UNIVERSIDAD INTERNACIONAL DE LA RIOJA

Universidad Internacional de La Rioja

Escuela Superior de Ingeniería y Tecnología

Máster Universitario en Análisis y Visualización de Datos Masivos

Sistema de recomendación de jugadores de baloncesto complementarios

Trabajo Fin de Máster

Tipo de trabajo: Desarrollo software

Presentado por: Conde Nóvoa, Edgar

Director/a: Cervantes Rovira, Alejandro

Resumen

En este trabajo se realiza el desarrollo de una herramienta de apoyo a la toma de decisiones en el ámbito de la gestión de un equipo de baloncesto. En concreto, es una contribución que pretende presentar una solución que sea de ayuda a la hora de valorar posibles fichajes de un equipo.

El dominio sobre el que se desarrolla esta solución es la competición más importante actualmente en este deporte, la NBA. A pesar de existir tanto herramientas de análisis estadístico de jugadores y partidos como otras que sugieren traspasos de jugadores entre equipos teniendo en cuenta los presupuestos de cada franquicia, no existe ninguna con el propósito de encontrar jugadores que por sus características sean complementarios sobre la pista.

La contribución de este TFM es por tanto la descripción y construcción de un software que permite la clasificación de jugadores según sus características usando técnicas de Machine Learning como el clustering, y que al mismo tiempo analiza los partidos para ver para un determinado jugador con qué tipo de jugadores acompañándole en la pista aumenta el rendimiento del equipo. A partir de toda esa información se puede entonces sugerir jugadores de otros equipos de la liga que sean del mismo tipo que los jugadores con los que un determinado jugador es complementario, muy útil a la hora de buscar traspasos de jugadores.

Palabras Clave: Machine Learning, clustering, jugadores, baloncesto

Abstract

The purpose of this work is the development of a decision-making support tool in the field of basketball team management. Specifically, it is a contribution that intends to present a helpful solution when assessing possible signings of a team.

The domain on which the solution develops is the most important competition in this sport today, the NBA. Despite the existence of both statistical analysis tools for players and matches and others that suggest transfers of players between teams taking into account the budgets of each franchise, there is none for the purpose of finding players that by their characteristics are complementary on the court.

The contribution of this TFM is therefore the description and development of a software that allows the classification of players according to their characteristics using Machine Learning techniques such as clustering, and that at the same time analyzes the games to know for a specific player with what kind of players accompanying him on the court increases the team's performance. Based on all this information, it is then possible to suggest players from other teams in the league who are of the same type as the players with whom a certain player is complementary, very useful when looking for player transfers.

Keywords: Machine Learning, clustering, players, basketball

Índice de contenidos

Índice de tablas

Índice de figuras

1. Introducción

En los entornos deportivos se genera una gran cantidad de datos y es por eso por lo que cada vez se ven más ejemplos de aplicación del Big Data en este dominio. En los últimos años, se han desarrollado métodos muy sofisticados para la recolección de datos en el ámbito deportivo haciendo uso de todo tipo de sensores, GPS, sistemas de grabación de vídeo multicámara, etc. Todos esos datos en un principio se utilizaban mayormente para estadística descriptiva, por ejemplo, para ofrecer información al espectador en transmisiones deportivas por televisión. Pero el uso de estos datos ha evolucionado y, hoy en día, sirven para alimentar incluso a sistemas inteligentes que aportan información que ayuda a prevenir lesiones.

El deporte que constituye el dominio en el que tiene lugar el presente trabajo es el baloncesto, y en concreto para el desarrollo que se pretende realizar, la competición norteamericana NBA. Toda la información que se puede extraer de los datos que se recogen en el dominio de esta competición es útil para diseñar estrategias de juego, averiguar qué jugadores un equipo debería usar para enfrentarse a otro equipo determinado, determinar qué aspectos del juego debería entrenar un determinado jugador…en definitiva, para conocer de forma más profunda a los jugadores y a los equipos.

1.1 Justificación

Como ya se ha comentado, la tecnología hoy en día es clave en el mundo del deporte para mejorar el rendimiento de equipos y jugadores/atletas. Sin embargo, existen más decisiones que las estrictamente relacionadas con el desempeño sobre la pista. Existe un sinfín de situaciones en las que los propietarios o managers de un club toman decisiones a más alto nivel y aquí es donde cobran importancia las herramientas de ayuda a la toma de decisiones (Decision Support Tools).

En el dominio de la NBA, la toma de decisiones a la hora de realizar traspasos de jugadores entre equipos es crucial, mucho más que en otras competiciones debido a ciertas características de esta competición que se exponen en el Capítulo 2. A menudo muchos traspasos que a priori son beneficiosos para un equipo resultan siendo lo contrario debido a que normalmente lo único que se tiene en cuenta para incorporar a un jugador es su rendimiento anterior en otros equipos y las limitaciones económicas del club. Obviamente, a la hora de buscar un jugador a incorporar al equipo, los responsables tienen en cuenta que cuanto mayor el grado de afinidad de ese jugador con los ya presentes en el equipo, mejor.

Aunque se puede llegar a intuir ese grado de afinidad mediante la observación de muchos partidos de los jugadores por parte de ojeadores con un gran conocimiento táctico del juego, a menudo vemos ejemplos de traspasos de jugadores en la NBA que evidencian que ese método conduce frecuentemente a una mala decisión.

Es útil por tanto construir una solución que aporte, basándose en datos, el elemento que falta a la ecuación que determina el éxito de la incorporación de un nuevo jugador: el grado de complementariedad de ese jugador con los ya pertenecientes al equipo.

De este modo, entrando más en detalle, la solución que se propone supone una herramienta de ayuda que aporta información relevante al usuario, quien a partir de esa información puede apreciar conclusiones claras. Un par de ejemplos de conclusiones podrían ser:

- Los jugadores de posición alero o escolta y de un perfil tirador (jugadores que tienen el balón poco tiempo en sus manos, se limitan a recibirlo y lanzar a canasta) se complementan bien con jugadores de posición base y perfil organizador (jugadores que tienen el balón mucho tiempo en sus manos y que se lo hacen llegar a sus compañeros para que finalicen la jugada).
- Jugar con un jugador en la posición ala-pívot de perfil tirador (realiza tiros exteriores, por tanto, se posiciona lejos del aro) y otro en la de pívot con similares características, da un mal resultado ya que, al posicionarse estos dos jugadores (son normalmente los dos de más envergadura del equipo) lejos del aro, la capacidad del equipo para hacerse con rebotes disminuye.

1.2 Planteamiento del trabajo

En este trabajo de fin de máster se pretende desarrollar un software que ayude en la toma de decisiones a la hora de realizar traspasos de jugadores entre equipos. Como ya se ha comentado, conocer el grado de complementariedad de los jugadores de otros equipos con los de nuestro club sería muy útil para establecer a esos jugadores como objetivos de una potencial operación de traspaso.

Dada la complejidad de buscar jugadores que maximicen el grado de complementariedad con todos los jugadores de un equipo y dado que normalmente lo que buscan los equipos de la NBA es rodear a sus mejores jugadores de otros con los que se acoplen bien, lo que se busca por tanto es una solución que, para un determinado jugador de un equipo, sugiera jugadores que sean complementarios a él.

Para llevar a cabo la solución propuesta, es imprescindible conocer qué tipo de compañeros son con los que mejor se desempeña en la pista el jugador, por tanto, un objetivo será el de estudiar partidos pasados para averiguar con qué tipo de compañeros acompañando a ese jugador el equipo obtenía mejores resultados. Para acometer esta tarea, lo que primero se debe abordar es la clasificación de los jugadores. Se obtiene la información de los jugadores con todas sus características del sitio web 2kratings.com (2K Ratings, 2017), utilizando técnicas de web scraping mediante el uso de la librería beautifulsoup4 para el lenguaje Python (Python Software Foundation, 2021). Se dispondrá entonces de un conjunto de datos de alrededor de unos 500 jugadores, sobre el cuál se aplican técnicas para que esos datos sean normalizados antes de enviarse como entrada a un algoritmo de clustering (Geron, 2020). Dado que muchos jugadores pueden jugar en dos posiciones distintas, la información de los jugadores genera en torno a 900 documentos a almacenar en una base de datos no relacional. Cada documento contará entonces con los datos que identifiquen al jugador, todos los atributos relativos a las características físicas, técnicas y tácticas normalizados, e indicadores que nos proporcionen el grado de pertenencia de ese jugador a cada uno de los *n* clústeres que hemos indicado como hiperparámetro al algoritmo de clustering.

Posteriormente habrá que analizar una muestra de partidos de las últimas temporadas. Para este desarrollo se dispone de toda la información jugada a jugada de todos los partidos de la NBA desde la temporada 2015-1016 hasta el 20 de enero de la temporada 2020-2021 (Schmadamco, 2021). La muestra es más que suficiente para este desarrollo ya que cada uno de los 30 equipos juega un mínimo de 82 partidos por temporada. De estos partidos, la información a extraer será todas las combinaciones posibles de pares de jugadores que hayan jugado juntos en un mismo equipo, calculando para cada par los minutos que han compartido sobre la pista y el resultado parcial acumulado de su equipo durante ese tiempo.

Finalmente, se utilizará esta información que se ha obtenido en los pasos previos para encontrar jugadores de otros equipos que reúnan características similares a esos que al acompañar a un jugador concreto en la pista causan un aumento del rendimiento del equipo. Es decir, como paso final en el desarrollo de la solución y para servir también como herramienta para probar el mismo, se construirá una visualización que para un jugador concreto que tomamos como entrada, presente (recomiende) los jugadores más compatibles con él.

El planteamiento comentado (técnicas, conjuntos de datos, visualización, etc.) se describe de forma más detallada en el Capítulo 4.

1.3 Estructura de la memoria

En los anteriores apartados de este capítulo se ha introducido la motivación del presente trabajo. A continuación, se indica brevemente el contenido de los apartados de esta memoria:

En el Capítulo 2, “Contexto y estado del arte”, se realiza una descripción del dominio sobre el cuál se desarrolla la solución propuesta en este trabajo, así como se presentan herramientas similares, pero de distinto propósito, ya existentes.

Posteriormente, en el Capítulo 3, “Objetivos concretos y metodología”, se describen los objetivos que persigue este trabajo. Además, se indica la metodología seguida para el desarrollo de la solución, dividida en fases.

Más adelante, en el Capítulo 4, “Descripción de la herramienta software desarrollada”, se explican detalladamente los pasos seguidos durante el desarrollo de la solución propuesta en este trabajo. En primer lugar, se identifican los requisitos funcionales y no funcionales que debe cumplir el software objeto de desarrollo. Posteriormente, se describen de forma pormenorizada todas las tareas llevadas a cabo, desde la selección de herramientas para el desarrollo hasta la construcción de una interfaz para la solución; pasando por la obtención, limpieza y transformación de datos, la aplicación de algoritmos de machine learning, el despliegue del software, etcétera.

El Capítulo 5, “Evaluación”, expone la valoración realizada sobre el software resultado de este trabajo. Se evalúa el cumplimiento de los requisitos que se identifican en el Capítulo 4, así como se realiza una valoración del software por parte de usuarios externos.

Finalmente, el Capítulo 6, “Conclusiones y trabajo futuro”, presenta conclusiones surgidas de la realización del presente trabajo de fin de máster. Así mismo, se tratan en este apartado las líneas de trabajo futuras sobre el desarrollo presentado.

El presente documento se cierra con la justificación del cumplimiento del RGPD (Capítulo 7) y un listado bibliográfico (Capítulo 8).

2. Contexto y estado del arte

Durante los últimos años se ha producido un incremento exponencial en el uso, ya no sólo de sistemas de información, sino de la Inteligencia Artificial como una herramienta de apoyo en el ámbito deportivo. Se usa como un elemento facilitador para la toma de decisiones en todos los niveles del deporte, tanto para analizar las tácticas de los equipos/atletas en pista como para sugerir estrategias a más alto nivel para la gestión de un club deportivo, existiendo incluso herramientas enfocadas al uso del Big Data para la prevención de lesiones.

La solución que se desarrolla en este trabajo tiene un dominio concreto, que es la competición estadounidense de baloncesto NBA. Por lo tanto, se considera necesario realizar una descripción de algunos detalles de este dominio:

La NBA es una competición de baloncesto que se desarrolla en Estados Unidos. A pesar de ser una competición muy similar a otras de otros países u otros deportes, tiene ciertas características particulares que aportan más motivación al propósito de este trabajo.

En la NBA los equipos son considerados franquicias de una organización (la NBA) que establece una serie de normas a nivel administrativo. De este modo, y a diferencia de la inmensa mayoría de competiciones de cualquier deporte, las franquicias cuentan con un límite salarial para abordar los contratos de los empleados (entre los cuales se encuentran los jugadores). Si una franquicia sobrepasa este límite es multada severamente. Además, cuanto mayor sea el gasto sobre el límite, el importe de la multa crece, por lo que todos los equipos vigilan muy de cerca su economía para intentar maximizar la calidad de los recursos que pueden contratar dentro del límite salarial.

En cuanto a las contrataciones de jugadores, las franquicias NBA no pueden fichar directamente a jugadores de otro equipo. Únicamente pueden ofrecer contratos a jugadores que se encuentren sin equipo en ese momento, pero esos jugadores no suelen ser sus objetivos de contratación. La forma que tienen las franquicias de obtener jugadores es intercambiándolos por otros jugadores de otras franquicias, como si de un intercambio de cromos se tratase (donde los cromos son los contratos de los jugadores). Estos intercambios se convierten a veces en obras de ingeniería financiera, ya que las franquicias vigilan su límite salarial y frecuentemente se dan traspasos en los que para obtener a un jugador de alto nivel (y por ende elevado contrato) una franquicia da a cambio tres o cuatro jugadores de menor nivel para deshacerse de la masa salarial que van a absorber con el

nuevo jugador que llega. También se producen traspasos en los que llega a intervenir una tercera o incluso una cuarta franquicia como facilitadoras para llevar a cabo traspasos complejos.

Todo este sistema evidencia la importancia que tiene sobre el éxito deportivo de una franquicia el realizar los correctos traspasos de jugadores para construir el mejor equipo posible. Como existe un límite salarial, las franquicias no son capaces de tener en su plantilla una gran cantidad de jugadores de primer nivel, por lo que en esta competición en concreto cobra vital importancia el conseguir reunir en un equipo a jugadores que sean complementarios, cuyas características no hagan que se solapen, sino que sumen al rendimiento del equipo.

No existe actualmente una aplicación que proporcione esta funcionalidad, que es la de obtener jugadores complementarios a un jugador determinado de un equipo, pero sí existen algunas herramientas relacionadas tanto con el análisis estadístico de jugadores/partidos de la NBA, como con la recomendación de traspasos según el espacio salarial de las franquicias. Se describe a continuación la solución más importante para cada uno de los dos propósitos mencionados:

Basketball Reference y Stathead

Ambas son herramientas de Sports Reference LLC. En concreto, Basketball Reference (Sports Reference LLC, 2021) es la plataforma dónde se puede consultar la ficha de un jugador con sus estadísticas (Figura 1) y los registros jugada a jugada de todos los partidos (Figura 2). Por otro lado, la herramienta web Stathead (Sports Reference LLC, 2020) es el sitio web por excelencia para la consulta de estadísticas avanzadas de la NBA. En ella se puede consultar todo el histórico estadístico de la competición desde sus inicios, y proporciona datos tanto por partido como temporada. Permite realizar comparativas entre jugadores y encontrar datos interesantes como rachas de jugadores y de equipos. Por ejemplo, en la Figura 3 podemos ver el resultado de una búsqueda de las mayores rachas de partidos consecutivos de un jugador anotando 20 o más puntos durante la temporada 2020-2021. Esta herramienta es de pago y los datos que utiliza son los recogidos en la ya mencionada Basketball Reference (Sports Reference LLC, 2021).

