posibles acciones de un partido se encuentran: sustitución, periodo, rebote, tiro libre, tiro, falta, asistencia, pérdida, recuperación, tiempo muerto y tapón.

Figura 3.6: Ejemplo de un fragmento de un play by play extraído de [1]

- Shot Chart: Es una representación visual de todos los tiros de un partido y sus respectivas ubicaciones. En la Figura 3.7, los puntos de color azul indican los tiros fallados y los puntos verdes los acertados por ambos equipos a lo largo del encuentro.

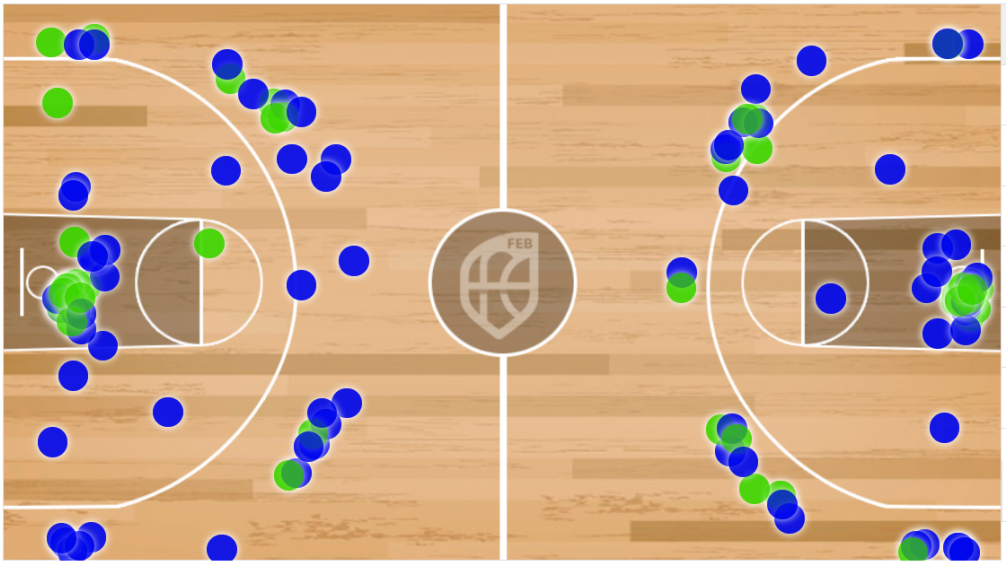

Figura 3.7: Ejemplo de un Shot Chart extraído de [1]

- Desde su web se descargan los archivos PDF que contienen las estadísticas completas de cada cuarto. Su contenido se estructura con la librería pandas [15] y se procesa con pdfplumber [16] y pdfminer [19].

El contenido que se extrae de estas operaciones se acaba registrando en una serie de objetos definidos que, utilizando la librería pickle [20], se almacenan de forma persistente en un archivo serializado “database.pkl”. Una vez se han procesado correctamente todos los partidos solicitados, el script procede con la eliminación de todos los archivos PDF, CSV y PNG generados, ya sea en anteriores ejecuciones o en la actual, exceptuando algunos archivos estáticos de la aplicación web que persisten siempre en el sistema, como los logos de los equipos o las imágenes principales de la aplicación. Por último, se actualiza la API del backend incorporando los nuevos equipos, jugadores y partidos que se hayan detectado durante este proceso, asegurando que el sistema siempre refleje el conjunto de datos más reciente.

El script “populate_database.py” está diseñado de tal manera que, al ejecutarlo, primero lea los partidos existentes en “database.pkl” y después únicamente procese los partidos nuevos, permitiendo la reutilización de los datos previamente guardados y evitando dedicar tiempo y recursos innecesarios en tareas de recolección ya realizadas con anterioridad. Además, desde este script se validan todos los datos procesados para prevenir entradas incompletas o inconsistencias, se manejan errores y excepciones de forma robusta en todo momento, y se registran logs para el seguimiento detallado del proceso de scraping. En caso de que se quiera restablecer la base de datos de Pickle, basta con eliminar el archivo “database.pkl” y ejecutar nuevamente “populate_database.py” para regenerar su contenido desde cero.

La principal limitación de este proceso de scraping radica en que la API requiere de un bearer token que únicamente puede adquirirse de forma manual al inspeccionar la página web y que se actualiza diariamente. Por consiguiente, cada vez que se desee efectuar una nueva extracción de datos en fechas diferentes es necesario actualizar previamente dicho token en el fichero de configuración del sistema. Este fichero “configuration_file.txt” está compuesto únicamente por dos pares clave-valor: “id_games”, que contiene todos los identificadores de los partidos requeridos separados por comas, y “bearer_token”.

Tras obtener toda la información necesaria para el sistema y almacenarla en “database.pkl”, los siguientes pasos consisten en crear las representaciones de dicho contenido mediante dos scripts: “scouting.py” y “prediction.py”. Ambos, explicados en sus respectivos subapartados de este capítulo, son responsables de desarrollar e incluir estas imágenes, tablas, gráficas, mapas de calor y comparaciones en el directorio public del frontend para su posterior acceso desde la aplicación web. Cada script, como su propio nombre indica, está enfocado en automatizar la generación de recursos visuales clave asociados a una funcionalidad específica. Esta separación funcional contribuye a la escalabilidad, modularidad y mantenibilidad del sistema, permitiendo la incorporación ágil de nuevas funcionalidades, así como futuras mejoras y correcciones de posibles errores existentes.

3.2.2. Aplicación web

La aplicación web es el componente software que posibilita la interacción del usuario con el sistema de scouting y predicción de resultados en ligas semiprofesionales de baloncesto. Su interfaz centrada en la eficiencia, la usabilidad y la experiencia fluida del usuario ofrece el acceso a múltiples representaciones visuales derivadas del previo procesamiento de los datos.

Tanto los datos relevantes del sistema como la propia aplicación en su conjunto son objeto de copias de seguridad periódicas, con el objetivo de preservar su integridad ante posibles fallos. Asimismo, el desarrollo del proyecto se lleva a cabo en un entorno virtual para garantizar la correcta gestión de dependencias y una mayor portabilidad y consistencia del mismo.

Desde una perspectiva arquitectónica, la aplicación se organiza en dos módulos claramente desacoplados: el backend y el frontend. El frontend se comunica con el backend a través de peticiones HTTP a una API REST implementada en el propio backend. Esta composición permite manejar grandes volúmenes de datos sin comprometer el rendimiento del sistema.

Backend

Para desarrollar el backend del sistema en Django [22] y construir la API REST clave para la comunicación con el frontend, es necesario seguir los siguientes pasos:

1.- Configuración del entorno virtual y creación del proyecto y la aplicación Django, junto con la modificación de sus ajustes generales para adaptarlos a las necesidades específicas de dicho proyecto.

2.- Integración de la herramienta Django REST Framework [21] dentro de la configuración general para la creación de la API REST con la que se va a comunicar el frontend.

3.- Definición de los modelos de datos que representan las entidades principales del sistema: Team, Player, Game y Prediction.

4.- Creación y migración de la base de datos. Entre las opciones de bases de datos a escoger se opta por SQLite [25], una solución ligera que requiere de una mínima configuración y que es ideal especialmente para fases de desarrollo y pruebas.

5.- Creación de los serializadores para convertir las instancias de los modelos a formato JSON.

6.- Implementación de las vistas mediante clases y funciones para gestionar las solicitudes del usuario y devolver las respuestas correspondientes.

7.- Configuración de las rutas URL de la API.

8.- Verificación del correcto funcionamiento del backend, probando la integración de la API, validando las respuestas y comprobando que los endpoints devuelvan los datos esperados.

El proyecto Django cuenta además con una página de administración integrada que permite gestionar de forma sencilla y centralizada las entidades principales del sistema, tales como equipos, jugadores, partidos y predicciones. El acceso a esta interfaz se encuentra restringido únicamente a un superusuario que se ha creado previamente con estos fines.

Como se puede apreciar en la Figura 3.8, la API REST dispone de una página principal, generada a partir de un template, que actúa como interfaz inicial, en la que se listan y describen brevemente todos los endpoints y recursos accesibles. Esta página actúa como una ayuda para la navegación y comprensión de las distintas funcionalidades ofrecidas por la API.