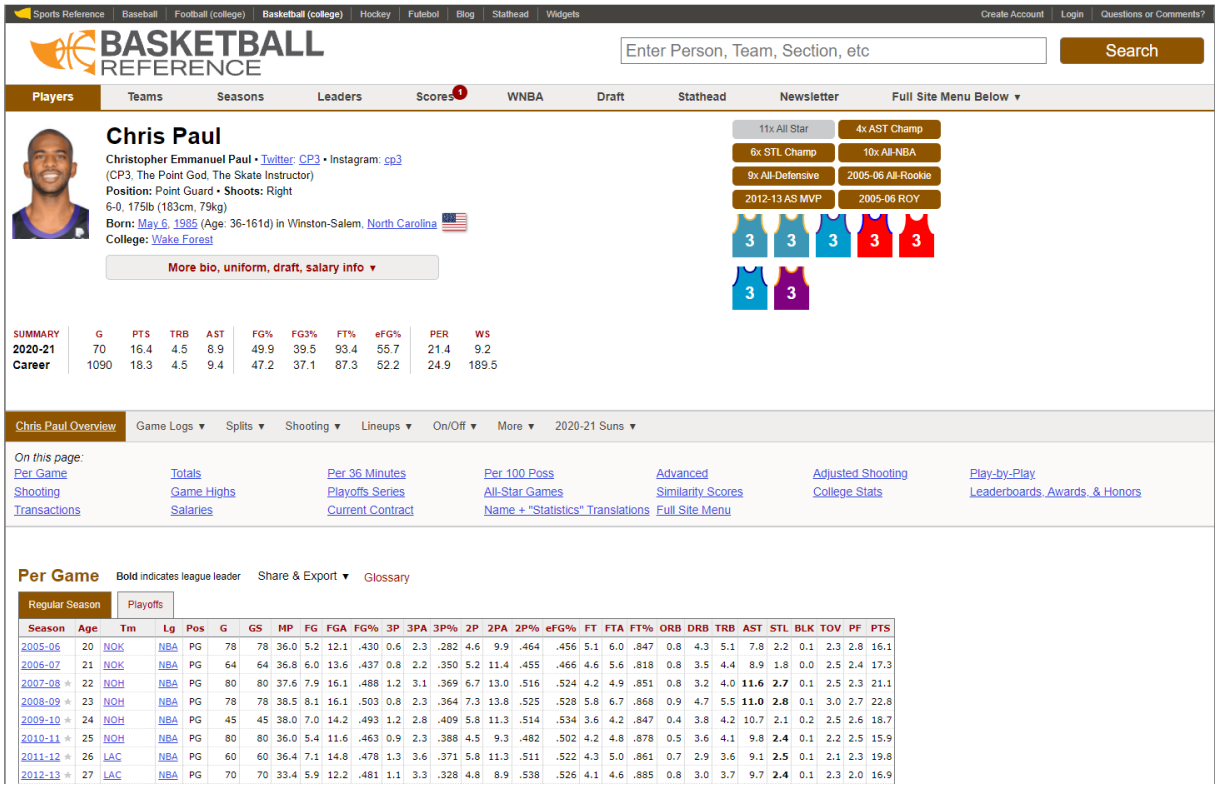

Figura 1. Estadísticas de un jugador en Basketball Reference

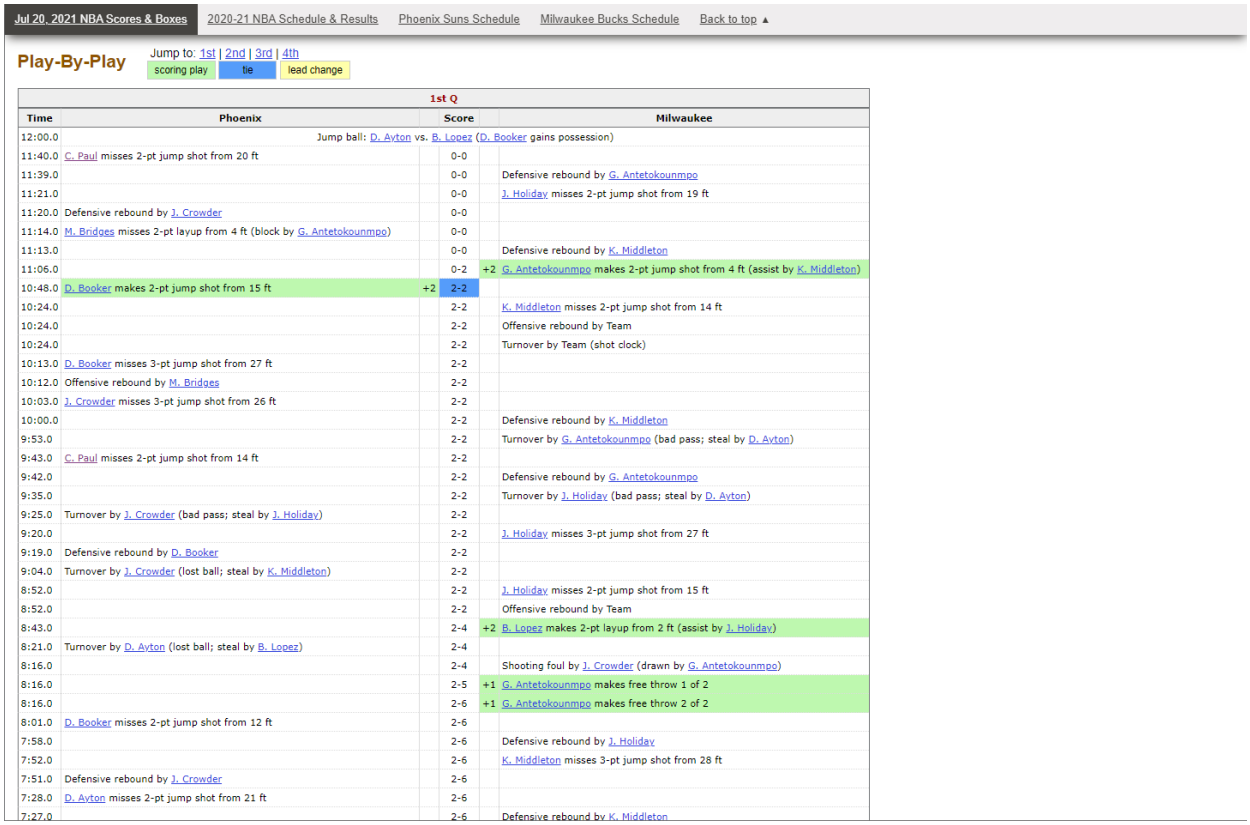

Figura 2. Datos jugada-a-jugada de un partido NBA en BasketballReference.

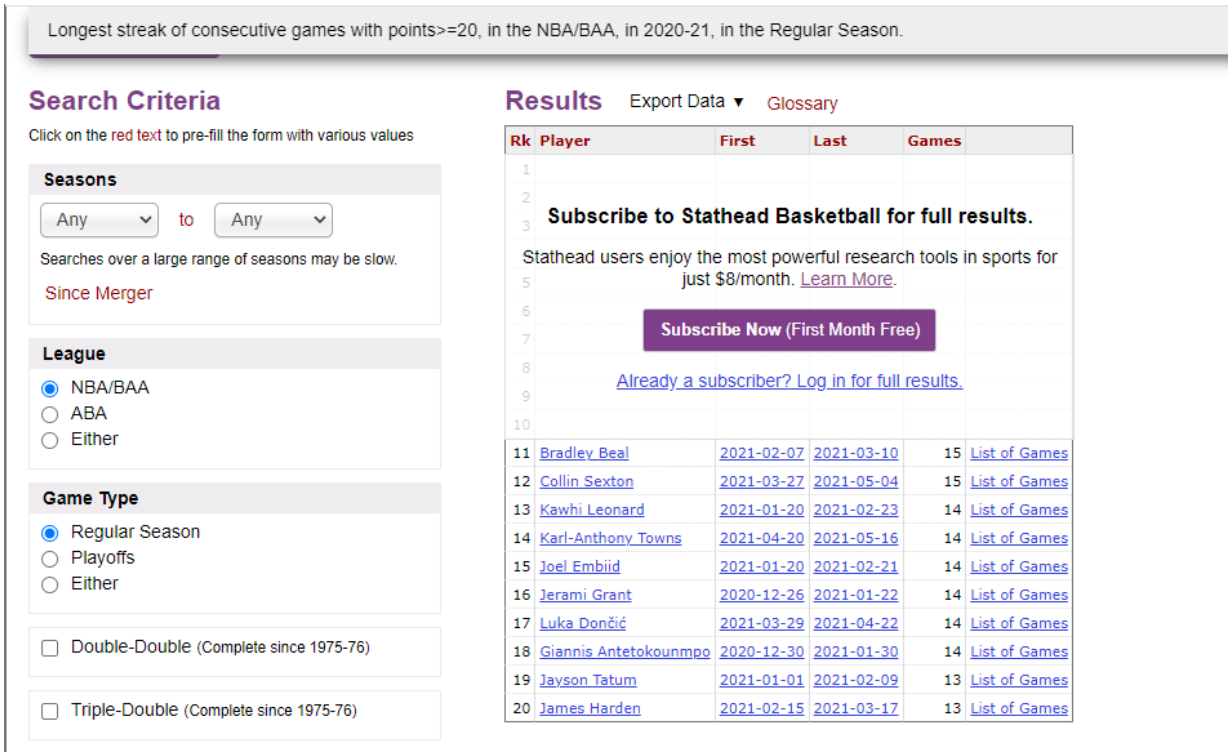

Figura 3. Búsqueda avanzada de estadísticas en Stathead

Fanspo NBA Trade Machine & Cap Manager

Esta aplicación web (Fanspo Inc., 2021) es una herramienta que actúa como una calculadora de traspasos de jugadores entre equipos, teniendo en cuenta el valor de los contratos de los jugadores, el límite salarial de las franquicias y las reglas financieras establecidas por la NBA. Es una fantástica herramienta de apoyo que determina la viabilidad de un intercambio de jugadores, cuya interfaz se muestra en la Figura 4.

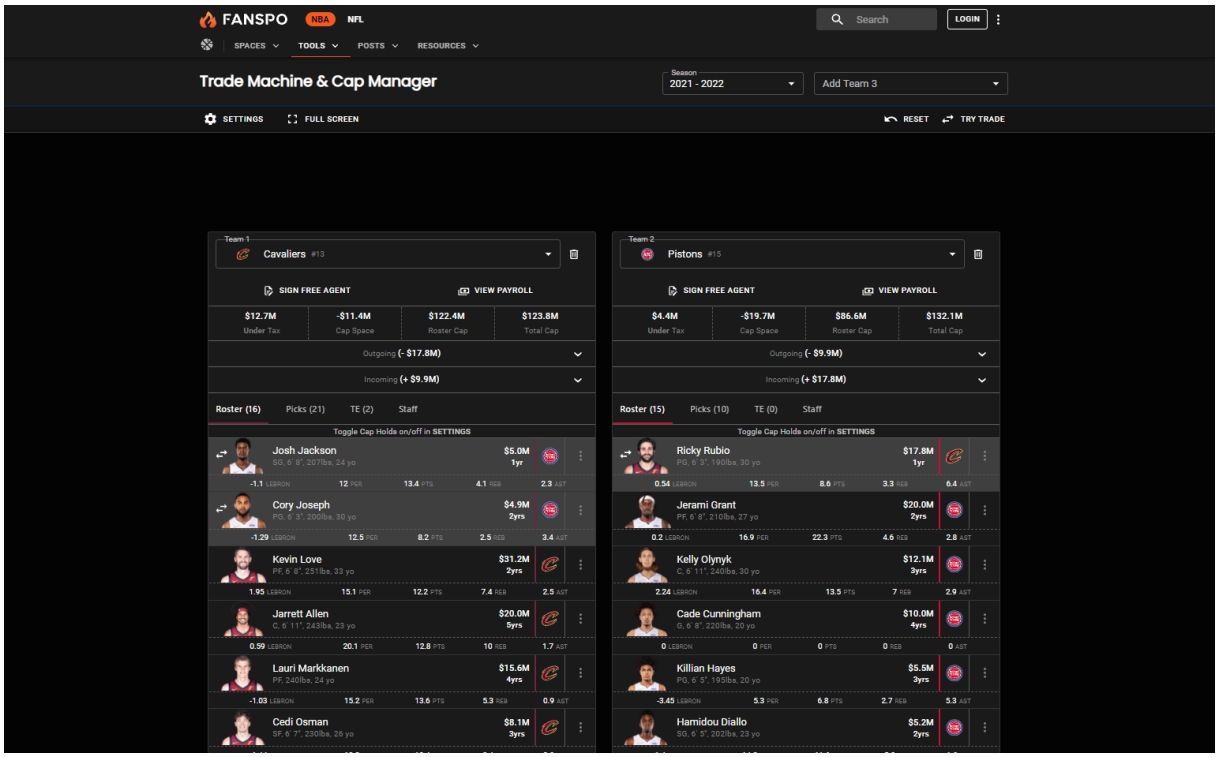

Figura 4. Fanspo NBA Trade Machine & Cap Manager

ESPN NBA Trade Machine

La aplicación web NBA Trade Machine de ESPN (ESPN Internet Ventures, 2009) es una herramienta cuya finalidad y funcionalidad son las mismas que las de la ya mencionada Fanspo NBA Trade Machine & Cap Manager. Ambas aplicaciones apenas difieren en la interfaz, aunque se le otorga más fiabilidad a esta de ESPN en cuanto a la actualidad de los datos de los contratos. La Figura 5 muestra la interfaz de esta herramienta web.

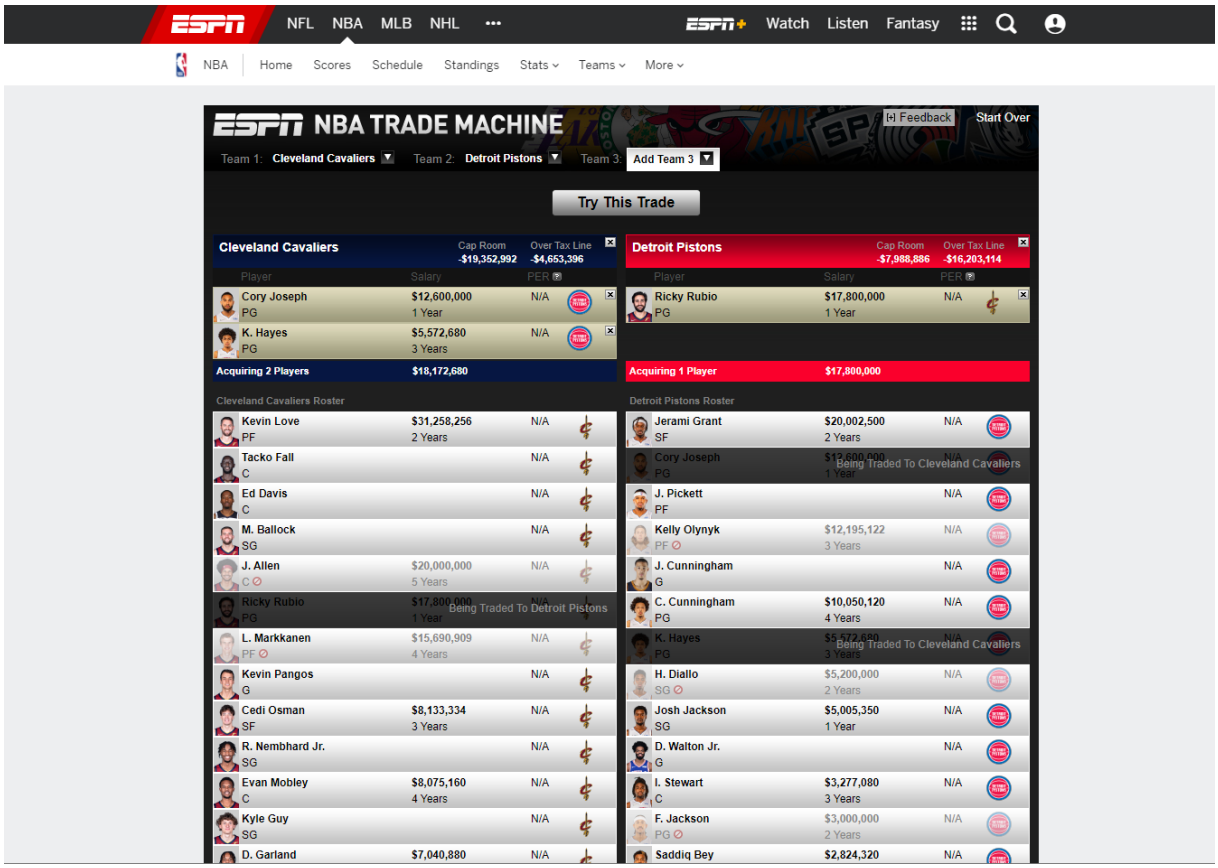

Figura 5. ESPN NBA Trade Machine.

Además de las herramientas mencionadas en el ámbito del baloncesto, existen otras en los dominios de otros deportes, sobre todo el fútbol.

Por ejemplo, las empresas Driblab (Driblab, 2021) y StatsBomb (StatsBomb Services Ltd, 2020) ofrecen servicios a clubes de fútbol para la búsqueda de jugadores con determinadas características. Uno de los motivos por los que un club se decanta por usar estos servicios porque necesita sustituir a un jugador de su equipo y quiere encontrar un jugador alternativo al que contratar que reúna las mismas características para que el cambio afecte lo mínimo posible al equipo.

El propósito de este trabajo está bastante ligado a las herramientas/servicios descritos anteriormente, pero la pregunta que se pretende responder con el desarrollo de esta solución es diferente. Este desarrollo no pretende encontrar jugadores similares a uno dado. Se pretende, para un jugador, estudiar con qué tipo de compañeros el equipo obtiene mejores resultados, y posteriormente buscar qué jugadores de ese tipo hay en otros equipos de la liga. De ese modo, un usuario puede usar una herramienta como la ya mencionada

NBA Trade Machine (ESPN Internet Ventures, 2009) y estudiar la viabilidad de traspasos en los que el equipo obtenga los jugadores sugeridos por la solución que se describe en el presente trabajo.

Para poner fin a este capítulo, se sintetizan a continuación en la Tabla 1 las características de las herramientas/servicios estudiados y del software a desarrollar en este trabajo.

Tabla 1. Resumen de herramientas y servicios de análisis de datos en el deporte

Herramienta / empresa	**¿Qué aporta?**	**Funcionalidad / servicio principal**
○ Basketball Reference	Datos y estadística básica de la NBA.	Estadísticas promedio y totales de los jugadores y registro de datos jugada a jugada de los partidos.
○ Stathead	Estadística y búsqueda avanzada de información de la NBA.	Búsqueda sofisticada (rachas de jugadores o equipos, comparativas de jugadores, etc).
○ Fanspo NBA Trade Machine & Cap Manager ○ ESPN Trade Machine	Apoyo a la toma de decisiones en traspasos de jugadores en la NBA.	Estudio de la viabilidad de traspasos de jugadores según las reglas financieras de la NBA.
○ Driblab ○ StatsBomb	Servicio de consultoría Big Data para fútbol que puede sustituir a una red de ojeadores.	Búsqueda de jugadores que cumplan ciertos patrones que les hagan ser objetivo de un fichaje.
○ Este trabajo (Sistema de recomendación de jugadores de baloncesto complementarios)	Información extraída de datos en crudo de jugadores y partidos de la NBA, útil como apoyo a la toma de decisiones en traspasos.	Encontrar jugadores con un alto grado de compatibilidad con uno dado.

3. Objetivos concretos y metodología de trabajo

En este apartado se describen los objetivos que se pretenden cumplir con la elaboración del presente trabajo. Se explica el objetivo general, se detallan los objetivos específicos y se explica la metodología seguida para el desarrollo del software y la consecución de los objetivos descritos.

3.1. Objetivo general

El objetivo principal de este proyecto es aportar una solución que apoye en la toma de decisiones a usuarios de alto nivel dentro del dominio de la gestión de un club de baloncesto. En concreto, el objetivo es ayudar en la búsqueda de traspasos de jugadores que sean beneficiosos para un determinado equipo.

Cuando se conoce qué tipos de jugadores elevan el rendimiento de un equipo al jugar juntos, se pueden buscar jugadores del mismo tipo en otros equipos para intentar incorporarlos al equipo a través de traspasos.

Para la consecución del objetivo se utilizarán técnicas de IA (algoritmos de machine learning) y de recolección, limpieza y transformación de datos.

3.2. Objetivos específicos

De forma más concreta, los objetivos específicos que persigue este trabajo son los siguientes:

1. Identificar los orígenes de datos necesarios.
2. Explorar los conjuntos de datos de los que se dispone y aplicar técnicas de limpieza y transformación sobre los mismos.
3. Crear una base de datos para la persistencia de los datos.
4. Analizar qué técnicas de IA (en concreto qué algoritmos de machine learning) son las adecuadas a las necesidades del proyecto y seleccionar una.
5. Identificar las herramientas y tecnologías idóneas para llevar a cabo las tareas que constituyen el proyecto.
6. Evaluar la solución propuesta, valorando los resultados que ofrece.
7. Construir un pequeño dashboard que permita probar la solución de forma visual.

3.3. Metodología del trabajo

Para cumplir los objetivos establecidos anteriormente se divide el proyecto en diferentes fases:

1. **Identificación de requisitos**: En esta fase se establecen los requisitos tanto funcionales como no funcionales que debe cumplir el software que se pretende desarrollar.
2. **Selección de herramientas y obtención, limpieza y transformación de los conjuntos de datos:** Durante esta fase se analizan y eligen las herramientas a utilizar en las tareas de ésta y de las posteriores fases del proyecto. Además, haciendo ya uso de alguna de esas herramientas, se obtienen los datos de los diferentes orígenes contemplados, se aplican técnicas de limpieza sobre los mismos y se realizan las transformaciones necesarias.
3. **Creación del modelo de machine learning para el clustering de jugadores y construcción de la base de datos:** En esta fase se crea un modelo de machine learning que utiliza un algoritmo de clustering para clasificar jugadores en grupos con características similares. A continuación, se construye la base de datos y se vuelcan en ella los datos relativos a los jugadores.
4. **Extracción de información del rendimiento por parejas de jugadores en partidos:** Durante esta fase se toma el conjunto de datos relativo a los partidos (Schmadamco, 2021) y se extrae de ellos la información acerca de la cantidad de minutos compartidos en pista por cada par de jugadores posible, junto con el resultado parcial acumulado para su equipo durante ese tiempo.
5. **Construcción de una interfaz para la solución y despliegue:** En esta fase se elabora una pequeña visualización con el objetivo de poder probar la solución a través de una interfaz y consultar los resultados que ofrece.
6. **Evaluación de la solución, conclusiones y descripción de futuras líneas de trabajo**: En esta última fase se realiza una evaluación del cumplimiento de los objetivos descritos anteriormente en el presente capítulo. La evaluación será llevada a cabo por el propio autor del trabajo, así como por dos colaboradores afines al dominio del trabajo que serán encuestados sobre la herramienta desarrollada y los resultados que ofrece. El presente proyecto se da por finalizado en esta fase exponiendo conclusiones y describiendo líneas de trabajo futuras.

Las tareas mencionadas anteriormente en los puntos 1 al 5 son explicadas en el Capítulo 4, en el cual se describe en detalle el proceso de desarrollo del software. La evaluación de la

solución se presenta en el Capítulo 5. Las conclusiones obtenidas del presente trabajo, así como las líneas de trabajo futuro, se detallan en el Capítulo 6.

4. Descripción de la herramienta software desarrollada

Los apartados que componen el presente capítulo describen en detalle el proceso de desarrollo llevado a cabo para la implementación de la solución aportada en este trabajo. El código y el resto de los artefactos que componen la solución a los que se hará referencia se pueden consultar en detalle en el repositorio de GitHub del trabajo (Conde, 2021).

A continuación, y a modo de introducción a este capítulo, la Figura 6 resume el proceso de extracción, limpieza, transformación y carga de datos llevado a cabo, mientras que la Figura 7 ilustra de forma simple y a un alto nivel la arquitectura del software que se ha desplegado para interactuar con la solución desarrollada en el proyecto.

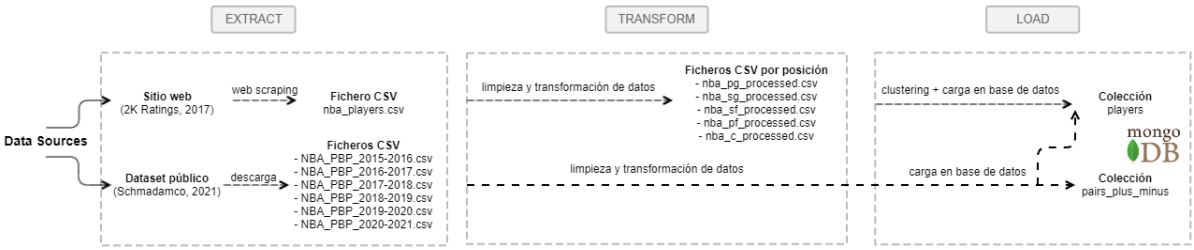

Figura 6. Resumen del proceso ETL llevado a cabo

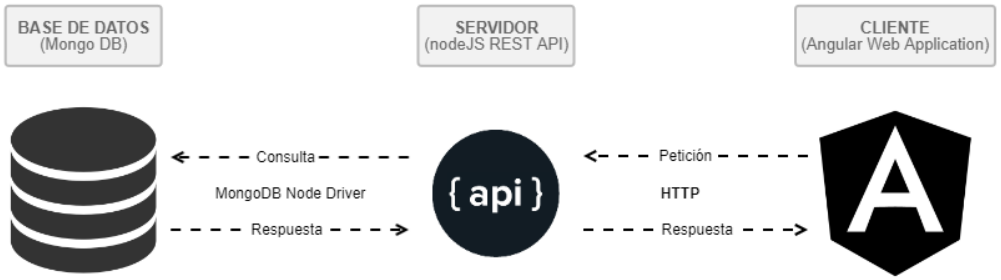

Figura 7. Arquitectura software de la aplicación desplegada

4.1. Identificación de requisitos

Como se ha comentado en capítulos anteriores, el propósito de este trabajo es realizar un desarrollo que permita a un usuario que lo utilice obtener información acerca de la compatibilidad entre jugadores de baloncesto. A continuación, en las tablas Tabla 2 y Tabla 3 se exponen los requisitos que debe cumplir la solución a nivel funcional y no funcional.

Tabla 2. Requisitos funcionales

Requisitos funcionales		
Id	Tipo	Descripción
F1	Interfaz	La interfaz deberá permitir al usuario consultar la información de cualquier jugador, presentando un listado sobre el cuál se puedan realizar búsquedas por nombre de jugador o nombre de equipo.
F2	Interfaz	El usuario deberá seleccionar en la interfaz valores para los parámetros necesarios para realizar una recomendación de jugadores compatibles.
F3	Interfaz	Los parámetros de entrada para la recomendación serán: Jugador, Posición de los jugadores a recomendar
F4	Interfaz	Los valores posibles para los parámetros de entrada serán seleccionables desde la interfaz.
F5	Lógica	La recomendación será un listado de jugadores que jueguen en la posición que se ha indicado como parámetro de entrada, acompañados por su índice de compatibilidad con el jugador indicado como parámetro de entrada, su valoración en un rango del 0 al 100 y su índice de pertenencia al clúster con el que el jugador indicado como parámetro de entrada tiene mayor compatibilidad.
F6	Lógica	El software implementará la lógica necesaria para obtener y generar los atributos de los jugadores necesarios para la solución.
F7	Acceso a datos	La base de datos deberá almacenar, para todos los jugadores, los valores para sus atributos baloncestísticos (físicos y técnicos).
F8	Acceso a datos	La base de datos deberá almacenar, para todos los jugadores, campos que indiquen qué tipo de jugador es y campos que indiquen su compatibilidad con cada uno de los tipos de jugadores.

Tabla 3. Requisitos no funcionales

Requisitos no funcionales		
Id	Tipo	Descripción
NF1	Interfaz	La interfaz será ejecutable en los navegadores Chrome y Firefox.
NF2	Interfaz	La interfaz de la solución deberá ser intuitiva, sencilla de usar.
NF3	Interfaz	Los resultados que proporcione la interfaz deberán presentarse de forma que interpretados sin ninguna dificultad por el usuario.
NF4	Lógica	Las recomendaciones deberán generarse en un tiempo inferior a 5 segundos.
NF5	Lógica	El servidor que sirva la solución deberá permitir el acceso concurrente a la misma de al menos 10 usuarios.
NF6	Acceso a datos	La base de datos donde se persistan los datos con los que trabajará la solución deberá estar configurada para controlar accesos no deseados a la misma.

4.2. Selección de herramientas y obtención, limpieza y transformación de los conjuntos de datos

Selección de herramientas

Al principio de un desarrollo, una tarea obligatoria es la de seleccionar las herramientas a utilizar durante el mismo. Para este caso en concreto, ha sido necesario buscar herramientas para acometer las tareas relativas a la obtención, limpieza y transformación de los datos, además de escoger en qué tipo de base de datos almacenar los datos.

Tras valorar muchas de las opciones disponibles, se ha utilizado el lenguaje Python para escribir los scripts encargados de obtener, limpiar y transformar los datos con los que se ha trabajado, así como de persistirlos en una base de datos. El editor de código utilizado ha sido Visual Studio Code (Microsoft, 2021) por varios motivos: ligereza, compatibilidad con Jupyter Notebooks, depurador, etc.

En cuanto a la base de datos, se ha optado por MongoDB (MongoDB, Inc., 2009), una base de datos NOSQL cuyas características se alinean bien con el tratamiento que se hace de los datos en este trabajo. MongoDB cuenta con una documentación amplia y detallada, y es ideal para entornos con pocos recursos de computación. Además, ofrece un gran

rendimiento en aplicaciones en las que la mayoría de las operaciones son de lectura y sin necesidad de hacer operaciones de tipo *Join*, como es el caso de este trabajo.

Obtención de datos

Como se ha comentado en capítulos anteriores, para este desarrollo se ha dispuesto de datos tanto de los jugadores como de partidos de la NBA.

El conjunto de datos relativo a los partidos de la NBA se ha obtenido de la plataforma Kaggle (Schmadamco, 2021), y se trata de un dataset público que contiene los datos jugada a jugada de todos los partidos de la NBA desde la temporada 2015-2016 hasta el 20 de enero de la temporada 2020-2021. Este dataset está conformado por 5 archivos CSV que suman un total de 665 MB de información. Los datos que contiene cada registro se corresponden con cada una de las jugadas registradas en la web Bastekball Reference (Sports Reference LLC, 2021). En la Figura 8 se puede ver cómo se han traducido esos datos a registros en un fichero CSV., mientras que la Figura 9 muestra el registro jugada a jugada de un partido entre Golden State y Brooklyn en Basketball Reference.

URL	Game Type	Location	Date	Time	Winning Team	Quarter	Sec Left	Away Team	Away Play	Away Score	Home Team	Home Play	Home Score	Shooter
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	720	GSW	Jump ball: J. Wise	0	BRK		0	
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	710	GSW		0	BRK	Turnover by D. Jo	0	
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	698	GSW	Shooting foul by	0	BRK		0	
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	698	GSW	S. Curry makes fr	1	BRK		0	
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	698	GSW	S. Curry makes fr	2	BRK		0	
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	682	GSW		2	BRK	K. Irving makes 2	2	K. Irving - irvinky
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	671	GSW	J. Wiseman make	4	BRK		2	J. Wiseman - wisi
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	649	GSW		4	BRK	K. Durant makes	5	K. Durant - durar
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	631	GSW	A. Wiggins misse	4	BRK		5	A. Wiggins - wigg
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	625	GSW		4	BRK	Defensive rebour	5	
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	623	GSW		4	BRK	K. Durant makes	7	K. Durant - durar
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	616	GSW	S. Curry misses 3	4	BRK		7	S. Curry - curryst
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	616	GSW		4	BRK	Defensive rebour	7	
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	609	GSW		4	BRK	J. Harris makes 3	10	J. Harris - harrijol
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	597	GSW	A. Wiggins misse	4	BRK		10	A. Wiggins - wigg
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	591	GSW		4	BRK	Defensive rebour	10	
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	589	GSW		4	BRK	S. Dinwiddie mis	10	S. Dinwiddie - dii
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	584	GSW	Defensive rebour	4	BRK		10	
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	578	GSW	S. Curry makes 2	6	BRK		10	S. Curry - curryst
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	570	GSW		6	BRK	J. Harris misses 3	10	J. Harris - harrijol
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	565	GSW	Defensive rebour	6	BRK		10	
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	558	GSW	J. Wiseman misse	6	BRK		10	J. Wiseman - wisi
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	554	GSW		6	BRK	Defensive rebour	10	
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	550	GSW		6	BRK	K. Irving misses 3	10	K. Irving - irvinky
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	547	GSW	Defensive rebour	6	BRK		10	
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	541	GSW	E. Paschall misse	6	BRK		10	E. Paschall - pasc
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	536	GSW		6	BRK	Defensive rebour	10	
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	526	GSW		6	BRK	K. Durant makes	12	K. Durant - durar
/boxscores/2020	regular	Barclays Center E	December 22 20	7:00 PM	BRK	1	526	GSW		6	BRK	Shooting foul by	12	

Figura 8. Datos jugada-a-jugada del partido entre Golden State y Brooklyn el 22 de diciembre de 2020 en el fichero CSV descargado de la plataforma Kaggle (Schmadamco, 2021)

Play-By-Play Jump to: 1st | 2nd | 3rd | 4th

scoring play | tie | lead change

1st Q					
Time	**Golden State**		**Score**		**Brooklyn**
12:00.0	Jump ball: J. Wiseman vs. D. Jordan (J. Harris gains possession)				
11:50.0			0-0		Turnover by D. Jordan (bad pass)
11:38.0	Shooting foul by K. Irving (drawn by S. Curry)		0-0		
11:38.0	S. Curry makes free throw 1 of 2	+1	1-0		
11:38.0	S. Curry makes free throw 2 of 2	+1	2-0		
11:22.0			2-2	+2	K. Irving makes 2-pt jump shot from 22 ft (assist by K. Durant)
11:11.0	J. Wiseman makes 2-pt dunk from 1 ft (assist by A. Wiggins)	+2	4-2		
10:49.0			4-5	+3	K. Durant makes 3-pt jump shot from 26 ft (assist by J. Harris)
10:31.0	A. Wiggins misses 3-pt jump shot from 24 ft		4-5		
10:25.0			4-5		Defensive rebound by J. Harris
10:23.0			4-7	+2	K. Durant makes 2-pt jump shot from 5 ft
10:16.0	S. Curry misses 3-pt jump shot from 27 ft		4-7		
10:16.0			4-7		Defensive rebound by D. Jordan
10:09.0			4-10	+3	J. Harris makes 3-pt jump shot from 26 ft (assist by S. Dinwiddie)
9:57.0	A. Wiggins misses 3-pt jump shot from 24 ft		4-10		
9:51.0			4-10		Defensive rebound by K. Irving
9:49.0			4-10		S. Dinwiddie misses 3-pt jump shot from 26 ft
9:44.0	Defensive rebound by K. Oubre		4-10		
9:38.0	S. Curry makes 2-pt layup from 4 ft (assist by K. Oubre)	+2	6-10		
9:30.0			6-10		J. Harris misses 3-pt jump shot from 25 ft
9:25.0	Defensive rebound by J. Wiseman		6-10		
9:18.0	J. Wiseman misses 2-pt jump shot from 22 ft		6-10		
9:14.0			6-10		Defensive rebound by K. Irving
9:10.0			6-10		K. Irving misses 3-pt jump shot from 25 ft
9:07.0	Defensive rebound by J. Wiseman		6-10		
9:01.0	E. Paschall misses 2-pt jump shot from 17 ft		6-10		
8:56.0			6-10		Defensive rebound by D. Jordan
8:46.0			6-12	+2	K. Durant makes 2-pt jump shot from 15 ft
8:46.0			6-12		Shooting foul by J. Wiseman (drawn by K. Durant)
8:46.0			6-13	+1	K. Durant makes free throw 1 of 1
8:30.0	Shooting foul by D. Jordan (drawn by A. Wiggins)		6-13		
8:30.0	A. Wiggins makes free throw 1 of 2	+1	7-13		
8:30.0	A. Wiggins makes free throw 2 of 2	+1	8-13		
8:18.0			8-13		K. Durant misses 2-pt jump shot from 11 ft
8:16.0			8-13		Offensive rebound by Team

Figura 9. Datos jugada-a-jugada del partido entre Golden State y Brooklyn el 22 de diciembre de 2020 en Basketball Reference

Los datos relativos a los jugadores no se han podido obtener con la misma facilidad que los de los partidos. No se ha podido obtener un dataset con la información que se requería sobre los jugadores para el este desarrollo, por lo que se ha tenido construir uno con datos extraídos del sitio web 2kratings (2K Ratings, 2017).