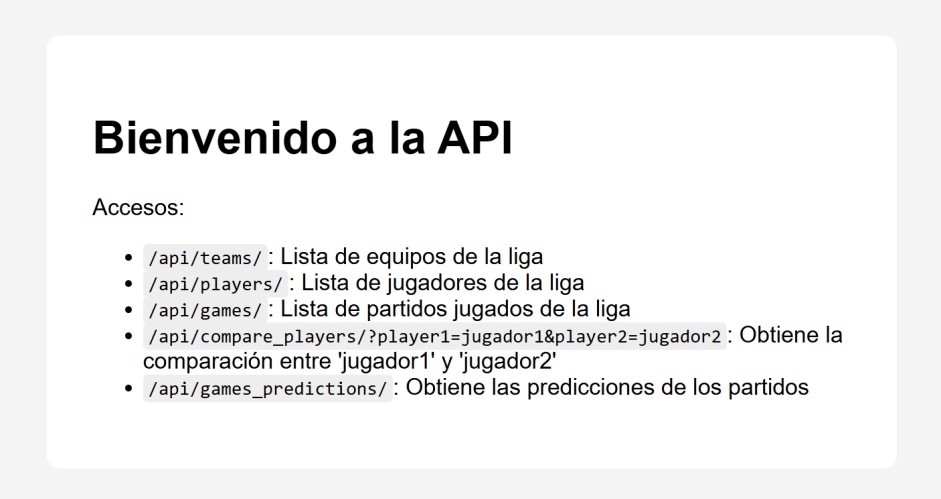

Figura 3.8: Página principal de la API

Originalmente, el objetivo de la API era que, al acceder a las distintas páginas de la aplicación que incluían selectores, estos se cargaran automáticamente con la información correspondiente, la cual ya había sido creada previamente y se encontraba disponible en la carpeta public del proyecto Angular. Los nombres de los equipos y los jugadores constituyen el contenido de dichos selectores. En cambio, los partidos disputados de la competición solo se emplean para comprobar si los dos equipos seleccionados, uno como local y otro como visitante, ya se han enfrentado durante la competición.

No obstante, durante el proceso de generación de representaciones visuales de los datos procesados con el script “scouting.py”, surgió un problema al intentar elaborar las imágenes de las comparaciones entre jugadores. Esta funcionalidad implicaba la creación de más de 34.000 imágenes, lo que exigía más de 10 horas de ejecución continua en el equipo empleado para el desarrollo del sistema. Para solventar este inconveniente se decidió aprovechar las capacidades que ofrece la API creando un endpoint específico que, al recibir los nombres de dos jugadores, crea y devuelve su respectiva comparativa. Estas imágenes se muestran en la aplicación de forma dinámica. La principal desventaja que presenta esta solución es que, pese al considerable ahorro de almacenamiento, recursos y tiempo de procesamiento, se pierde resolución en las imágenes generadas, incluso aumentando el valor del dpi.

Posteriormente, durante la implementación del apartado de la predicción en la aplicación, surgió una nueva adversidad. El modelo inicial de generación y lectura de las representaciones no permitía, dado un equipo local y otro visitante, obtener de forma directa una cadena de caracteres con el equipo ganador de la predicción y otra con la probabilidad de acierto de la misma. Como la visualización de toda la información en una única tabla no proporcionaba resultados suficientemente satisfactorios, se recurrió nuevamente a la API y se incorporó un nuevo endpoint que devuelve en formato JSON una lista con todas las predicciones de los partidos junto con sus respectivos resultados. El resto del proceso para mostrar esta información al usuario se gestiona directamente desde el frontend de manera eficiente y dinámica.

Frontend

El frontend de la aplicación se ha desarrollado utilizando Angular [26], un framework robusto de desarrollo web, basado en TypeScript [39], que permite crear interfaces dinámicas, modulares y altamente reactivas. Este módulo representa la parte visual e interactiva del sistema para el scouting y la predicción de resultados, siendo la única parte a la que el usuario tiene acceso de forma directa. A continuación, se detallan los pasos llevados a cabo para construir el frontend:

1.- Se crea la estructura base del proyecto y se configuran las opciones y los ajustes iniciales. Además, se opta por el uso de componentes standalone [40], que permiten definir cada componente de forma independiente sin necesidad de declararlos en un módulo. Esta situación facilita la reutilización de código, proporciona una mayor independencia y simplifica la gestión de las dependencias.

2.- Se generan los componentes que representan las vistas del sistema: clasificacion, equipos, equipos-compare, equipos-stats, home, home-button, jugadores, jugadores-compare, jugadores-search, jugadores-stats, partidos, partidos-boxscore, partidos-compare, partidos-plays, partidos-rankings, partidos-scorechart, partidos-shotchart, prediccion, ranking-equipos, ranking-jugadores, rankings y scouting.

3.- Se definen las rutas en el archivo “app-routing.module.ts” para enlazar cada componente con su URL.

4.- Se diseñan las vistas con el HTML y el CSS de cada componente.

5.- Se implementa la funcionalidad de los componentes a través de sus archivos TS.

El componente home se utiliza para simular la pantalla principal del sistema, visible en la Figura 3.9. Desde esta página se puede acceder a todas las funcionalidades disponibles mediante dos botones: uno que dirige al submódulo de scouting y otro al de predicción. Esta estructura simplifica significativamente la navegación al centralizar el acceso a las principales áreas del sistema desde un único punto de entrada.

Figura 3.9: Página principal de la aplicación

Como se menciona previamente, las representaciones visuales generadas se almacenan en la carpeta public del proyecto Angular, desde donde el frontend las carga directamente. Asimismo, es importante destacar que el diseño del frontend es responsive, lo que permite una correcta visualización y funcionamiento en pantallas con distintos tamaños.

3.2.3. Scouting

El scouting constituye el primer submódulo del sistema y sus representaciones visuales se generan a través del script “scouting.py”. Este tipo de recursos visuales se desarrollan con las librerías pandas [15] y matplotlib [41], que permiten procesar, estructurar y mostrar los datos de forma clara y eficaz. Seguidamente, se describen los diferentes tipos de representaciones existentes y detalles relevantes sobre sus respectivas implementaciones:

- **Tablas convertidas a imágenes:** Entre estas se encuentran la clasificación de la liga, los rankings y los cuadros de puntuación de partidos, y las estadísticas generales y los rankings por estadística de equipos y jugadores. Algunas de estas estadísticas difieren entre equipos y jugadores; por ejemplo, los equipos incluyen métricas como la racha de partidos, mientras que los jugadores cuentan con otras como las titularidades o el balance individual.

PO	Equipo	PJ	PG	PP	PF	PC	PT	R
1	MOVISTAR ESTUDIANTES	13	10	3	1075	953	23	+6
2	CB ARIDANE	13	9	4	1017	884	22	+4
3	REAL CANOE N.C.	13	9	4	1053	982	22	+1
4	C.B. TRES CANTOS	13	9	4	915	852	22	-2
5	LUJISA GUADALAJARA BASKET	13	9	4	1035	973	22	-3
6	SUN CHLORELLA DRAGONS	13	8	5	1014	935	21	+3
7	CB LA MATANZA	13	8	5	1011	973	21	-1
8	BALONCESTO TALAVERA	13	6	7	968	938	19	-1
9	BALONCESTO ALCALA	13	5	8	924	939	18	-1
10	TOSCONES CORRALEJO	13	5	8	826	937	18	-5
11	CABEZUELO CB SOCUÉLLAMOS	13	4	9	891	965	17	-2
12	CESUR DISTRITO OLIMPICO	13	4	9	883	963	17	+2
13	C.B. VALSEQUILLO	13	3	10	834	1006	16	+1
14	PABLO LASO ACADEMY	13	2	11	898	1044	15	+1

Figura 3.10: Ejemplo de clasificación tras 13 jornadas

- **Tablas filtrables:** En el sistema solo hay dos tablas de este tipo: una para la búsqueda por perfil de jugadores y otra para el jugada a jugada de los partidos. Estas tablas ofrecen la posibilidad de filtrar dinámicamente sus columnas. Por ello, se incorpora un botón de ayuda que explica los distintos tipos de filtros disponibles e indica varios ejemplos de uso.