Para acometer tal tarea se ha utilizado la técnica de web scraping, haciendo uso de la librería beautifulsoup (Python Software Foundation, 2021b). Como se muestra en la Figura 10, en la página principal del sitio web 2kratings se encuentra un listado con enlaces a los 30 equipos de la NBA en el lateral izquierdo. El script que se ha desarrollado para esto, disponible en el repositorio (Conde, 2021), se encarga de navegar a través de esos enlaces para acceder a la página de cada equipo (véase Figura 11). En la página de cada equipo se encuentra el listado de enlaces a las páginas de cada uno de los jugadores. En la Figura 12

podemos ver la sección de la página de jugador que contiene la información de todos los atributos con su valoración.

Todos esos datos extraídos de la web se vuelcan en un fichero CSV conforme se van leyendo. El resultado de este proceso es el conjunto de datos que se muestra parcialmente en la Figura 13, donde existe un registro por jugador y se han almacenado todos esos atributos valorados de 0 a 100 junto con otros que indican la posición del jugador o el equipo al que pertenece.

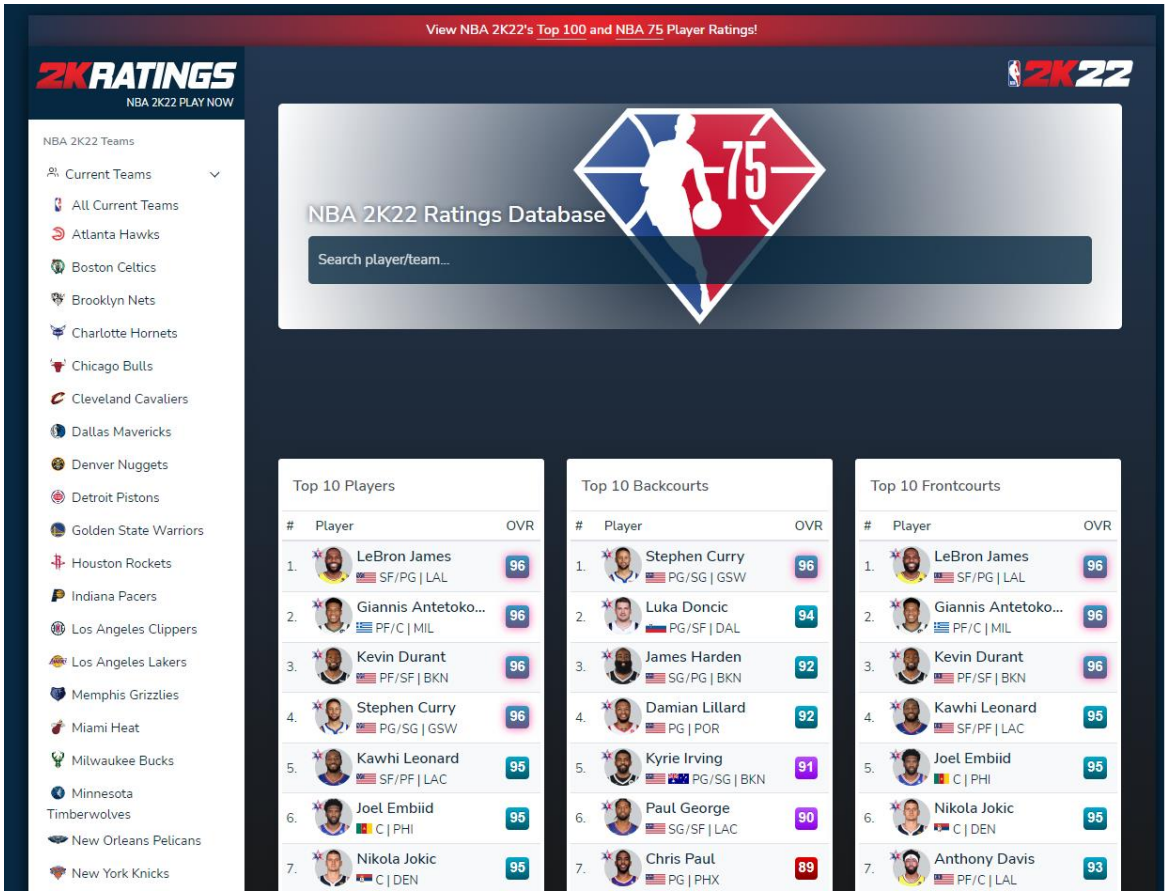

Figura 10. Página principal del sitio web 2kratings con el listado de equipos en el panel izquierdo

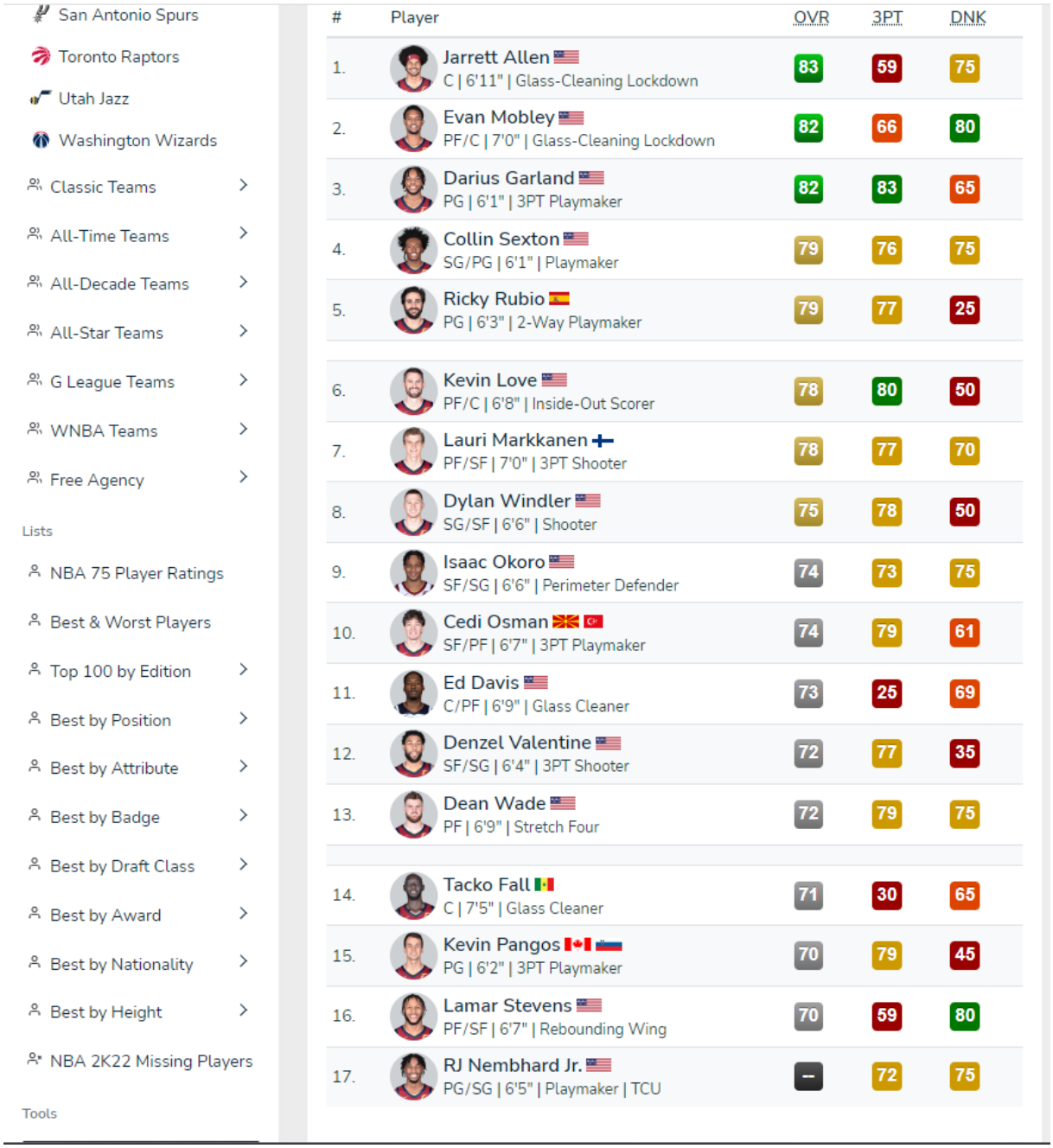

Figura 11. Página de un equipo en 2kratings con el listado de jugadores que lo componen

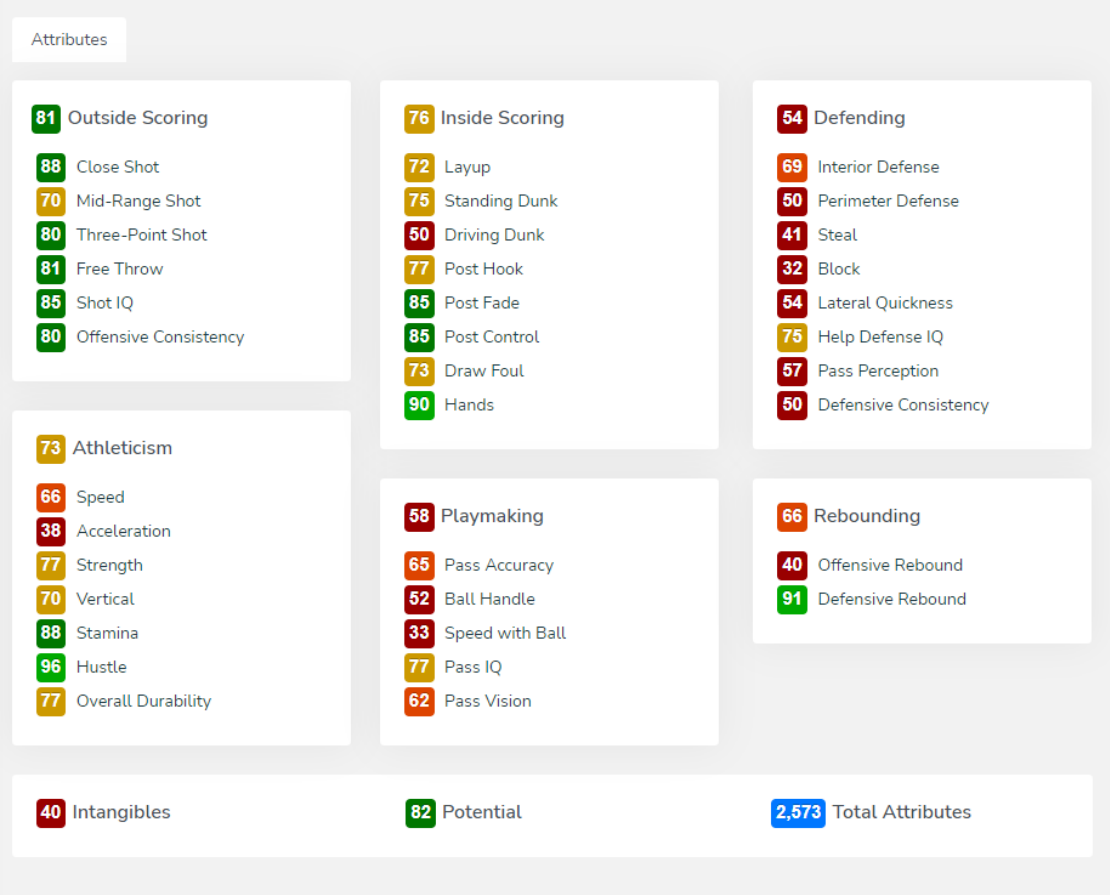

Figura 12. Página de un jugador en 2kratings con el listado de atributos valorados

Name	Team	Position1	Position2	Overall	Outside Scoring	Close Shot	Mid-Range Shot	Three-Point Shc	Fre
Trae Young	Atlanta Hawks	PG		89	90	95	84	86	
Clint Capela	Atlanta Hawks	C		85	67	77	60	25	
John Collins	Atlanta Hawks	PF	C	84	87	94	90	81	
Bogdan Bogdanc	Atlanta Hawks	SG	SF	79	90	87	91	85	
De'Andre Hunter	Atlanta Hawks	SF	PF	79	84	85	85	78	
Cameron Reddisl	Atlanta Hawks	SF	SG	79	61	62	76	72	
Danilo Gallinari	Atlanta Hawks	PF	SF	79	84	84	79	84	
Lou Williams	Atlanta Hawks	PG	SG	76	78	77	74	82	
Kevin Huerter	Atlanta Hawks	SG	SF	76	81	88	90	82	
Delon Wright	Atlanta Hawks	PG	SG	76	80	76	65	80	
Onyeka Okongwı	Atlanta Hawks	C	PF	74	62	84	60	60	
Gorgui Dieng	Atlanta Hawks	C		74	79	85	76	78	
Jalen Johnson	Atlanta Hawks	PF	SF	73	68	74	74	69	
Timothe Luwawu	Atlanta Hawks	SF	SG	72	64	53	65	74	
Sharife Cooper	Atlanta Hawks	PG		71	69	77	73	68	
Skylar Mays	Atlanta Hawks	SG	SF	71	78	84	84	75	
Solomon Hill	Atlanta Hawks	PF	SF	70	71	77	67	72	
Jayson Tatum	Boston Celtics	PF	SF	89	89	90	85	86	
Jaylen Brown	Boston Celtics	SF	SG	87	86	89	93	84	
Robert Williams	Boston Celtics	C		80	67	93	69	25	
Al Horford	Boston Celtics	C	PF	80	80	91	77	81	
Dennis Schroder	Boston Celtics	PG	SG	79	79	75	88	77	
Enes Kanter	Boston Celtics	C		77	80	85	77	55	
Marcus Smart	Boston Celtics	SG	PG	77	67	89	79	77	
Juan Hernangom	Boston Celtics	SF	PF	75	75	69	68	78	
Payton Pritchard	Boston Celtics	PG	SG	74	85	84	73	84	
Josh Richardson	Boston Celtics	SG	SF	74	75	84	86	76	
Grant Williams	Boston Celtics	PF	C	74	64	89	70	80	

Figura 13. Dataset resultante de la ejecución del script que obtiene los datos de los jugadores de 2kratings mediante web scraping.

Limpieza y transformación de datos

Hasta este punto en el proceso de desarrollo se cuenta con dos conjuntos de datos en formato CSV, uno con la información de los partidos y otro con la información de los jugadores.

El conjunto de datos relativo a los partidos se encuentra ya en las condiciones deseadas para ser la entrada de un algoritmo que extraiga la información que queremos de los partidos, Por tanto, no se tratará este dataset en este punto, ya que no requiere que se efectúe ningún tipo de limpieza o transformación sobre los datos.

Respecto al dataset de jugadores, éste sí requiere que se apliquen algunas transformaciones sobre sus datos. Se detallan a continuación:

- Puede darse el poco común caso de que en 2kratings no exista la valoración de un atributo determinado para un jugador. Se ha comprobado que en ese caso la web muestra la cadena "—". Para evitar problemas, se busca esa cadena en el dataset y se sustituye por NaN (Not a Number) de Python. Esto permitirá que la columna

pueda ser tratada como numérica y que tampoco haya problemas cuando el algoritmo de clustering que se utilice tenga que tratar con estos datos.

- Relacionado con el punto anterior, si una columna que se supone numérica (representa la valoración entre 0 y 100 de la capacidad de jugador para capturar rebotes, por ejemplo) cuenta con un valor que no es numérico como la cadena "—", el tipo de la columna se inferirá incorrectamente, por lo que para cada columna que se refiere a un atributo valorado entre 0 y 100 se realiza la sustitución mencionada en el punto anterior y se fuerza el tipo numérico para esa columna si no se había inferido así.

- El algoritmo de clustering que se aplicará posteriormente a los datos lo hará por bloques, uno por posición (PG, SG, SF, PF y C). Es decir, se ejecutará el algoritmo de clustering sobre el subconjunto de jugadores con posición PG (base) y se obtendrá una serie de clústeres en los que se pueden clasificar los jugadores de esa posición. Lo mismo se haría para los jugadores con posición SG, SF, PF y C. Dado que cada jugador puede tener hasta dos posibles posiciones, en este punto ese detalle se tiene en cuenta y para cada jugador que pueda jugar en 2 posiciones se duplicará su registro. Por ejemplo, un jugador del dataset de entrada con los atributos Position 1 = SF y Position 2 = PF tendrá dos registros que lo representen en el dataset de salida, el primero con el atributo Position = SF y el segundo con el atributo Position = PF.

- Para que el algoritmo de clustering que queremos aplicar sobre el conjunto de datos de jugadores funcione correctamente y clasifique a los jugadores según patrones en los valores de sus atributos baloncestísticos, se deben normalizar esos valores. De no normalizarse, lo más probable es que el algoritmo de clustering clasificase a los jugadores con valores altos en esos atributos en uno, con valores bajos en otro, con valores intermedios en otro...es decir, clasificaría a los jugadores según cómo de buenos son, pero no según cómo son. La técnica que se ha decidido utilizar (se podrían pensar/aplicar otras, o combinarlas) ha consistido en tomar el valor más alto que tenga un jugador para sus atributos baloncestísticos y asignarle un 100, que es el valor máximo. El resto de los atributos se recalculan manteniendo la proporcionalidad con el valor que tenía ese atributo con el valor máximo. Es decir, para realizar esta transformación sobre los atributos baloncestísticos de un jugador, bastaría con aplicar el siguiente cálculo a cada atributo *A* del jugador, siendo *MAX* el atributo con el valor más alto para ese jugador:

$$A = A * 100/MAX$$

- Para poder identificar en fases posteriores del proyecto a qué jugador se refieren cada una de las jugadas del dataset de partidos, se ha incorporado un campo “BasketballReference Player Id” que se corresponde con el identificador único que otorga BasketballReference (Sports Reference LLC, 2021) a cada jugador. Este proceso ha resultado tedioso al haber tenido que llevarse a cabo de forma manual.

Este proceso de limpieza y transformación se lleva a cabo con la ejecución de uno de los scripts desarrollados, disponibles en el repositorio (Conde, 2021), y da como resultado un conjunto de datos de jugadores con un registro por jugador y posición, y con los atributos normalizados tal y como se puede apreciar en la Figura 14.

Name	Team	Position	Overall	Outside Scoring	Close Shot	Mid-Range Sho	Three-Point Sho	Free
Trae Young	Atlanta Hawks	PG	89	93.75	98.96	87.5	89.58	
Clint Capela	Atlanta Hawks	C	85	69.07	79.38	61.86	25.77	
John Collins	Atlanta Hawks	C	84	90.63	97.92	93.75	84.38	
John Collins	Atlanta Hawks	PF	84	90.63	97.92	93.75	84.38	
Bogdan Bogdanc	Atlanta Hawks	SG	79	94.74	91.58	95.79	89.47	
Bogdan Bogdanc	Atlanta Hawks	SF	79	94.74	91.58	95.79	89.47	
De'Andre Hunter	Atlanta Hawks	PF	79	93.33	94.44	94.44	86.67	
De'Andre Hunter	Atlanta Hawks	SF	79	93.33	94.44	94.44	86.67	
Cameron Reddisl	Atlanta Hawks	SG	79	67.78	68.89	84.44	80	
Cameron Reddisl	Atlanta Hawks	SF	79	67.78	68.89	84.44	80	
Danilo Gallinari	Atlanta Hawks	PF	79	84	84	79	84	
Danilo Gallinari	Atlanta Hawks	SF	79	84	84	79	84	
Lou Williams	Atlanta Hawks	PG	76	88.64	87.5	84.09	93.18	
Lou Williams	Atlanta Hawks	SG	76	88.64	87.5	84.09	93.18	
Kevin Huerter	Atlanta Hawks	SG	76	90	97.78	100	91.11	
Kevin Huerter	Atlanta Hawks	SF	76	90	97.78	100	91.11	
Delon Wright	Atlanta Hawks	PG	76	83.33	79.17	67.71	83.33	
Delon Wright	Atlanta Hawks	SG	76	83.33	79.17	67.71	83.33	
Onyeka Okongw	Atlanta Hawks	C	74	70.45	95.45	68.18	68.18	
Onyeka Okongw	Atlanta Hawks	PF	74	70.45	95.45	68.18	68.18	
Gorgui Dieng	Atlanta Hawks	C	74	86.81	93.41	83.52	85.71	
Jalen Johnson	Atlanta Hawks	SF	73	80	87.06	87.06	81.18	
Jalen Johnson	Atlanta Hawks	PF	73	80	87.06	87.06	81.18	
Timothe Luwawu	Atlanta Hawks	SF	72	75.29	62.35	76.47	87.06	
Timothe Luwawu	Atlanta Hawks	SG	72	75.29	62.35	76.47	87.06	
Sharife Cooper	Atlanta Hawks	PG	71	78.41	87.5	82.95	77.27	
Skylar Mays	Atlanta Hawks	SF	71	84.78	91.3	91.3	81.52	
Skylar Mays	Atlanta Hawks	SG	71	84.78	91.3	91.3	81.52	

Figura 14. Dataset resultante de la ejecución del script que obtiene los datos de los jugadores de 2kratings mediante web scraping.

4.3. Creación del modelo de machine learning para el clustering de jugadores y construcción de la base de datos

Una vez se cuenta con el conjunto de datos sobre jugadores ya transformado, el siguiente paso en el desarrollo es el de aplicar un algoritmo de clustering (aprendizaje no supervisado) sobre esos datos y construir una BD donde almacenarlos. Las tareas descritas en este apartado se han implementado mediante scripts escritos en el lenguaje Python, disponibles en el repositorio del proyecto (Conde, 2021).

Clustering de jugadores

La primera tarea que realizar es la de separar el conjunto de datos en cinco subconjuntos, uno por cada posición que existe en baloncesto (PG, SG, SF, PF y C). Esto se hace porque es obvio que la posición de un jugador está ligada a las características de este, y aplicar un algoritmo de clustering sobre el conjunto de datos completo podría dar como resultado clústeres identificativos de cada posición, y no es lo que se pretende. Lo que se quiere conseguir es formar distintos grupos de jugadores con ciertos patrones en sus atributos. Para este propósito se ha considerado que lo oportuno es dividir primero el conjunto de datos, para posteriormente aplicar el algoritmo de clustering sobre cada uno de los cinco subconjuntos. Se ha decidido, para cada subconjunto, clasificar a los jugadores que pertenecen a él en 4 clústeres y, por lo que hay que tener en cuenta que los pasos que se describen a continuación se han realizado para cada uno de los 5 subconjuntos de datos. Para cualquier ejemplo que se muestre en este apartado, tomaremos como referencia el subconjunto de jugadores con posición PG.

En este punto del desarrollo se cuenta con los datos ya transformados en pasos anteriores, que cuentan con los siguientes atributos para cada jugador:

- Name
- Overall
- Mid-Range Shot
- Shot IQ
- Speed
- Vertical
- Overall Durability
- Standing Dunk
- Post Fade
- Hands
- Ball Handle
- Team
- Outside Scoring
- Three-Point Shot
- Offensive Consistency
- Acceleration
- Stamina
- Inside Scoring
- Driving Dunk
- Post Control
- Playmaking
- Speed with Ball
- Position
- Close Shot
- Free Throw
- Athleticism
- Strength
- Hustle
- Layup
- Post Hook
- Draw Foul
- Pass Accuracy
- Pass IQ

- Pass Vision
- Perimeter Defense
- Lateral Quickness
- Defensive Consistency
- Defensive Rebound
- Total Attributes
- Defending
- Steal
- Help Defense IQ
- Rebounding
- Intangibles
- Interior Defense
- Block
- Pass Perception
- Offensive Rebound
- Potential

Entre todos estos atributos hay 6 que se corresponden cada uno a una habilidad baloncestística global que ha sido calculada como la media de varios atributos más específicos. La correspondencia es la que se muestra a continuación en la Tabla 4:

Tabla 4. Correspondencia entre atributos que representan habilidades globales y los atributos específicos a partir de los cuales se han calculado

Atributo calculado (habilidad global)	Atributos específicos (destrezas)
Outside Scoring	o Close Shot o Mid-Range Shot o Three-Point Shot o Free Throw o Shot IQ o Offensive Consistency
Athleticism	o Speed o Acceleration o Strength o Vertical o Stamina o Hustle o Overall Durability
Inside Scoring	o Layup o Standing Dunk o Driving Dunk o Post Hook o Post Fade o Post Control o Draw Foul o Hands
Playmaking	o Pass Accuracy o Ball Handle o Speed with Ball o Pass IQ o Pass Vision
Defending	o Interior Defense o Perimeter Defense

	○ Steal ○ Block ○ Lateral Quickness ○ Help Defense IQ ○ Pass Perception ○ Defensive Consistency
Rebounding	○ Offensive Rebound ○ Defensive Rebound

Dada la correlación entre los atributos específicos y los calculados, para esta primera versión de la solución se ha decido utilizar esas 6 habilidades globales como entrada del algoritmo de clustering y clasificar entonces a los jugadores según esos 6 atributos. De este modo, los atributos que tendrán en cuenta el algoritmo serán los siguientes:

- Outside Scoring
- Athleticism
- Inside Scoring
- Playmaking
- Defending
- Rebounding

El algoritmo de clustering que se ha aplicado sobre los datos ha sido el algoritmo K-Means. De entre los diferentes algoritmos de clustering que ofrece la librería scikit-learn (scikit-learn developers (BSD License), 2011a), se ha escogido K-Means por ser un algoritmo de propósito general que da buenos resultados en casos con un gran número de muestras y un número de clústeres no muy grande, y por ser la distancia entre puntos la métrica que utiliza para generar los clústeres y clasificar las instancias.

El parámetro más importante que indicar al algoritmo K-Means en sklearn es *n_clusters*, que nos permite indicar el número de clústeres a formar y que será también el número de centroides que se generarán. Se ha decidido usar un n_clusters=4 ya que ha sido el número de clústeres que mejores resultados proporcionaba en líneas generales al probar el algoritmo KMeans con el software Orange3 (Bioinformatics Laboratory, University of Ljubljana, 2016) para cada una de las posiciones con diferentes n_clusters. La decisión de utilizar el mismo n_clusters para clasificar a los jugadores de cada una de las posiciones tiene como motivación el simplificar el desarrollo, ya que el código que describe la lógica que calcula y almacena los campos que indican los grados de pertenencia y compatibilidad con los clústeres es el mismo para las 5 posiciones. Es obvio que tomar esta decisión implica

que la clasificación de los jugadores perderá precisión, pero resulta necesario para poder cumplir con el alcance del proyecto.

La clase *sklearn.cluster.KMeans* dispone del método *fit_transform*. Este método permite ejecutar el clustering sobre el conjunto de datos y retorna, para cada instancia, un array con las distancias a cada uno de los clústeres que se han generado. Esta información es muy útil, ya que no interesa saber únicamente a qué clúster pertenece una instancia. Para el propósito de este trabajo lo que se quiere conocer es el grado de pertenencia de una instancia a cada clúster, es decir, en qué grado un jugador es de un tipo determinado.

En este punto se añaden cuatro campos más a cada jugador que indican su grado de pertenencia a cada clúster. Esos campos son *Pertenencia Cluster 0*, *Pertenencia Cluster 1*, *Pertenencia Cluster 2* y *Pertenencia Cluster 3*. Para darles un valor, se busca para cada clúster *c* la mayor y menor distancia que exista de un jugador a ese clúster, y calculamos la pertenencia de cada jugador j a c del siguiente modo:

$$Pertenencia_{j,c} = 1 - ((distanceToCluster_{j,c} - minDistanceToCluster_c)/ (maxDistanceToCluster_c - minDistanceToCluster_c))$$

De este modo se le asignará, por ejemplo, un 1 en el campo *Pertenencia Cluster 0* al jugador más cercano al centroide del clúster 0, y un 0 al más lejano.

En el Capítulo 5 de esta memoria se realizará la evaluación del software construido, donde se realizarán comentarios sobre los resultados obtenidos del clustering realizado en esta fase del proyecto.

Construcción de la base de datos

En este punto del desarrollo, con los datos de los jugadores transformados y con campos indicadores de su grado de pertenencia a los diferentes clústeres en los que se pueden clasificar, es oportuno persistir toda esta información.

Como tecnología para garantizar la persistencia de los datos se ha seleccionado MongoDB. En concreto, se ha utilizado el servicio MongoDB Atlas, que ofrece el uso gratuito de un clúster MongoDB con prestaciones básicas: un réplica-set de 3 nodos con un almacenamiento de 512mb (Figura 15). Para la cantidad de datos que se maneja en este desarrollo será suficiente, pero no lo sería si se escalase la solución.

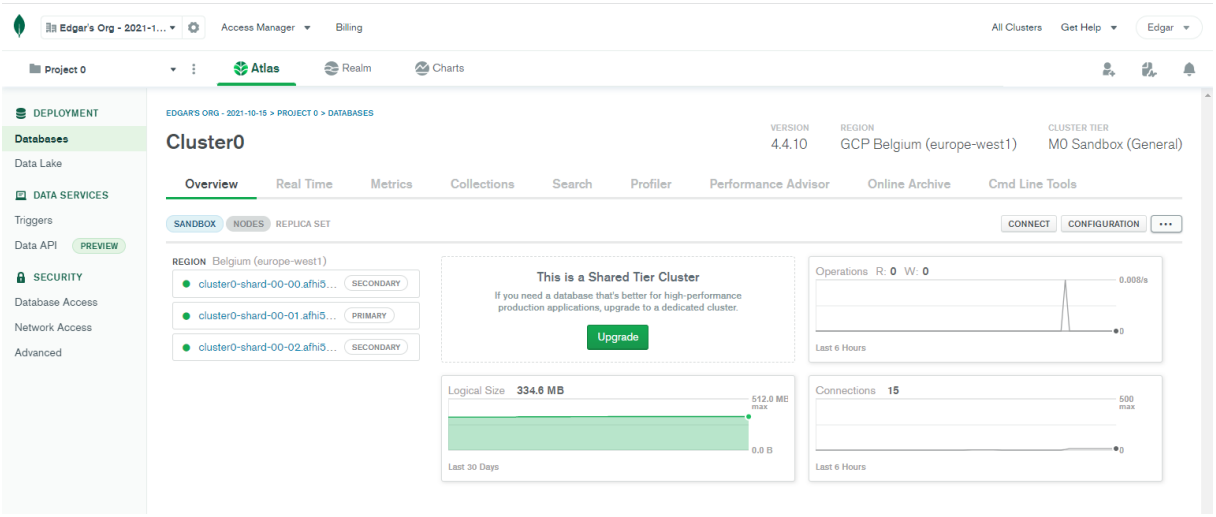

Figura 15.Panel de gestión de un clúster en MongoDB Atlas.

Para garantizar que no haya accesos no deseados a la base de datos, se restringe el acceso a ésta requiriendo siempre una contraseña para conectarse. Como se muestra en la Figura 16, se pueden gestionar los usuarios de la base de datos desde el panel de seguridad de MongoDB Atlas.

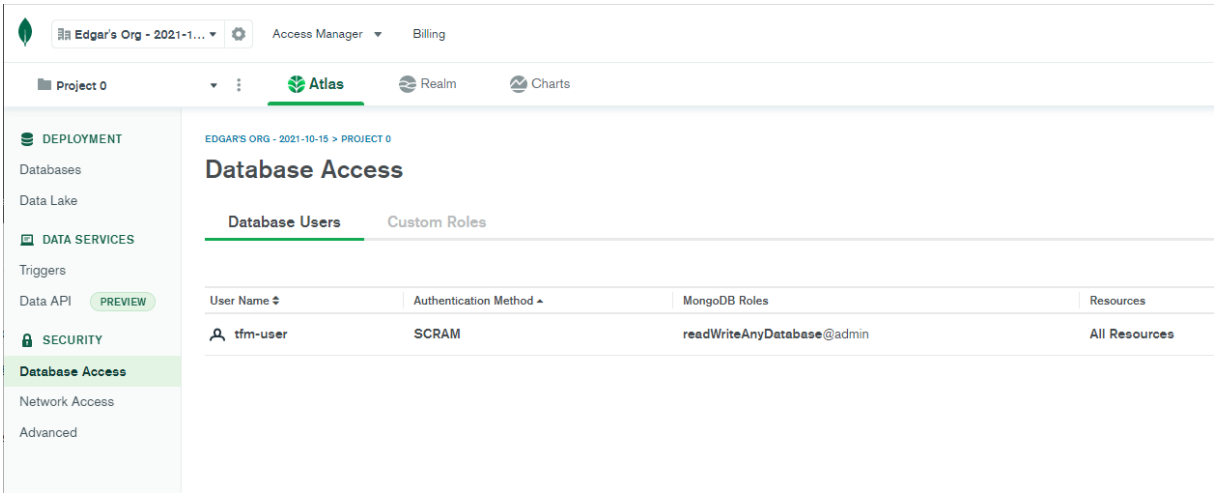

Figura 16. Panel de gestión de usuarios de MongoDB Atlas.