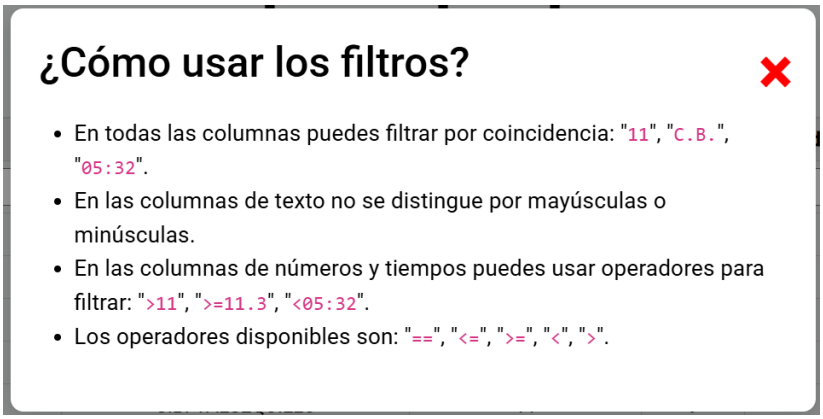

Figura 3.11: Ejemplo de ayuda de los filtros

- **Comparaciones:** Se incluyen en los apartados de comparativas de equipos de partidos y de valores medios de equipos. Se utilizan gráficos de barras enfrentadas para mostrar estadísticas comparadas. El vencedor de cada estadística, si no se produce un empate, se marca con una estrella. En el caso de estadísticas con valores negativos, como puede suceder con la valoración, se evita que las barras se superpongan representando ese valor como si dicha barra fuera un cero. Cabe resaltar que, aunque las comparativas entre jugadores se obtienen de forma dinámica a través de la API y no mediante el script “scouting.py”, se integra conceptualmente en este grupo.

Figura 3.12: Ejemplo de comparaciones de estadísticas

- **Gráficos de puntuación de los partidos:** Representan la evolución del marcador en un partido mediante dos líneas, una por cada equipo, que reflejan la puntuación acumulada por minuto a lo largo del encuentro. Además, se añade para cada equipo una etiqueta sobre los puntos en los que la diferencia de puntuación alcanza su valor máximo durante el partido, si esta existe, para destacarlos visualmente. Con este gráfico se puede identificar de manera rápida y clara los momentos de mayor dominio o ventaja por parte de un equipo dentro del desarrollo del encuentro.

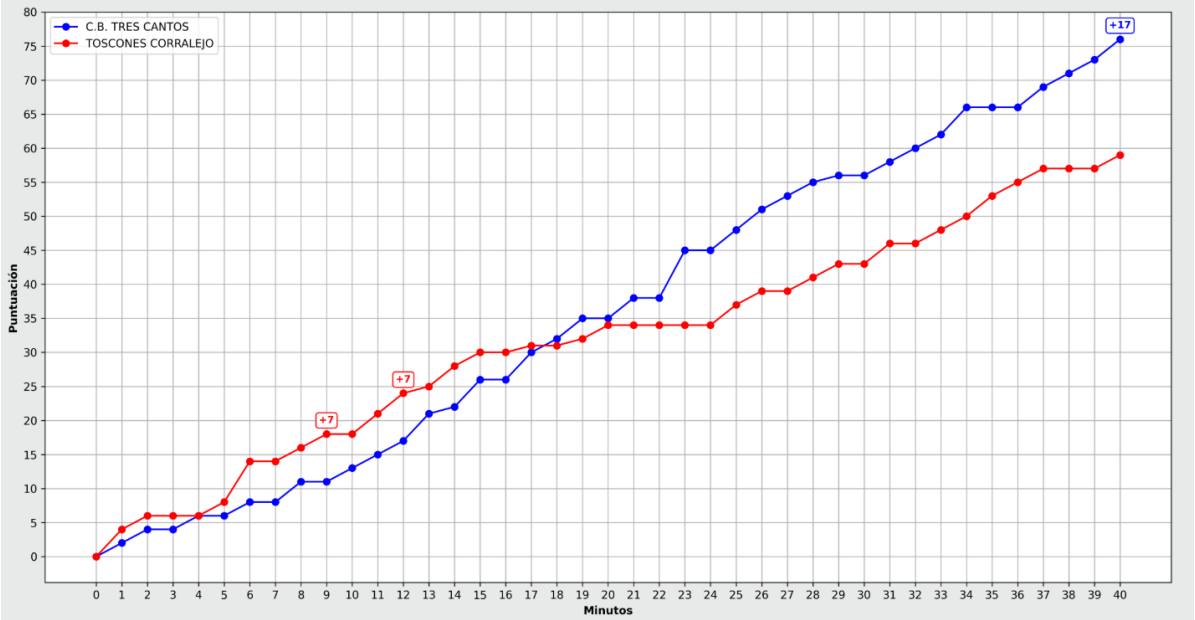

Figura 3.13: Ejemplo de gráfico de puntuación de un partido

- **Gráficos de tiro de los partidos:** Se construyen sobre una imagen de una pista de baloncesto obtenida de la FEB. En ella, se colocan puntos en función de las coordenadas de los tiros realizados durante el partido. Los tiros acertados se muestran en color verde y los fallados en rojo. En los partidos se ofrece un gráfico de tiro independiente para cada equipo.

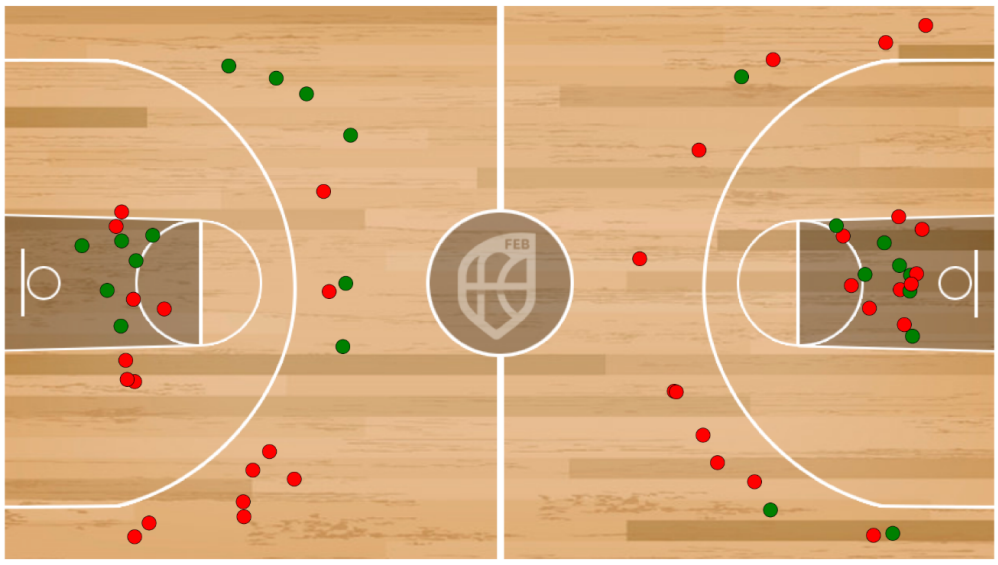

Figura 3.14: Ejemplo de gráfico de tiro de un equipo en un partido

- **Mapas de calor:** Se sitúan en los apartados de las estadísticas generales de equipos y jugadores. Para su creación se utiliza la mitad del campo de los gráficos de tiro y se divide en regiones. Las coordenadas de los tiros se ajustan invirtiendo sus ejes, horizontal y vertical,

y se colocan sobre la pista. Acto seguido, se agrupan los tiros por región y, en cada una de ellas, se indica el número de tiros metidos e intentados junto a su porcentaje de acierto.

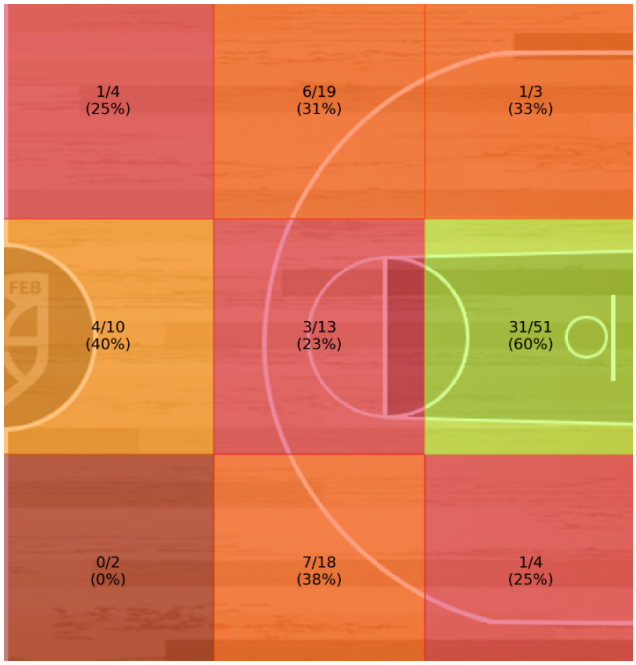

Figura 3.15: Ejemplo de mapa de calor de un partido

En cada región se asigna un color en función de su porcentaje de acierto. El color blanco indica ausencia de tiros en esa región. La escala de los colores de las regiones se puede observar en la Figura 3.16.