Haciendo uso del driver de MongoDB para Python, todos estos datos, cargados desde ficheros CSV y transformados utilizando *pandas* y *numpy*, son volcados en una colección llamada *players* dentro de la base de datos *tfm*. La Figura 17 muestra un ejemplo de documento correspondiente a un jugador.

```
{
    "_id": {
        "$oid": "619dcb8a4af3a4dc167dc49b"
    },
    "Name": "Delon Wright",
    "Team": "Atlanta Hawks",
    "Position": "PG",
    "Overall": 76,
    "Outside Scoring": 83.33333333333333,
    "Close Shot": 79.16666666666667,
    "Mid-Range Shot": 67.70833333333333,
    "Three-Point Shot": 83.33333333333333,
    "Free Throw": 83.33333333333333,
    "Shot IQ": 100,
    "Offensive Consistency": 83.33333333333333,
    "Athleticism": 80.20833333333333,
    "Speed": 76.04166666666667,
    "Acceleration": 76.04166666666667,
    "Strength": 60.41666666666666,
    "Vertical": 71.875,
    "Stamina": 93.75,
    "Hustle": 93.75,
    "Overall Durability": 87.5,
    "Inside Scoring": 54.16666666666666,
    "Layup": 85.41666666666667,
    "Standing Dunk": 26.041666666666668,
    "Driving Dunk": 62.5,
    "Post Hook": 26.041666666666668,
    "Post Fade": 47.91666666666666,
    "Post Control": 28.125,
    "Draw Foul": 67.70833333333333,
    "Hands": 88.54166666666667,
    "Playmaking": 83.33333333333333,
    "Pass Accuracy": 89.58333333333333,
    "Ball Handle": 89.58333333333333,
    "Speed with Ball": 75,
    "Pass IQ": 88.54166666666667,
    "Pass Vision": 73.95833333333333,
    "Defending": 67.70833333333333,
    "Interior Defense": 34.375,
    "Perimeter Defense": 78.125,
    "Steal": 89.58333333333333,
    "Block": 57.29166666666666,
    "Lateral Quickness": 78.125,
    "Help Defense IQ": 65.625,
    "Pass Perception": 83.33333333333333,
    "Defensive Consistency": 57.29166666666666,
    "Rebounding": 54.16666666666666,
    "Offensive Rebound": 46.875,
    "Defensive Rebound": 60.41666666666666,
    "Intangibles": 67.70833333333333,
    "Potential": 79.16666666666667,
    "Total Attributes": "2,595",
    "Pertenencia Cluster 0": 0.7525306422560829,
    "Pertenencia Cluster 1": 0.7321262476661148,
    "Pertenencia Cluster 2": 0.8013198624528266,
    "Pertenencia Cluster 3": 0.8419995891492709
}
```


Figura 17.Ejemplo de un documento de la colección de jugadores en MongoDB.

4.4. Extracción de información del rendimiento por parejas de jugadores en partidos

En este punto del desarrollo ya se dispone de una base de datos MongoDB donde está almacenada toda la información relativa a los jugadores. La siguiente tarea por realizar es la de extraer de datos de partidos la información que nos indique el rendimiento de los jugadores por pares. A continuación, se detalla el procedimiento seguido para acometer ese objetivo, implementado mediante un script escrito en lenguaje Python, disponible en el repositorio del proyecto (Conde, 2021).

Partimos de un dataset que contiene información de partidos desde 2015 a 2021. Se trata de ficheros CSV (uno por temporada) que contienen los datos jugada a jugada de todos los partidos, con campos que permiten obtener toda la información necesaria de cada jugada. Cada registro, correspondiente a una jugada, cuenta con todos los siguientes campos:

- URL
- Date
- Quarter
- AwayPlay
- HomePlay
- ShotType
- Assister
- Fouler
- ReboundType
- TimeoutTeam
- FreeThrowNum
- TurnoverPlayer
- TurnoverCauser
- JumpballPoss
- GameType
- Time
- SecLeft
- AwayScore
- HomeScore
- ShotOutCome
- Blocker
- Fouled
- ViolationPlayer
- FreeThrowShooter
- EnterGame
- TurnoverType
- JumpballAwayPlayer
- Location
- WinningTeam
- AwayTeam
- HomeTeam
- Shooter
- ShotDist
- FoulType
- Rebounder
- ViolationType
- FreeThrowOutcome
- LeaveGame
- TurnoverCause
- JumpbalHomePlayer

Para extraer la información necesaria para el sistema que se está desarrollando, únicamente se trabajará con los campos:

- URL: Indica una dirección web dónde se pueden consultar los quintetos iniciales de los equipos.
- Quarter: Indica en qué cuarto del partido transcurre la jugada actual.
- SecLeft: Segundos restantes en el cuarto actual.
- AwayScore: Puntos del equipo visitante.
- HomeScore: Puntos del equipo local.

- EnterGame: Si hay una sustitución, jugador que entra en la pista.
- LeaveGame: Si hay una sustitución, jugador que sale de la pista.

Se ha desarrollado un script que vuelca el CSV en un Dataframe y ejecuta un algoritmo que lee una a una las filas de éste, y teniendo en cuenta las sustituciones de jugadores y el marcador del partido, obtiene como resultado, para cada partido, todas las combinaciones de pares de jugadores de un mismo equipo que han compartido tiempo en pista. Cada par de jugadores se almacena en la colección *pairs_plus_minus* de la base de datos *tfm*, acompañado de un campo que indica el tiempo compartido y el resultado parcial del equipo durante ese tiempo. La Figura 18 es un ejemplo de documento de esta colección.

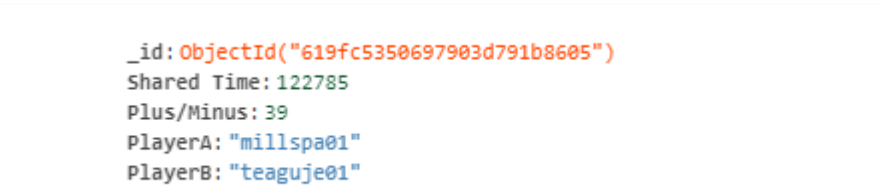

Figura 18.Ejemplo de un documento de la colección pairs_plus_minus.

Los campos *Shared Time* y *Plus/Minus* de esta colección se van actualizando conforme se leen más partidos, ya que un par de jugadores puede llegar a disputar juntos un centenar de partidos en una temporada, por lo que *Shared Time* se incrementará cada partido con el tiempo en segundos compartido en ese partido, mientras que *Plus/Minus* se incrementará o decrementará según el más/menos obtenido en ese partido.

Cálculo de compatibilidad de jugadores con clústeres

Ya en disposición de la información acerca del rendimiento por parejas de jugadores, el siguiente paso es el de calcular la compatibilidad de los jugadores con cada clúster.

En este paso, de nuevo se ha hecho uso del lenguaje Python, en este caso para realizar una conexión a la base de datos para consultar la colección de jugadores y para calcular, para cada uno, una serie de nuevos campos que indican el grado de compatibilidad con cada uno de los tipos de jugadores que se han definido al aplicar el algoritmo de clustering en etapas previas del desarrollo. Estos nuevos campos son:

- Compatibilidad PG Cluster 0
- Compatibilidad PG Cluster 1
- Compatibilidad PG Cluster 2
- Compatibilidad PG Cluster 3
- Compatibilidad SG Cluster 0
- Compatibilidad SG Cluster 1
- Compatibilidad SG Cluster 2
- Compatibilidad SG Cluster 3
- Compatibilidad SF Cluster 0
- Compatibilidad SF Cluster 1

- Compatibilidad SF Cluster 2
- Compatibilidad PF Cluster 0
- Compatibilidad PF Cluster 2
- Compatibilidad C Cluster 0
- Compatibilidad C Cluster 2
- Compatibilidad SF Cluster 3
- Compatibilidad PF Cluster 1
- Compatibilidad PF Cluster 3
- Compatibilidad C Cluster 1
- Compatibilidad C Cluster 3

Cada uno de estos campos es numérico y toma un valor entre 0 y 1 que representa el grado de compatibilidad de un jugador con jugadores de un clúster de una posición determinada. Por ejemplo, para calcular la compatibilidad de un jugador con los clústeres de la posición SG, los pasos que se realizan son los siguientes:

- Para cada clúster X:
  - Buscar todos los jugadores con posición SG que hayan compartido tiempo de juego con el jugador.
  - Sumar el campo *Shared Time* de todos esos jugadores (SharedTimeTotal).
  - Sumar el campo *Plus/Minus* de todos esos jugadores (Plus/MinusTotal), multiplicando el valor correspondiente a cada jugador por el valor de su campo *Pertenencia Cluster X*.
  - Calcular el campo *Compatibilidad PG Cluster X* como SharedTimeTotal dividido entre Plus/MinusTotal.
- Obtenidos los campos *Compatibilidad PG Cluster 0*, *Compatibilidad PG Cluster 1*, *Compatibilidad PG Cluster 2 y Compatibilidad PG Cluster 3*, se normalizan al rango 0-1:

$$Compatibilidad_{j,c} = (Compatibilidad_{j,c} - \min(compatibilidades))/$$
$$(\max(compatibilidades) - \min(compatibilidades))$$

- Se almacenan los nuevos campos en el documento del jugador en la base de datos.

4.5. Desarrollo y despliegue de la aplicación web

Para dotar a la solución de una interfaz a través de la cuál ejecutarla y probarla, se ha construido una pequeña SPA (aplicación web de una sola página) que consume una API REST sencilla. En este apartado se describe el desarrollo de la API y de la aplicación web, así como su despliegue.

Desarrollo de la API

Para dotar a la interfaz de la solución de los datos sobre jugadores y recomendaciones, se ha desarrollado una API en NodeJs (OpenJS Foundation, 2020), haciendo uso de librerías como Mongoose (*Mongoose ODM v6.1.3*, 2012) o Express (OpenJS Foundation, 2015).

La API se encarga de conectarse a la base de datos MongoDB que se ha construido en pasos anteriores del desarrollo, y servir los documentos de Mongo en formato JSON para una fácil interpretación de la respuesta por parte de la aplicación web que la va a consumir, así como de cualquier otro potencial cliente de la API. Los endpoints expuesto en el API son los indicados en la Tabla 5 para consultar la información de los jugadores y los compañeros recomendados.

Tabla 5. Correspondencia entre atributos

URL endpoint	Descripción	Query params
GET /api/players	Devuelve el listado completo de jugadores	N/A
GET /api/players/{id}	Devuelve la información de un solo jugador	N/A
GET /api/players/{id}/recomendaciones	Devuelve el listado de jugadores de una posición indicando la compatibilidad con el jugador. Ésta se calcula teniendo en cuenta la compatibilidad del jugador con los diferentes clústeres y el grado de pertenencia a los diferentes clústeres de los jugadores a recomendar.	**position**: la posición para la cual buscar jugadores a recomendar

Para realizar pruebas sobre la API, se ha ido comprobando el correcto funcionamiento de ésta haciendo uso del software Postman, como se ilustra a continuación en la Figura 19.

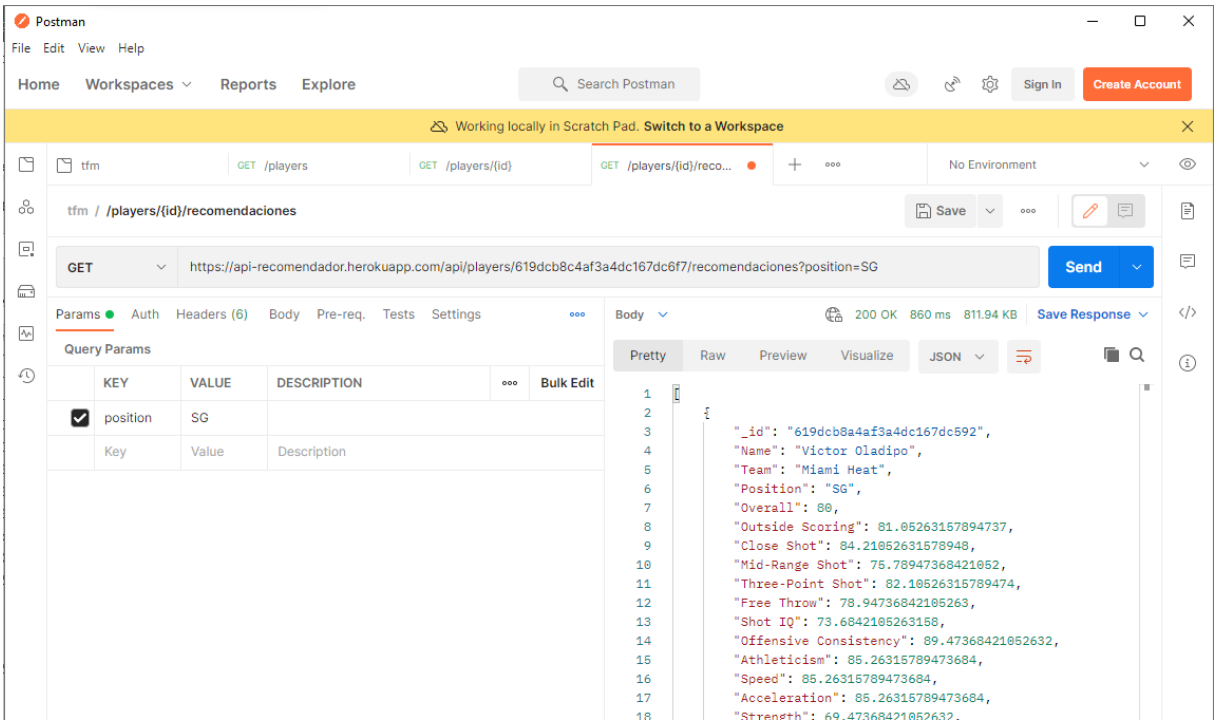

Figura 19.Ejecución de pruebas de la API a través de Postman.

Desarrollo de la interfaz

El siguiente paso en la construcción del software es dotarlo de una interfaz para que un usuario pueda interactuar con él y probarlo. Para tal propósito se ha decidido desarrollar una pequeña aplicación web que permita visualizar el listado de jugadores y proporcione los botones o elementos interactivos necesarios para consultar jugadores recomendados de forma rápida e intuitiva.

Para el desarrollo se ha utilizado el framework Javascript Angular (Google, 2015), que permite una rápida puesta en marcha de aplicaciones web de una sola página pensadas para interactuar con APIs a través del protocolo HTTP.

La interfaz presenta una pantalla con el listado completo de jugadores. Como se puede ver en la Figura 20, el listado se puede filtrar por nombre de jugador o equipo para permitir una búsqueda más ágil. Además, también se pueden ordenar los resultados de la tabla según el valor de cualquiera de las siguientes columnas: *Nombre, Equipo, Valoración, Anotación Exterior, Anotación Interior, Defensa, Capacidad Atlética, Playmaking* y *Rebote*.

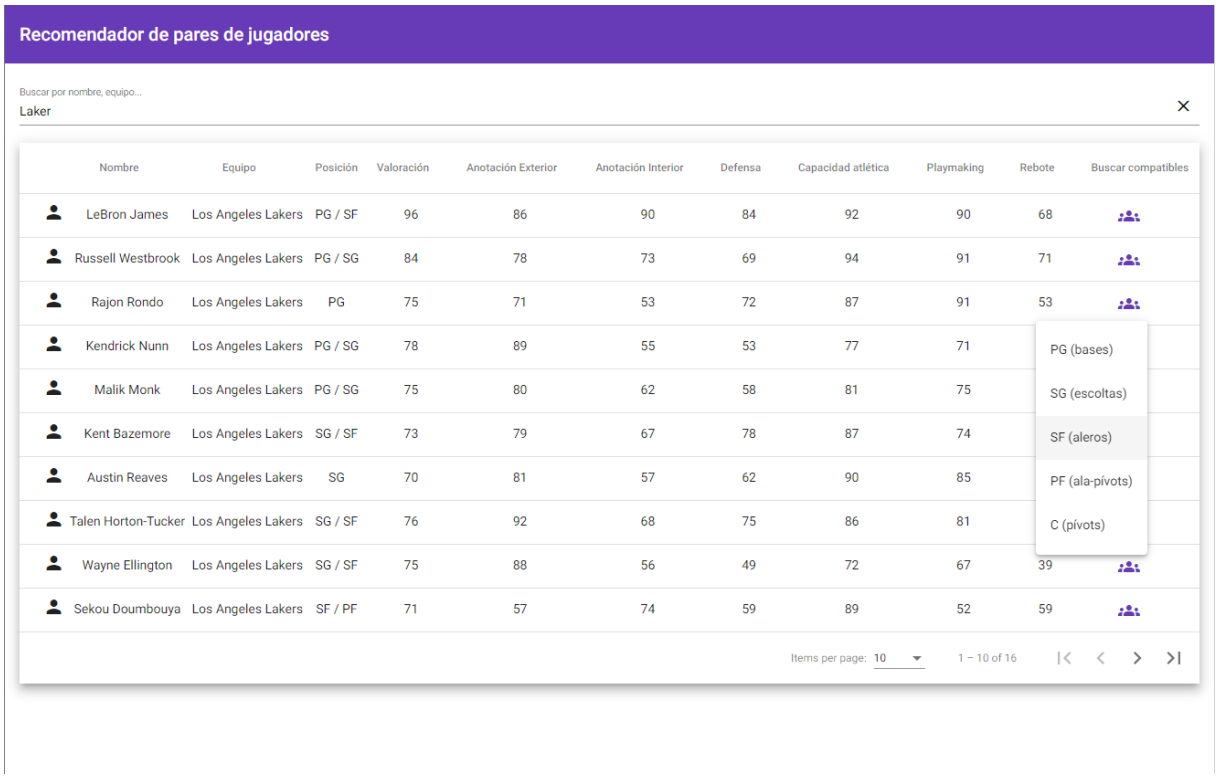

Figura 20.Interfaz web de la aplicación. Listado de jugadores.

Para consultar el listado de compañeros compatibles con un determinado jugador, basta con hacer click en el icono de la columna *Buscar compatibles* de la fila correspondiente al jugador deseado. A continuación, se desplegaría un menú contextual donde el usuario puede seleccionar para qué posición hacer la consulta. Finalmente, como muestra la Figura 21, se mostraría un *popup* con el listado de jugadores de la posición indicada, ordenados de mayor a menor *Índice de recomendación*.