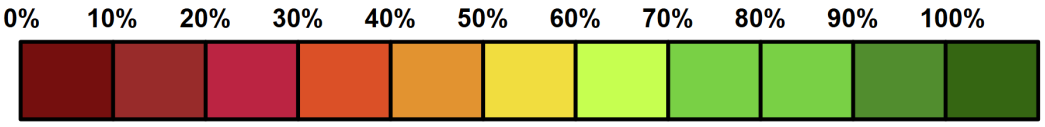

Figura 3.16: Porcentaje de acierto en tiros

Desde el apartado de partidos del sistema, como se ha comentado previamente, se puede acceder a toda la información relativa a un encuentro ya disputado entre dos equipos mediante seis botones: cuadros de puntuación, comparación entre equipos, gráfico de puntuación, jugada a jugada, rankings y gráfico de tiro. Durante su implementación, surgió un problema: al navegar a cualquiera de las páginas vinculadas a estos botones de un partido y regresar a la página de los partidos, los selectores no conservaban las opciones que el usuario había seleccionado con anterioridad. Esto implicaba que, al volver, los valores de los selectores se reiniciaban, causando pérdida de contexto y una experiencia poco fluida. Para resolver esta situación, se implementó un service en Angular. Este service mantiene el estado de estos selectores mientras el usuario navega entre las distintas secciones de un partido; así, al regresar a la página de los partidos, los selectores se inicializan con sus respectivos valores almacenados en dicho service.

3.2.4. Predicción

La predicción de resultados conforma el segundo submódulo del sistema. El contenido de los resultados de las predicciones mostrado en la pantalla de este submódulo proviene de dos fuentes distintas. Por un lado, la tabla que presenta los valores de predicción para cada equipo se genera previamente, utilizando las librerías pandas [15] y matplotlib [41], mediante el script "prediction.py". Por otro lado, tanto el equipo ganador de la predicción como su correspondiente nivel de probabilidad de acierto, ambos extraídos a partir de los valores de predicción, se obtienen a través de peticiones a la API REST. Esta división funcional responde a las limitaciones expuestas previamente en la sección 3.2.2, dedicada a la implementación de la aplicación web, concretamente en el último párrafo del apartado del backend.

El objetivo de estas predicciones es estimar el ganador de un partido de baloncesto dados un equipo local y otro visitante. Para ello, el sistema calcula el valor predictivo para cada equipo mediante una fórmula fija que se ha diseñado en base a una selección de estadísticas consideradas claves y diferenciales. Este valor varía en función del rival y del factor cancha, es decir, de si el equipo juega como local o visitante. El equipo que obtiene un mayor valor predictivo es designado como el ganador de la predicción, y el nivel de probabilidad de acierto asociado a dicha predicción puede calificarse como baja, media o alta dependiendo de la diferencia entre los valores predictivos de ambos equipos.

La fórmula que calcula el valor predictivo P de un equipo se compone de dos partes principales:

1.- Baremos principales (B): Corresponden a una suma ponderada de indicadores estadísticos clave. Cada baremo se multiplica por su peso específico asociado, el cual refleja su relevancia dentro del modelo predictivo. Los valores más frecuentes de B oscilan entre 0 y 11; sin embargo, en circunstancias extremadamente excepcionales pueden registrarse valores negativos de hasta -5.5. Los baremos que se han considerado esenciales para el cálculo de las predicciones, así como sus posibles valores, son los siguientes:

- **win_probability:** La probabilidad de victoria del equipo, es decir, el número total de victorias entre los partidos jugados.
- **remainder:** La racha de victorias o derrotas consecutivas del equipo. Su valor solo puede ser cero si el equipo no ha jugado ningún partido en la competición. En la Tabla 3.1 se presentan los distintos valores que puede adoptar este baremo.

Remainder	Valor
$\geq$10	1
[5, 9]	0.5
[1, 4]	0.25
[-4, -1]	-0.25
[-9, -5]	-0.5
$\leq$-10	-1

Tabla 3.1: Tabla de posibles valores del baremo remainder

- **site_probability:** La probabilidad de victoria del equipo cuando juega como local o como visitante, según corresponda.
- **site:** Su valor es 0.5 si el equipo juega como local, -0.5 si lo hace como visitante o 0 si el partido se disputa en un campo neutral, ya que jugar en una pista que no es habitual para ninguno de los dos equipos no supone realmente una ventaja o una desventaja.
- **points_differential:** La diferencia entre los puntos totales anotados y los puntos totales recibidos del equipo. En la Tabla 3.2 se indican cuáles son sus posibles valores.
- **average_pir:** La media de valoración del equipo. En la Tabla 3.2 también se muestran los posibles valores de este baremo.

Points differential	Average PIR	Valor
$\geq$500	$\geq$100	2
[400, 499]	[90, 99]	1.8
[300, 399]	[80, 89]	1.6
[200, 299]	[70, 79]	1.4
[100, 199]	[60, 69]	1.2
[75, 99]	[50, 59]	1
[50, 74]	[40, 49]	0.8
[25, 49]	[30, 39]	0.5
[1, 24]	[1, 29]	0.25
0	0	0
[-24, -1]	[-29, -1]	-0.25
[-49, -25]	[-39, -30]	-0.5
[-74, -50]	[-49, -40]	-0.8
[-99, -75]	[-59, -50]	-1
[-199, -100]	[-69, -60]	-1.2
[-299, -200]	[-79, -70]	-1.4
[-399, -300]	[-89, -80]	-1.6
[-499, -400]	[-99, -90]	-1.8
$\leq$-500	$\leq$-100	-2

Tabla 3.2: Tabla de posibles valores de los baremos points_differential y average_pir

Los pesos asignados a cada baremo han sido determinados, según la importancia relativa estimada para cada uno, tras investigaciones detalladas y rigurosas sobre datos históricos de la competición. Estos valores reflejan la influencia de cada factor en la fórmula de predicción general:

Baremo	Peso
win_probability	27.27 %
remainder	9.09 %
site_probability	22.73 %
site	4.55 %
points_differential	18.18 %
average_pir	18.18 %

Tabla 3.3: Tabla de los baremos y sus pesos

Después de describir los baremos involucrados, se presenta la fórmula para calcular B:

$$\begin{aligned} B = &(win\_probability \times 3 + remainder) \\ &+ (site\_probability \times 2{,}5 + site) \\ &+ (points\_differential + average\_pir) \end{aligned} \tag{3.1}$$

2.- Coeficiente de ajuste (C): Permite ajustar o diferenciar las predicciones de los equipos que se enfrentan, teniendo en cuenta factores contextuales no capturados directamente por los baremos estadísticos. En este caso, se basa en el último enfrentamiento previo entre los dos equipos implicados, teniendo en cuenta que el sistema de competición sigue un formato que enfrenta a los equipos dos veces. Además, con el fin de mejorar la precisión global de la predicción, por cada prórroga jugada en el último partido se suma 0.15 al coeficiente de ajuste del equipo con menor valor predictivo y se resta 0.15 al del equipo con mayor valor.

Situación	Valor
Si gana por más de 19 como local	1.5
Si gana por más de 9 como visitante	1.5
Si gana por [10, 19] como local	1
Si gana por [1, 9] como visitante	1
Si gana por [6, 9] como local	0.5
Si gana por [1, 5] como local	0.25
Si no han disputado ningún partido previo	0
Si pierde por [1, 5] como visitante	-0.25
Si pierde por [6, 9] como visitante	-0.5
Si pierde por [1, 9] como local	-1
Si pierde por [10, 19] como visitante	-1
Si pierde por más de 9 como local	-1.5
Si pierde por más de 19 como visitante	-1.5

Tabla 3.4: Tabla de posibles valores del coeficiente de ajuste

En definitiva, la fórmula general de la predicción de resultados se expresa como:

$$P = B + C = \sum_{i=1}^{n} w_i \cdot b_i + C \tag{3.2}$$

Donde:

- wi y bi son el peso y el valor del i-ésimo baremo, respectivamente.
- n es el número total de baremos considerados.

En conclusión, las predicciones del sistema incluyen:

- Una tabla con los valores predictivos P de cada equipo calculados con la fórmula 3.2.
- El ganador de dicha predicción o el equipo que cuenta con un valor predictivo mayor.
- La probabilidad de acierto, que se establece en función de la magnitud de la diferencia entre los valores predictivos:

Diferencia entre los valores predictivos	Probabilidad de acierto
< 2	BAJA
[2, 4]	MEDIA
> 4	ALTA

Tabla 3.5: Tabla de posibles valores para la probabilidad de acierto de una predicción

En la Figura 3.17 se muestra un ejemplo de una predicción entre dos equipos:

Figura 3.17: Ejemplo de una predicción desde la aplicación web

4

Pruebas y Resultados

En este capítulo se detallan las pruebas realizadas sobre la funcionalidad de predicción del sistema desarrollado. Para ello, se comienza por la descripción del entorno de pruebas utilizado y se continúa con la evaluación específica de predicción de resultados mediante la fórmula fija, con y sin coeficiente de ajuste, y los modelos predictivos. Posteriormente, se comparan los resultados obtenidos entre las pruebas realizadas para valorar su precisión y efectividad.

4.1. Entorno de pruebas

Para la realización de las pruebas del sistema se ha utilizado el mismo entorno que durante la implementación, con el objetivo de garantizar la consistencia entre las fases de desarrollo y de pruebas al minimizar posibles incompatibilidades, errores inesperados o desviaciones en los resultados obtenidos en dicho proceso de verificación. Esta decisión permite evaluar el rendimiento y el comportamiento del sistema en un entorno controlado, reproducible e idéntico al de implementación, lo que asegura que cualquier fallo detectado se pueda atribuir al propio sistema y no a diferencias en el entorno de ejecución.

En la Tabla 4.1 se recogen las especificaciones técnicas del equipo en el que se han ejecutado todas las pruebas:

Componente	Características
Sistema operativo	Microsoft Windows 11 Home
CPU	Intel Core i5-1035G1
GPU	Intel UHD Graphics
Memoria RAM	8 GB
Entorno virtual	venv (Python 3.13.3)
Editor / IDE	Visual Studio Code 1.101.0
Navegador	Google Chrome 137.0 (64 bits)

Tabla 4.1: Tabla de especificaciones técnicas del equipo

Por otra parte, se ha empleado un entorno virtual de Python para gestionar de forma adecuada, ordenada y eficiente las dependencias del proyecto. En la Tabla 4.2 se pueden consultar los principales paquetes utilizados a lo largo del ciclo de vida completo del sistema, junto con sus respectivas versiones.

Nombre del paquete	Versión
asgiref	3.8.1
Django	5.2.1
django-cors-headers	4.7.0
djangorestframework	3.16.0
joblib	1.5.1
matplotlib	3.10.3
numpy	2.2.5
pandas	2.2.3
pdfminer.six	20250327
pdfplumber	0.11.6
pickle	4.0
python	3.13.3
python-dateutil	2.9.0.post0
pytz	2025.2
requests	2.32.3
scikit-learn	1.7.0
six	1.17.0
sqlparse	0.5.3
tabulate	0.9.0
typing_extensions	4.13.2
tzdata	2025.2

Tabla 4.2: Tabla de paquetes y versiones del entorno de pruebas

4.2. Pruebas

Inicialmente, se dispone de una base de datos que contiene las estadísticas y los resultados de la primera vuelta de la competición, es decir, las 13 primeras jornadas completas. A partir de ella, se generan 7 nuevos conjuntos de datos, cada uno incorporando los 7 partidos de una jornada adicional, desde la 14 hasta la 20, sobre la base de datos de la jornada anterior. De este modo, se construyen un total de 8 bases de datos: una con las 13 primeras jornadas como referencia histórica y siete con actualizaciones progresivas para evaluar las predicciones jornada a jornada. En total, esto permite

evaluar predicciones de 49 partidos de liga, utilizando siempre la información disponible más reciente.

Para evaluar el rendimiento de la fórmula fija se genera una predicción por cada partido de cada jornada, utilizando la base de datos correspondiente a la jornada inmediatamente anterior. Posteriormente, se contrastan los resultados obtenidos con los reales, extraídos de la base de datos de la jornada que se intenta predecir. Con el fin de analizar el impacto del coeficiente de ajuste, se evalúa el comportamiento de la Ecuación 3.2 tanto con el coeficiente como sin él (factor C en dicha ecuación).

Además, se intenta pronosticar el resultado de los partidos mediante el uso de modelos predictivos de clasificación binaria: regresión logística, k-vecinos más cercanos o KNN, árbol de decisión, bosques aleatorios, refuerzo de gradiente y redes neuronales. Para estos modelos se utilizan las 13 primeras jornadas como conjunto de entrenamiento dentro de un enfoque de aprendizaje supervisado. Cada partido introduce como variables de entrada las mismas estadísticas empleadas como baremo en la fórmula fija, junto con su respectivo resultado: 0 si gana el equipo local y 1 si gana el visitante. El objetivo es predecir de forma consecutiva las 7 jornadas siguientes. Para ello, se extraen los datos de entrada de las bases de datos correspondientes a cada jornada y se comparan las predicciones de los modelos con los resultados reales. Dado que el conjunto de entrenamiento se limita exclusivamente a la primera vuelta del campeonato, no se dispone de información sobre enfrentamientos previos entre equipos. Por tanto, esta configuración es comparable a las predicciones de la fórmula fija sin coeficiente de ajuste.

A continuación, se presentan las distintas métricas utilizadas para evaluar las predicciones de la fórmula fija y de los modelos predictivos. En todos los casos se asume que la clase 0 corresponde a una victoria del equipo local, mientras que la clase 1 indica una victoria del equipo visitante:

- Número total de aciertos y media de aciertos por jornada.
- Número total de fallos y media de fallos por jornada.
- Accuracy.
- Para cada una de las dos clases, 0 y 1:
    - Precision.
    - Recall.
    - F1 score.

Adicionalmente, en las predicciones generadas mediante fórmula fija, se incluyen detalles de los errores cometidos en las predicciones:

- Debido a que la fórmula asigna un nivel de probabilidad de acierto a cada predicción, se recoge el porcentaje que representa cada uno de estos niveles dentro del conjunto de predicciones incorrectas. Esto permite visualizar si los errores tienden a concentrarse en un nivel u otro.

- Se elabora una tabla con los equipos que aparecen en al menos dos errores de predicción dentro de una misma situación, lo que puede indicar una mayor imprevisibilidad o irregularidad en su rendimiento. Si un equipo aparece solo una vez en una misma situación de error, se considera un caso aislado, sin valor analítico significativo. Las dos situaciones de error posibles son:
  - Equipos que ganan en la predicción pero pierden el partido.
  - Equipos que pierden en la predicción pero ganan el partido.

4.3. Resultados

En este apartado se muestran los resultados de las pruebas realizadas sobre las predicciones generadas mediante la fórmula fija, con y sin coeficiente de ajuste, y los modelos predictivos. Todas las métricas se obtienen a través de scripts automatizados; sin embargo, por motivos de legibilidad y presentación, se incluyen tablas comparativas, capturas de pantalla y representaciones gráficas derivadas de la ejecución de dichos scripts. Estas visualizaciones ofrecen una interpretación más clara y estructurada de los resultados, lo que facilita su análisis.

4.3.1. Fórmula fija de la predicción

En este apartado se presentan los resultados obtenidos mediante la fórmula fija, tanto con coeficiente de ajuste como sin él. La comparación entre ambas configuraciones permite identificar el impacto que tiene el coeficiente de ajuste en la precisión de las predicciones. La Tabla 4.3 muestra los valores correspondientes a cada una de las métricas evaluadas: aciertos y fallos sobre el total de partidos, accuracy, precision, recall y F1 score.