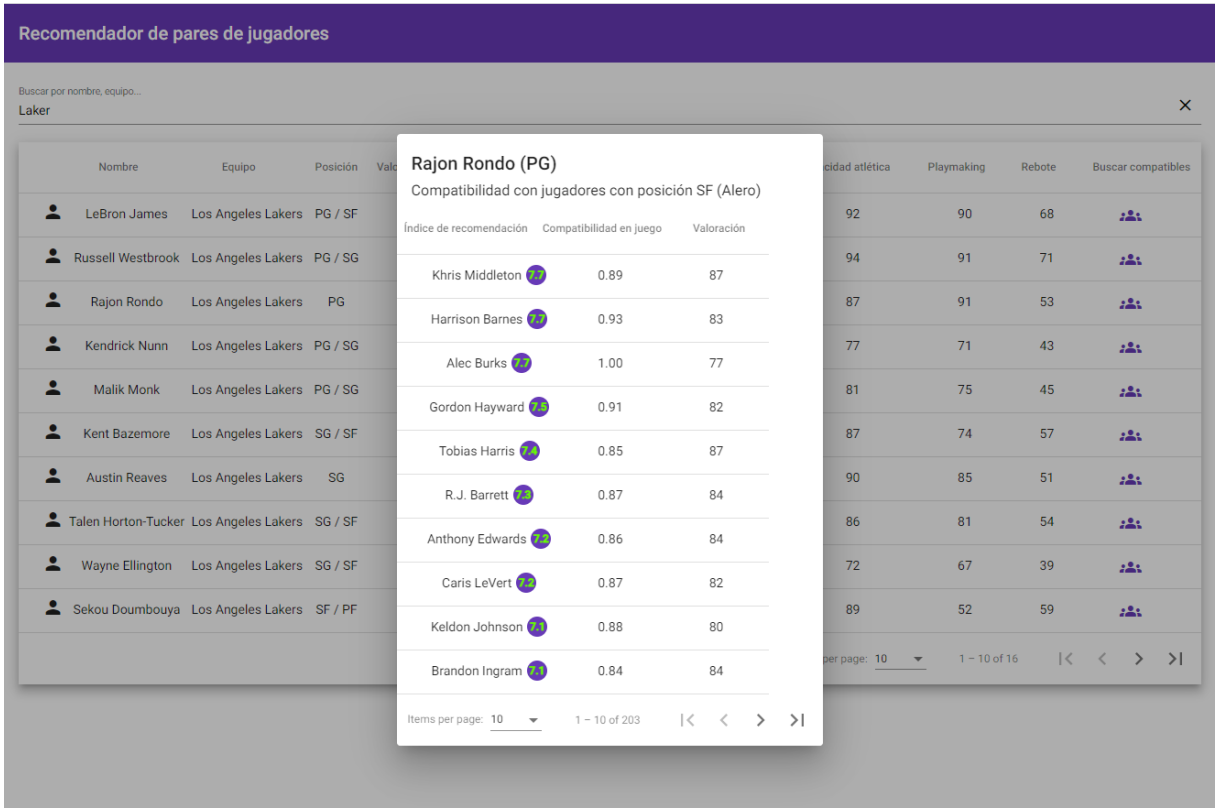

Figura 21.Interfaz web de la aplicación. Listado de jugadores recomendados.

Despliegue

Para hacer accesible la aplicación web a través de internet, se ha utilizado la plataforma Heroku para desplegar el software de manera gratuita.

En este caso, como muestra la Figura 22, se han creado 2 aplicaciones en Heroku ya que la API y la interfaz son piezas de software independientes que se despliegan por separado.

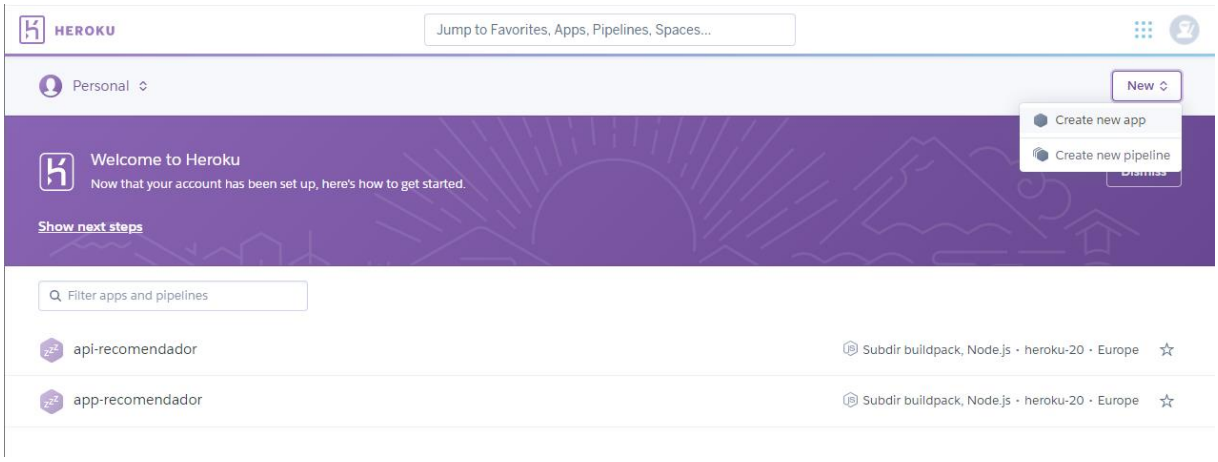

Figura 22.Web de Heroku. Listado de aplicaciones.

La Figura 23 muestra el dashboard de gestión de una aplicación en Heroku. Como se puede observar, se pueden configurar despliegues automáticos que se ejecutarán cuando se produzca un determinado evento (un commit en la rama master del repositorio, por ejemplo).

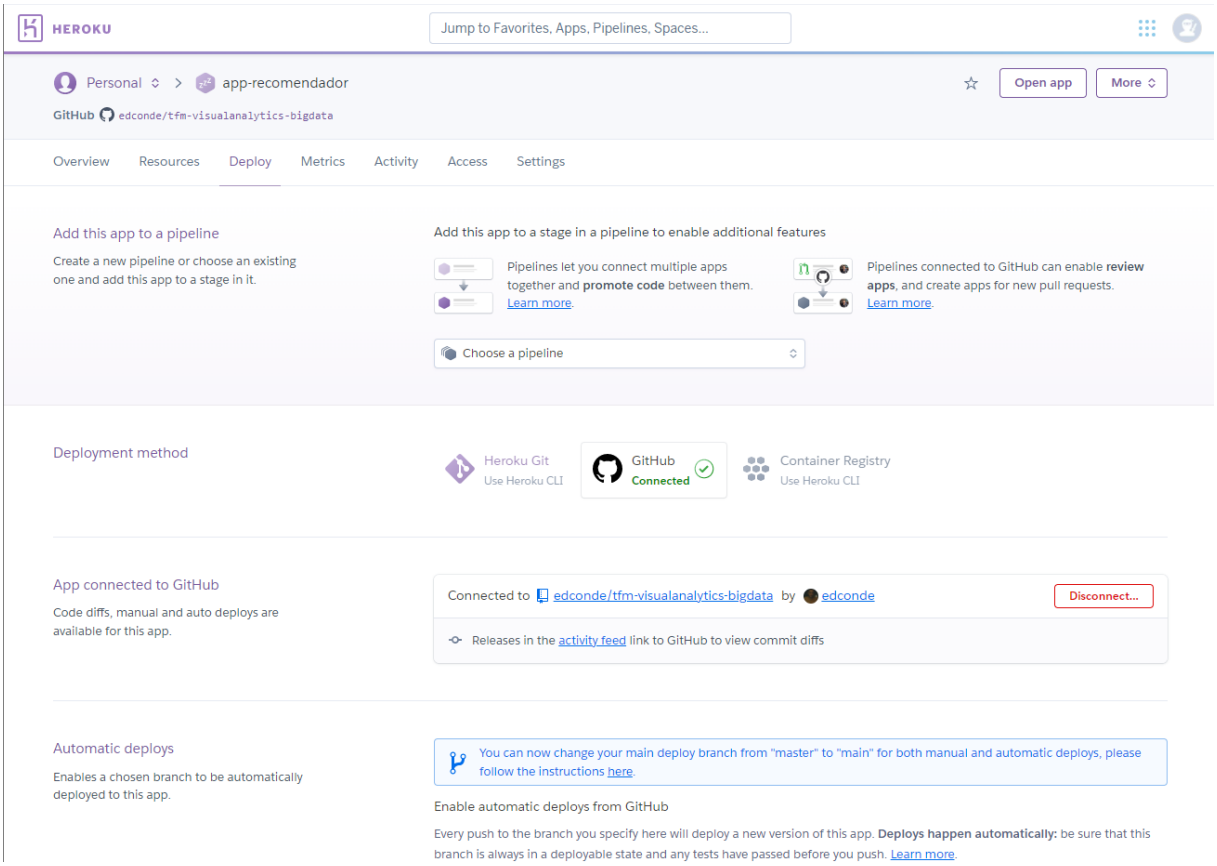

Figura 23.Web de Heroku. Dashboard de una aplicación.

Las aplicaciones desplegadas a través de Heroku lo hacen en la URL:

- *nombre_aplicación***.herokuapp.com**

Los dos artefactos software desplegados se encuentran en las siguientes direcciones web:

- API: https://api-recomendador.herokuapp.com
- Interfaz web: https://app-recomendador.herokuapp.com/

5. Evaluación

La evaluación de la herramienta desarrollada se ha llevado a cabo por dos vías. Una primera vía interna que consiste en la evaluación de la solución por parte del propio autor, y una segunda vía externa en la que dos usuarios afines al dominio del trabajo prestarán sus impresiones acerca del sistema.

5.1. Evaluación interna

La evaluación interna de la solución se ha realizado teniendo en cuenta los requisitos funcionales y no funcionales identificados en el Capítulo 4 para el desarrollo del software, así como los resultados obtenidos del clustering aplicado en el apartado 4.3.

A continuación, en la Tabla 6 y la Tabla 7 se recoge y justifica el cumplimiento o no de los requisitos funcionales y no funcionales.

Tabla 6. Evaluación del cumplimiento de requisitos funcionales

Requisitos funcionales		
Id	Cumplimiento SÍ / NO	Justificación / Comentarios
F1	SÍ	Además de poder filtrar la búsqueda, el usuario también puede ordenar los resultados de ésta.
F2	SÍ	La interfaz desarrollada permite seleccionar un jugador y la posición de los jugadores a recomendar, que son los dos parámetros necesarios.
F3	SÍ	Los parámetros Jugador y Posición de los jugadores a recomendar son seleccionables desde la interfaz y recibidos por la API para calcular la recomendación.
F4	SÍ	Los valores posibles para los parámetros de entrada son seleccionables desde la interfaz.
F5	SÍ	La recomendación es un listado de jugadores que juegan en la posición que se indica como parámetro de entrada, acompañados por su índice de compatibilidad con el jugador indicado como parámetro de entrada y su valoración en un rango del 0 al 100. Finalmente, no se ha incluido en la salida el índice de pertenencia al clúster con el que el jugador indicado como parámetro de entrada

		tiene mayor compatibilidad, ya que no se considera relevante para el usuario final que pueda usar la aplicación.
F6	SÍ	Se obtienen datos utilizando técnicas de web scraping, así como se hacen cálculos y transformaciones sobre los mismos.
F7	SÍ	Se almacenan todos los atributos obtenidos de la web de 2kratings.
F8	SÍ	Cada jugador cuenta con los campos *Pertenencia Cluster X* que indican que tipo de jugador es, y con los campos *Compatibilidad Cluster X,* que indican su compatibilidad con cada uno de los tipos de jugadores.

Tabla 7. Evaluación del cumplimiento de requisitos no funcionales

Requisitos no funcionales		
Id	Cumplimiento SÍ / NO	Comentarios
NF1	SÍ	La interfaz es ejecutable en los navegadores Chrome y Firefox.
NF2	SÍ	La interfaz de la solución es ser intuitiva y sencilla de usar, con dos clicks se puede obtener una recomendación para un jugador.
NF3	SÍ	Los resultados se presentan ordenados de mayor a menor valor de recomendación en una escala del 0 al 10, y con estilos CSS que facilitan al usuario su interpretación.
NF4	SÍ	Las recomendaciones se generan en un tiempo inferior a 5 segundos.
NF5	SÍ	El servidor que sirva la solución deberá permitir el acceso concurrente a la misma de al menos 10 usuarios.
NF6	SÍ	La base de datos donde se persisten los datos está protegida mediante usuario y contraseña.

Para finalizar la evaluación interna, se comentan los resultados obtenidos de la clasificación de los jugadores de cada una de las 5 posiciones. Las imágenes que se presentan son el resultado de aplicar la técnica MDS de escalado de múltiples dimensiones (Orange Data Mining, 2015) con el software Orange3 para poder plasmar en un gráfico bidimensional el resultado obtenido con el algoritmo KMeans, al cual se le habían proporcionado datos de 6 dimensiones como entrada.

La Figura 24 ilustra la clasificación de los jugadores con posición PG (base). Cabe mencionar casos como el de Kira Lewis Jr., clasificado en el clúster C1 (azul) a pesar de estar rodeado de jugadores del clúster C3 (naranja). Una explicación a este caso podría ser que la técnica MDS no haya funcionado del todo bien a la hora de plasmar los datos de forma bidimensional, aunque de ser así lo más probable es que hubiese más casos similares, pero en cambio, este caso parece bastante aislado. Se aprecia también que Luka Doncic y James Harden, que aparentemente están más próximos a jugadores del clúster C2 (rojo) están clasificados correctamente en el C3 (verde), ya que es en el que se ha clasificado también a LeBron James o Russell Westbrook, jugadores muy similares.

Figura 24. Gráfico bidimensional en Orange 3. Muestra los 4 clústeres de jugadores con posición PG.

La Figura 25 muestra cómo se clasifican los jugadores con posición SG. Este gráfico evidencia que quizá el número de clústeres en el que clasificar a este conjunto de jugadores debería ser distinto de 4, ya que la mayoría de los jugadores del clúster C1 (azul) y C4 (naranja) se concentran en la misma zona. En cambio, para los clústeres C2 (rojo) y C3

(verde) se aprecia una correcta clasificación, ya que el primero aglutina a jugadores conocidos por su habilidad de tiro como Seth Curry o Joe Harris y el segundo a jugadores famosos por su capacidad física y defensiva como Marcus Smart o Patrick Beverley.

Figura 25. Gráfico bidimensional en Orange 3. Muestra los 4 clústeres de jugadores con posición SG.

El gráfico presentado en la Figura 26 muestra cómo se han clasificado los jugadores con posición SF. Los jugadores asignados al clúster C2 (rojo) son todos jugadores que destacan por su capacidad física y defensiva, por lo que la clasificación en este clúster ha resultado ser muy correcta. Sin embargo, los clústeres C1 (azul) y C4 (naranja) parecen solaparse bastante. La causa podría ser el hecho de representar bidimensionalmente datos 6-dimensionales, aunque también habría que valorar la posibilidad de que estos dos clústeres pudiesen constituir un único clúster con jugadores de corte anotador como Paul George, Carmelo Anthony o Eric Gordon. También llama la atención el caso de Landry Shamet, que destaca por su habilidad para el tiro exterior, pero no es clasificado en el clúster C3 (verde) donde sí se encuentran correctamente clasificados otros tiradores como Duncan Robinson o Doug McDermott. El jugador Matisse Thybulle, a pesar de aparecer bastante apartado en esta representación bidimensional e invitar a pensar que podría tratarse de un outlier, está correctamente clasificado en el clúster C1 (azul) al compartir características similares con otros jugadores de ese clúster como Isaac Okoro o Mikal Bridges.

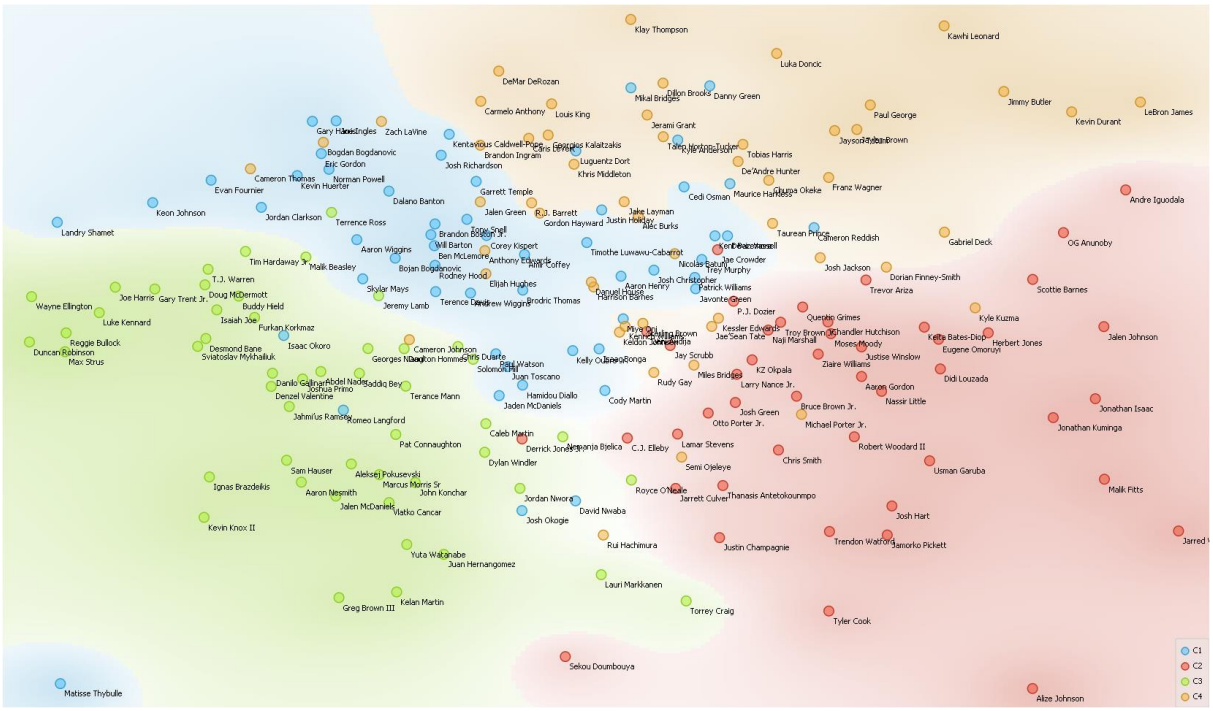

Figura 26. Gráfico bidimensional en Orange 3. Muestra los 4 clústeres de jugadores con posición SF.