Métrica	Sin coeficiente de ajuste	Con coeficiente de ajuste
Aciertos	35/49	40/49
Fallos	14/49	9/49
Accuracy	71.43 %	81.63 %
Precision (0)	73.53 %	81.82 %
Recall (0)	83.33 %	90.00 %
F1 Score (0)	78.12 %	85.71 %
Precision (1)	66.67 %	81.25 %
Recall (1)	52.63 %	68.42 %
F1 Score (1)	58.82 %	74.29 %

Tabla 4.3: Tabla de resultados de la fórmula fija

De igual forma, se añade información acerca de los errores de predicción de ambos modelos a través de la Figura 4.1. En la columna de frecuencias, la letra B indica probabilidad de acierto baja, la M media y la A alta.

ERRORES EN LAS PREDICCIONES CON COEFICIENTE DE AJUSTE

Probabilidades de acierto: BAJA = 55.56%, MEDIA = 33.33%, ALTA = 11.11%

	Situación	Equipo	Frecuencia (> 2)
1	Ganador en predicción y perdedor en partido	LUJISA GUADALAJARA BASKET	3 (B=2, M=1, A=0)
2	Ganador en predicción y perdedor en partido	REAL CANOE N.C.	2 (B=1, M=0, A=1)

Tiempo de ejecución: 19.13 segundos

ERRORES EN LAS PREDICCIONES SIN COEFICIENTE DE AJUSTE

Probabilidades de acierto: BAJA = 50.0%, MEDIA = 35.71%, ALTA = 14.29%

	Situación	Equipo	Frecuencia (> 2)
1	Perdedor en predicción y ganador en partido	BALONCESTO ALCALA	3 (B=2, M=1, A=0)
2	Perdedor en predicción y ganador en partido	CB ARIDANE	2 (B=1, M=1, A=0)
3	Perdedor en predicción y ganador en partido	CESUR DISTRITO OLIMPICO	2 (B=1, M=0, A=1)
4	Ganador en predicción y perdedor en partido	LUJISA GUADALAJARA BASKET	5 (B=3, M=2, A=0)
5	Ganador en predicción y perdedor en partido	REAL CANOE N.C.	2 (B=1, M=0, A=1)
6	Ganador en predicción y perdedor en partido	MOVISTAR ESTUDIANTES	2 (B=1, M=1, A=0)

Tiempo de ejecución: 21.65 segundos

Figura 4.1: Errores en las predicciones con fórmula fija

Las probabilidades de acierto de ambos modelos se representan mediante gráficas circulares 3D, recogidas en la Figura 4.2, para poder apreciar la distribución relativa de cada nivel. Esta Figura junto a la 4.1 ofrecen una perspectiva complementaria y enriquecedora sobre el impacto del coeficiente de ajuste, revelando detalles que las métricas de la Tabla 4.3 no pueden captar por sí solas. El reconocimiento de fortalezas y debilidades de cada modelo en diferentes rangos de predicción es un paso clave para futuras adaptaciones y mejoras que permitan aproximar este tipo de predicciones a su máximo potencial.

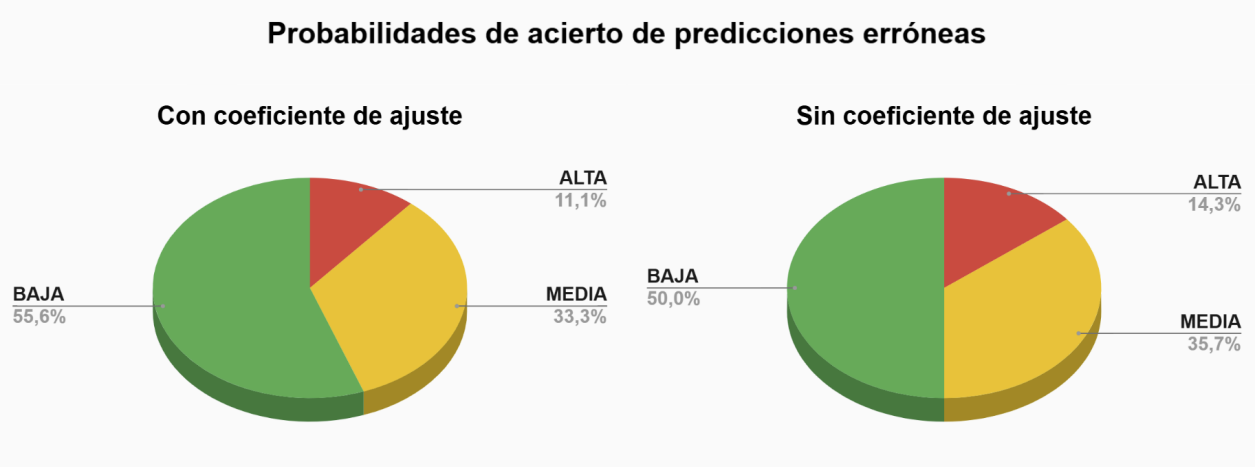

Figura 4.2: Probabilidades de acierto de predicciones erróneas con fórmula fija

4.3.2. Modelos predictivos

Para los modelos predictivos se han explorado distintos rangos de hiperparámetros hasta identificar las combinaciones que ofrecen un mejor rendimiento en este tipo de predicciones. Este proceso ha implicado una evaluación exhaustiva de múltiples configuraciones para maximizar las precisiones y las capacidades de generalización de cada modelo. En la Tabla 4.4 se recogen los resultados proporcionados por los seis modelos predictivos de clasificación binaria escogidos: regresión logística, KNN, árbol de decisión, bosques aleatorios, refuerzo de gradiente y redes neuronales.

Métrica	Regresión logística	KNN	Árbol de decisión	Bosques aleatorios	Refuerzo de gradiente	Redes neuronales
Aciertos	25/49	30/49	15/49	34/49	18/49	32/49
Fallos	24/49	19/49	34/49	15/49	31/49	17/49
Accuracy	51.02 %	61.22 %	30.61 %	69.39 %	36.73 %	65.31 %
Precision (0)	87.50 %	61.22 %	30.00 %	74.19 %	42.86 %	70.97 %
Recall (0)	23.33 %	100.00 %	10.00 %	76.67 %	10.00 %	73.33 %
F1 Score (0)	36.84 %	75.95 %	15.00 %	75.41 %	16.22 %	72.13 %
Precision (1)	43.90 %	0.00 %	30.77 %	61.11 %	35.71 %	55.56 %
Recall (1)	94.74 %	0.00 %	63.16 %	57.89 %	78.95 %	52.63 %
F1 Score (1)	60.00 %	0.00 %	41.38 %	59.46 %	49.18 %	54.05 %

Tabla 4.4: Tabla de resultados de los modelos predictivos

4.4. Comparativa de resultados

Como se puede apreciar en los resultados del apartado 4.3.1, la fórmula fija presenta un rendimiento superior cuando se le aplica el coeficiente de ajuste. En concreto, la diferencia total es de 5 aciertos, lo que supone una mejora de un 10 % en su accuracy. Además, el resto de métricas también muestran valores superiores al emplear el coeficiente. En la versión en la que no se aplica el coeficiente se observa un mayor número de equipos y mayores frecuencias en la tabla de errores de predicción, lo que sugiere más imprevisibilidades o irregularidades en equipos de la liga. Cabe destacar que, en ambos casos, los errores de predicción tienden a concentrarse en niveles de probabilidad de acierto más bajos, lo que indica que las predicciones con menor probabilidad de acierto son las más susceptibles de fallo. Todo esto evidencia que la función del coeficiente C, al ajustar o diferenciar las predicciones de los equipos que se enfrentan, contribuye de manera significativa a la mejora de la precisión global de las predicciones. En la Figura 4.3 se pueden ver las tendencias que sigue cada modelo a lo largo de las 7 jornadas (de la 14 a la 20, ambas incluidas).

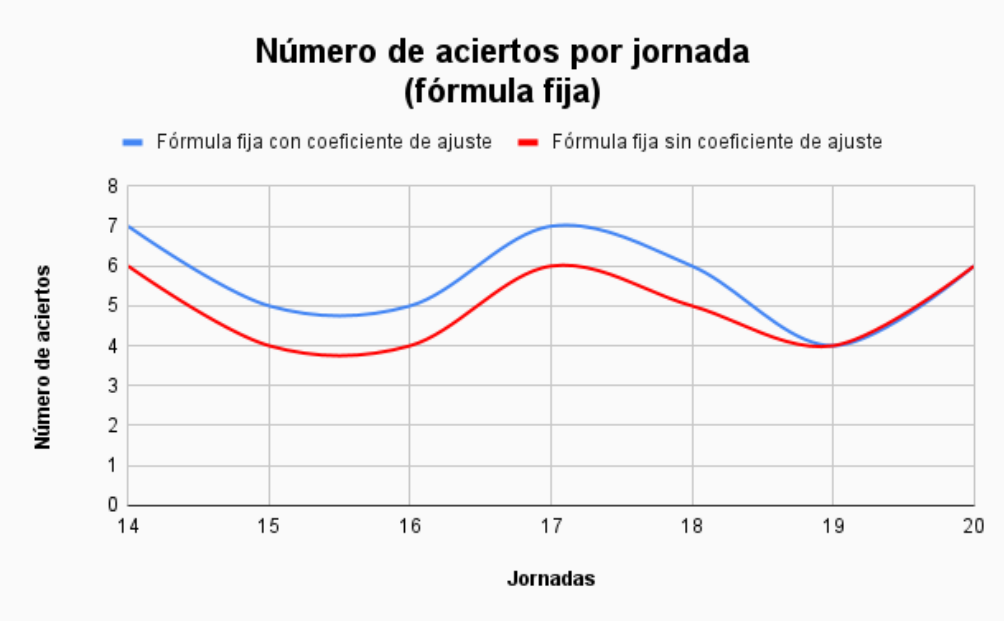

Figura 4.3: Número de aciertos por jornada para la fórmula fija

Por otro lado, al comparar los resultados de los apartados 4.3.1 y 4.3.2, se puede discernir que la fórmula fija, a pesar de no incorporar el coeficiente de ajuste, supera en rendimiento a cualquiera de los modelos predictivos. Esto puede atribuirse a que la fórmula fija ha sido diseñada con conocimiento específico del dominio para un propósito concreto, mientras que los modelos tienen un carácter más general. Además, los modelos suelen ofrecer mejores resultados cuando se dispone de mayores cantidades de datos, por lo que, en escenarios con un historial limitado, la fórmula fija resulta más efectiva. Dentro de los modelos evaluados, destacan frente al resto los bosques aleatorios, las redes neuronales y KNN, cuyos resultados se encuentran relativamente próximos a los de la fórmula fija sin coeficiente.

Al igual que en la comparación anterior, en la Figura 4.4 se pueden apreciar las tendencias que sigue cada modelo durante las 7 jornadas.

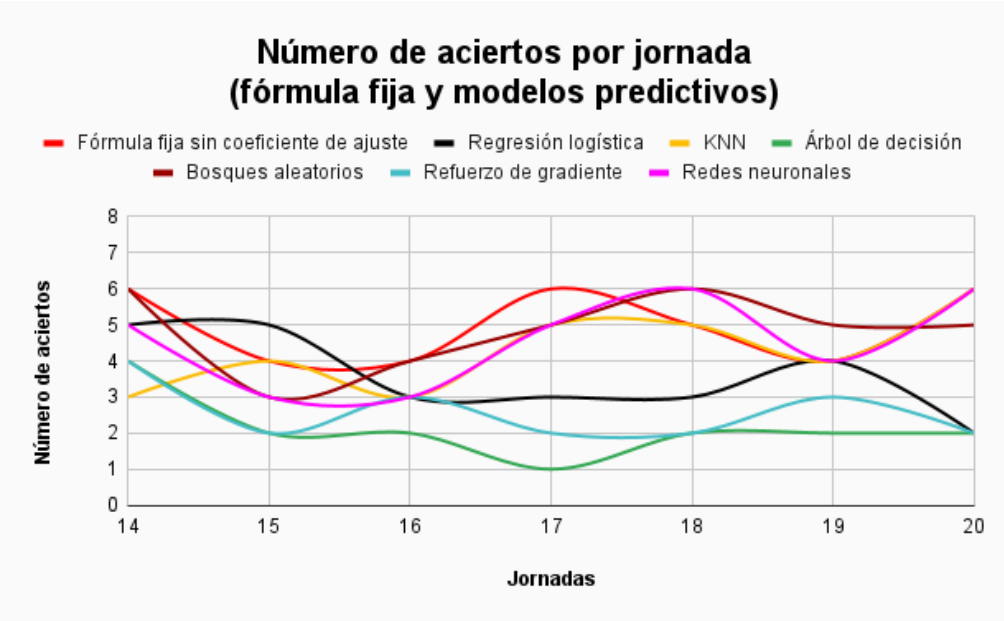

Figura 4.4: Número de aciertos por jornada para la fórmula fija y los modelos predictivos

Finalmente, la Figura 4.5 representa una comparación visual del accuracy obtenido por todos los modelos.

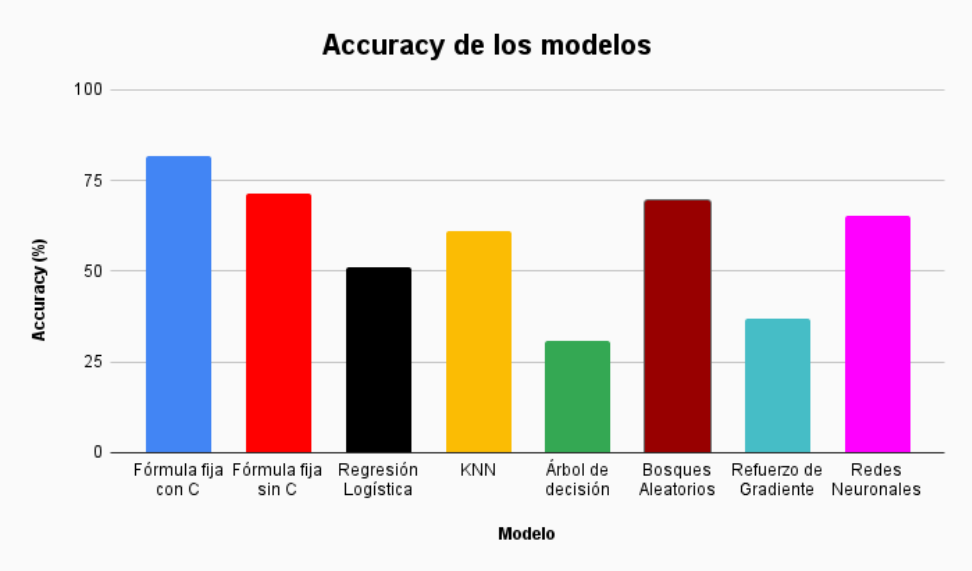

Figura 4.5: Gráfica del accuracy de los modelos

5

Conclusiones y Trabajo futuro

En este último capítulo se exponen las principales conclusiones derivadas del desarrollo del presente proyecto, así como las posibles líneas de trabajo futuro que podrían explorarse para la mejora y la ampliación del sistema propuesto.

5.1. Conclusiones

El objetivo primordial de este proyecto ha sido crear un sistema de scouting y predicción de resultados orientado a ligas semiprofesionales de baloncesto, un entorno donde el acceso a herramientas tecnológicas avanzadas y el análisis de datos sigue presentando amplios márgenes de mejora en comparación con las categorías de élite. A lo largo del desarrollo de este trabajo, se han automatizado procesos clave de recopilación, análisis y explotación de datos, permitiendo ofrecer una solución funcional tanto como herramienta de entretenimiento como de apoyo en la toma de decisiones técnicas y tácticas.

En primer lugar, se ha extraído información de la página web oficial de la Federación Española de Baloncesto mediante técnicas de scraping. A partir de este contenido se ha construido una base de datos estructurada y enfocada a dar soporte a las distintas funcionalidades del sistema.

A continuación, se ha diseñado y desplegado una aplicación web que integra los módulos de scouting, para facilitar el análisis detallado de equipos y jugadores, y de predicción de resultados, basado en una fórmula fija personalizada y elaborada específicamente para este propósito. Este sistema de predicción se ha evaluado y comparado posteriormente con las predicciones generadas por diversos modelos predictivos de clasificación binaria para observar sus rendimientos y comportamientos.

Los resultados obtenidos en la fase de pruebas muestran que la fórmula fija, especialmente cuando incorpora un coeficiente de ajuste, proporciona métricas de precisión superiores a las de los modelos de aprendizaje automático evaluados. Esto refuerza la idea de que, en situaciones donde la cantidad de datos es más limitada, es preferible emplear enfoques específicos diseñados para el problema concreto, en lugar de modelos genéricos que requieren grandes volúmenes de información para alcanzar un rendimiento comparable.

En definitiva, este proyecto ha supuesto una excelente demostración de los conocimientos y habilidades adquiridos a lo largo del grado de ingeniería informática, además de una oportunidad para explorar nuevas herramientas tecnológicas y ámbitos menos conocidos, como el scraping de datos o el diseño de fórmulas destinadas a la predicción de resultados. Esta experiencia ha permitido vislumbrar una ínfima parte del enorme potencial y las posibles aplicaciones que ofrece un campo en crecimiento y actualmente en auge, la ciencia de datos.

5.2. Trabajo futuro

El sistema desarrollado a lo largo de este proyecto conforma una base sólida, pero existen múltiples vías de mejora y ampliación que permitirían evolucionar la herramienta hacia un producto más completo, robusto y automatizado.

Una de las principales líneas de mejora sería la automatización completa del proceso de representación de datos, de forma que se elimine la necesidad de generar previamente los recursos y se extraigan directamente desde la API, como sucede con las comparaciones de los jugadores. Esto reduciría los tiempos de mantenimiento y minimizaría los posibles riesgos que supone la introducción de datos actual.

Durante el desarrollo del sistema, se mostró la aplicación web a un total de cinco entrenadores pertenecientes a esta categoría de competición. Todos coincidieron en señalar las ventajas prácticas que supondría contar con una herramienta como esta y manifestaron su interés en utilizarla de forma activa. Además, realizaron una serie de propuestas de mejora en las representaciones del sistema. Algunas de estas fueron incorporadas exitosamente, como la inclusión de las máximas diferencias de cada equipo en las gráficas de puntuación de los partidos. Otras, sin embargo, no pudieron implementarse debido a la falta de datos disponibles o a la complejidad de su obtención, como el cálculo preciso del número de posesiones por equipo en cada encuentro.

En relación con el módulo de predicción de resultados, a pesar de que la fórmula desarrollada ha ofrecido buenos resultados, sigue existiendo margen de mejora en el rendimiento de las predicciones. Una posible línea de trabajo futuro sería la elaboración de un sistema híbrido que combine la fórmula actual con modelos predictivos de aprendizaje automático, aprovechando las fortalezas de ambos enfoques.

Por último, se contempla la posibilidad de añadir nuevas funcionalidades desde el punto de vista del usuario que mejoren la experiencia y utilidad de la aplicación. Entre ellas se pueden considerar algunas como la generación de informes personalizados o la incorporación de un sistema de notificaciones.

Bibliografía

[1] FEB, "Federación española de baloncesto." https://www.feb.es/tercerafeb.aspx, 2025.

[2] J. C. Montero, "Tecnología en el deporte: un futuro impulsado por la ciencia y los datos." https://newsweekespanol.com/2024/10/04/tecnologia-deporte-ciencia-datos/, 2024.

[3] Data4Basket, "Introducción a la estadística avanzada en el baloncesto: Claves para entender el juego." https://data4basket.com/blog/introducci%C3%B3n-a-la-estad%C3%ADstica-avanzada-en-el-baloncesto/, 2025.

[4] M. Lewis, "It's a hard-knock life: Game load, fatigue, and injury risk in the national basketball association." https://pmc.ncbi.nlm.nih.gov/articles/PMC6107769/#:~:text=For%20professional%20athletes%2C%20injuries%20can,National%20Basketball%20Association%20(NBA)./, 2018.

[5] S. F. Perrone, "La ciencia en el parqué: cómo la estadística avanzada ha revolucionado la forma de ver el baloncesto." https://medium.com/fotoperiodismoumh/la-ciencia-en-el-parqu%C3%A9-c%C3%B3mo-la-estad%C3%ADstica-avanzada-ha-revolucionado-la-forma-de-ver-el-b671535916a9/, 2023.

[6] A. Monje, "La estadística avanzada en el baloncesto: diccionario de conceptos, explicaciones y utilidades, por andrés monje." https://www.gigantes.com/nba/la-estadistica-avanzada-en-el-baloncesto-diccionario-de-conceptos-explica 2019.

[7] NBA, "Nba advanced stats." https://www.nba.com/stats/, 2025.

[8] PyPI, "nba_api." https://pypi.org/project/nba_api/, 2025.

[9] E. Basketball, "Euroleague team and player stats." https://www.euroleaguebasketball.net/es/euroleague/stats/, 2025.

[10] ACB, "Acb.com." https://www.acb.com/, 2025.

[11] NBN23, "Swish app estadísticas de baloncesto amateur." https://nbn23.com/es/swish/, 2025.

[12] FEB, "Federación española de baloncesto." https://www.feb.es/inicio.aspx, 2025.

[13] Cloudflare, "¿qué es el scraping de datos?." https://www.cloudflare.com/es-es/learning/bots/what-is-data-scraping/, 2025.

[14] PyPI, "Requests." https://pypi.org/project/requests/, 2025.

[15] PyPI, "Pandas." https://pypi.org/project/pandas/, 2025.

[16] PyPI, "pdfplumber." https://pypi.org/project/pdfplumber/, 2025.

[17] PyPI, “Pypdf2.” https://pypi.org/project/PyPDF2/, 2025.

[18] PyPI, “camelot-py.” https://pypi.org/project/camelot-py/, 2025.

[19] PyPI, “Pdfminer.” https://pypi.org/project/pdfminer/, 2025.

[20] P. documentation, “pickle — python object serialization.” https://docs.python.org/3/library/pickle.html, 2025.

[21] D. R. Framework, “Django rest framework.” https://www.django-rest-framework.org/, 2025.

[22] Django, “Django.” https://www.djangoproject.com/, 2025.

[23] PostgreSQL, “Postgresql: The world’s most advanced open source database.” https://www.postgresql.org/, 2025.

[24] MySQL, “Mysql.” https://www.mysql.com/, 2025.

[25] SQLite, “Sqlite.” https://sqlite.org/, 2025.

[26] Angular, “Angular.” https://angular.dev/, 2025.

[27] Node.js, “Node.js.” https://nodejs.org/es, 2025.

[28] J. Segura, “El trabajo de scouting (por juanjo segura).” https://viveelbasket.blogspot.com/2020/03/el-trabajo-de-scouting.html, 2020.

[29] G. Spri, “Sistemas de predicción. machine learning.” https://www.spri.eus/es/teics-comunicacion/sistemas-de-prediccion-machine-learning/, 2014.

[30] Geekflare, “Explicación de la regresión frente a la clasificación en el aprendizaje automático.” https://geekflare.com/es/regression-vs-classification/, 2025.

[31] Scikit-learn, “scikit-learn: machine learning in python — scikit-learn 1.7.0.” https://scikit-learn.org/stable/, 2025.

[32] IBM, “¿qué es la regresión logística?.” https://www.ibm.com/es-es/think/topics/logistic-regression, 2025.

[33] IBM, “¿qué es el algoritmo de k vecinos más cercanos?.” https://www.ibm.com/es-es/think/topics/knn, 2025.

[34] IBM, “¿qué es un árbol de decisión?.” https://www.ibm.com/es-es/think/topics/decision-trees, 2025.

[35] IBM, “¿qué es un bosque aleatorio?.” https://www.ibm.com/es-es/think/topics/random-forest, 2025.

[36] IBM, “What is gradient boosting?.” https://www.ibm.com/think/topics/gradient-boosting, 2025.

[37] IBM, “¿qué son las redes neuronales?.” https://www.ibm.com/es-es/think/topics/neural-networks, 2025.

[38] Encord, “F1 score in machine learning explained.” https://encord.com/blog/f1-score-in-machine-learning/, 2025.

[39] TypeScript, “Typescript: Javascript with syntax for types..” https://www.typescriptlang.org/, 2025.

[40] Angular, “Getting started with standalone components.” https://v17.angular.io/guide/standalone-components, 2025.

[41] Matplotlib, “Matplotlib — visualization with python.” https://matplotlib.org/, 2025.


UÁM
Universidad Autónoma
de Madrid