La Figura 27 ilustra la clasificación de los jugadores con posición PF. Se presenta un caso parecido al anterior, donde los jugadores clasificados en el clúster C1 (azul) aparecen muy dispersos. Casos como el de David Bertans y Sadiqq Bey en el clúster C2 (rojo) o el de Khris Middleton y Obi Toppin en el C1 (azul) llaman la atención, al ser pares de jugadores con características muy dispares clasificados en el mismo clúster. Sin embargo, se aprecia una clasificación más que correcta en los clústeres C3 (verde) y C4 (naranja), que aglutinan jugadores con buenas aptitudes para la defensa y para el rebote, respectivamente.

Figura 27. Gráfico bidimensional en Orange 3. Muestra los 4 clústeres de jugadores con posición PF.

La Figura 28 presenta los clústeres en los que se clasifican los jugadores con posición C. Una clasificación bastante correcta en la que un usuario conocedor del dominio debería ser capaz de distinguir jugadores reboteadores en el clúster C1 (azul), pívots abiertos en el C2 (rojo), anotadores interiores en C3 (verde) y equilibrados en el C4 (naranja).

Figura 28. Gráfico bidimensional en Orange 3. Muestra los 4 clústeres de jugadores con posición C.

5.2. Evaluación externa

La evaluación externa de la solución ha sido realizada por parte de cuatro personas muy afines al dominio sobre el cual se ha desarrollado el proyecto. Además, todas ellas han practicado baloncesto (una a nivel semi-profesional) y dos de ellas son actualmente entrenadores en categorías inferiores de la Escuela Deportiva Ourense. Se les ha formulado una serie de preguntas a través de Google Forms para recoger sus impresiones acerca de la herramienta y los resultados que ofrece. En los siguientes apartados se exponen las respuestas a la encuesta.

Evaluación de la interfaz web

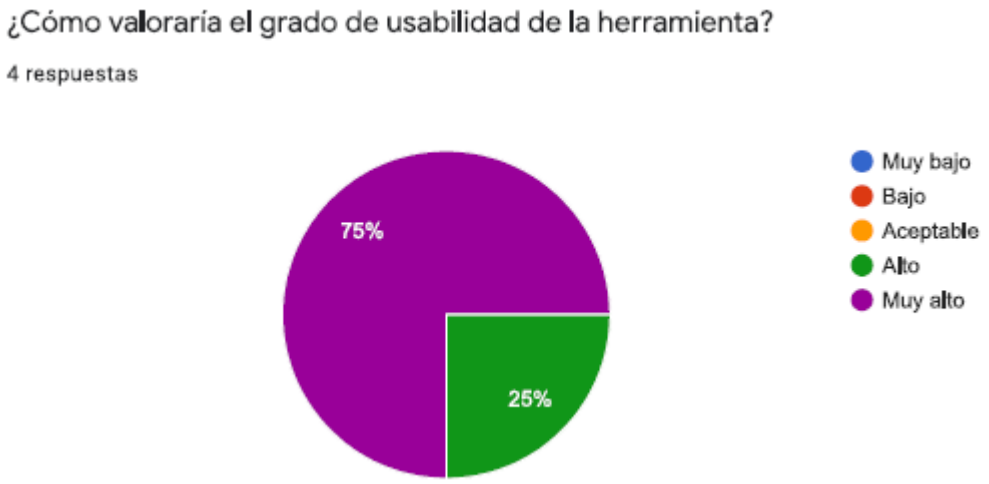

Figura 29. Evaluación de la interfaz web. Respuestas a la pregunta 1

Figura 30. Evaluación de la interfaz web. Respuestas a la pregunta 2

Figura 31. Evaluación de la interfaz web. Respuestas a la pregunta 3

Evaluación de la clasificación de jugadores

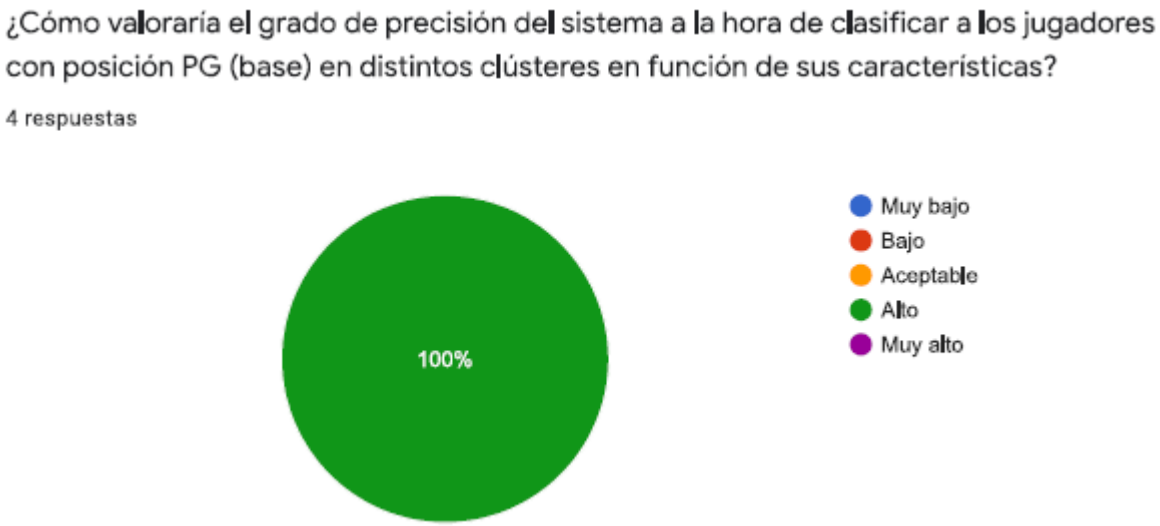

Figura 32. Evaluación de la clasificación de jugadores. Respuestas a la pregunta 1

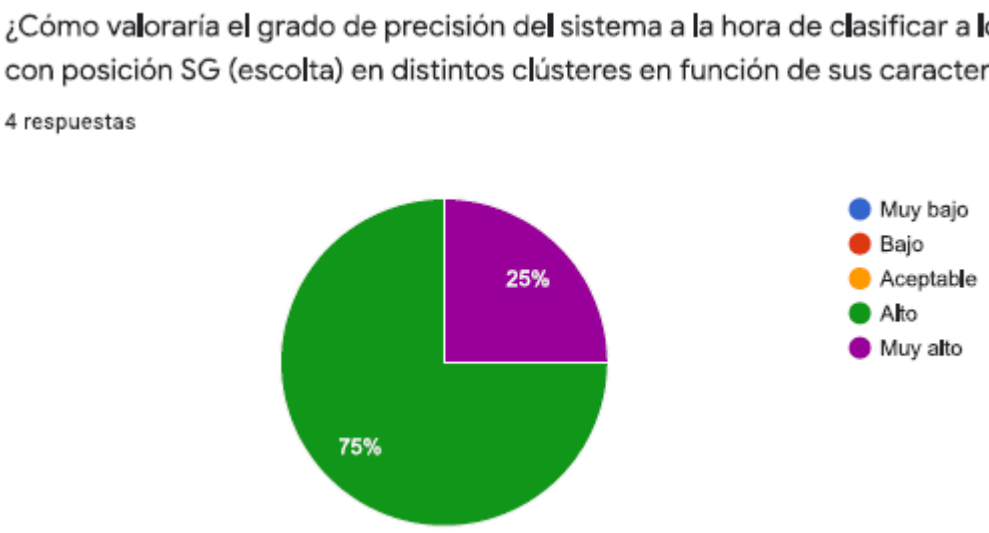

Figura 33. Evaluación de la clasificación de jugadores. Respuestas a la pregunta 2

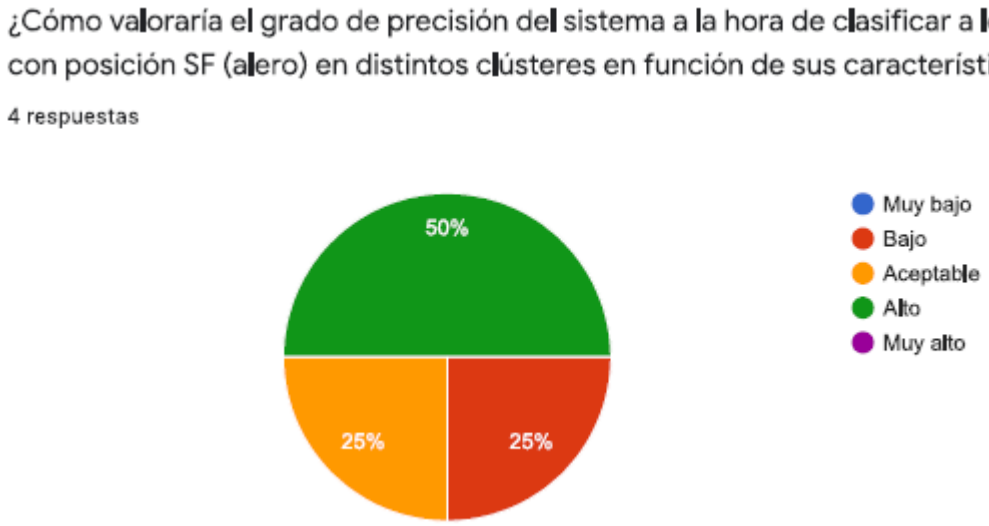

Figura 34. Evaluación de la clasificación de jugadores. Respuestas a la pregunta 3

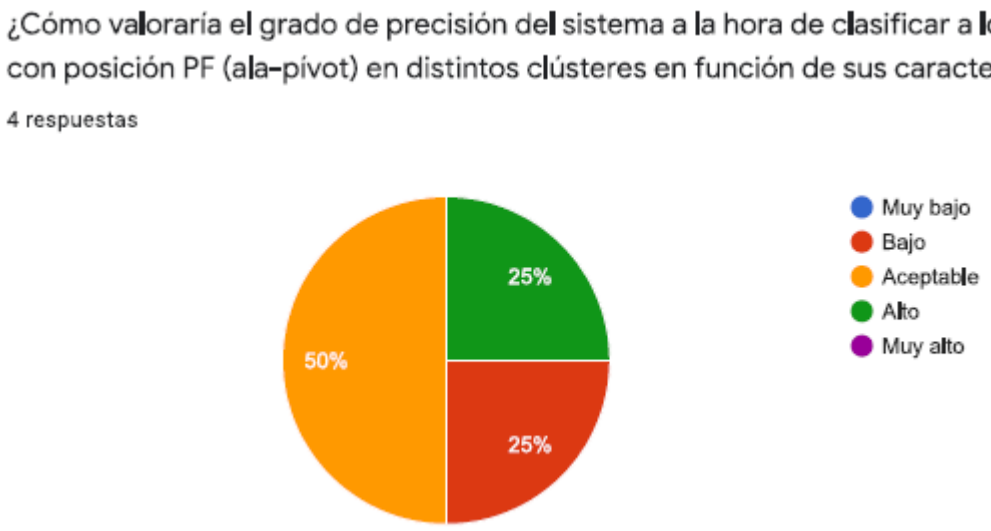

Figura 35. Evaluación de la clasificación de jugadores. Respuestas a la pregunta 4

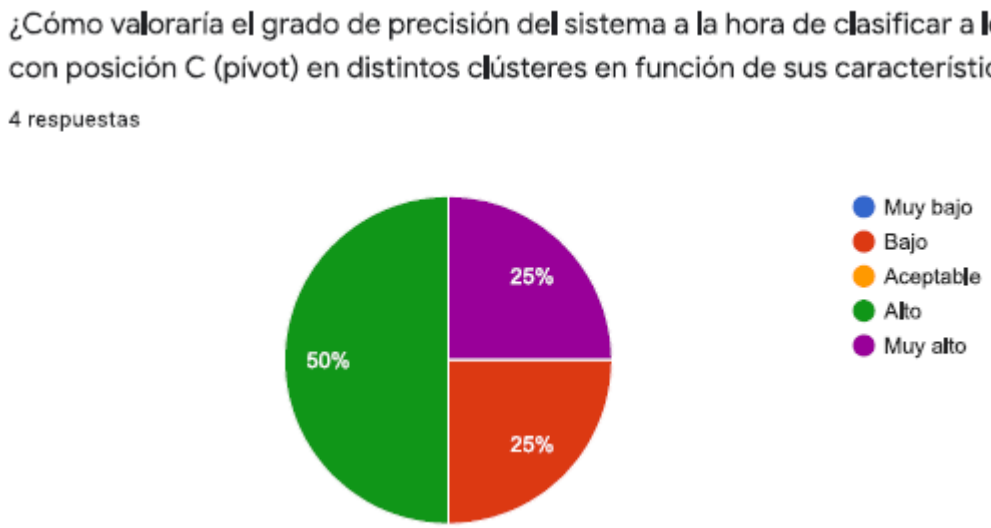

Figura 36. Evaluación de la clasificación de jugadores. Respuestas a la pregunta 5

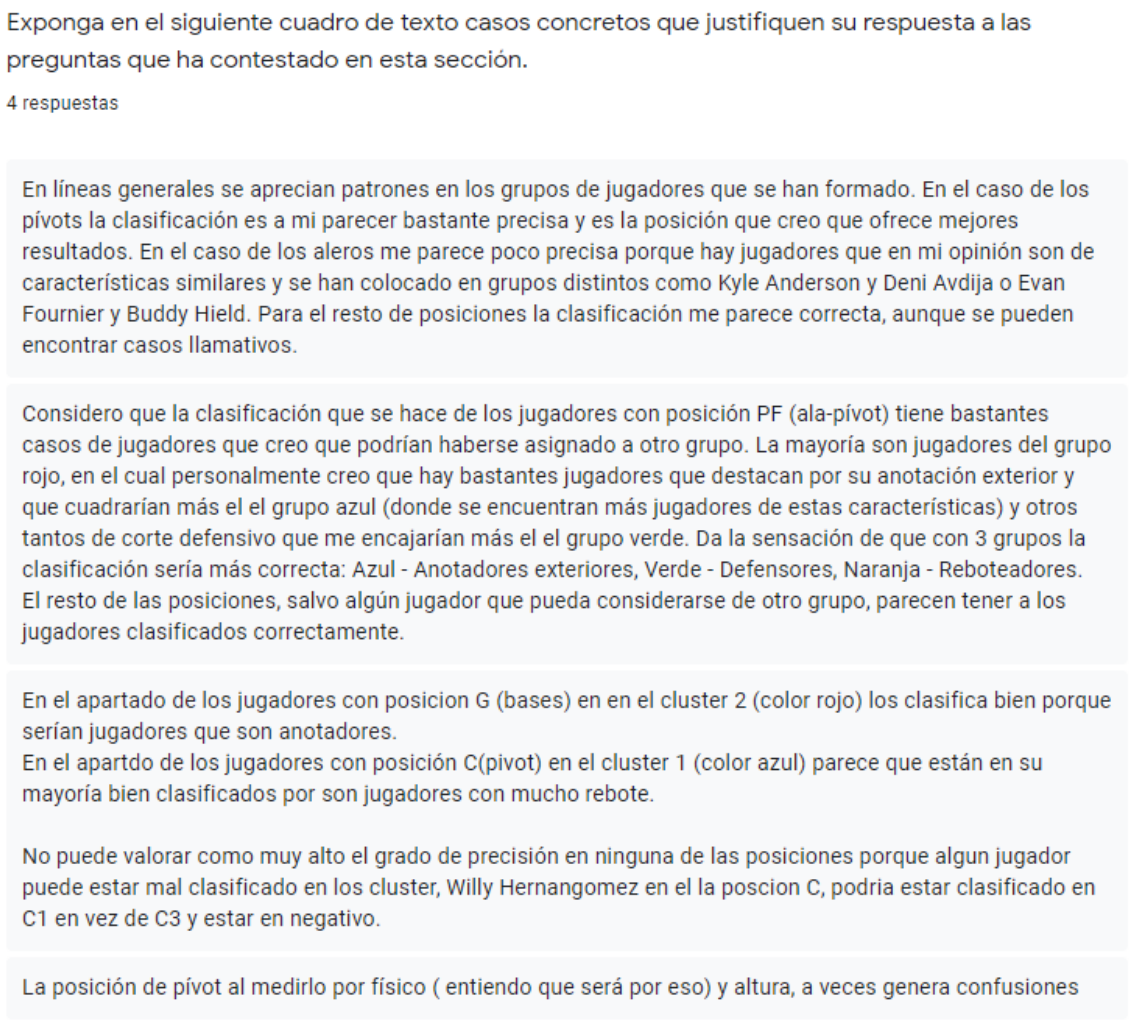

Figura 37. Evaluación de la clasificación de jugadores. Justificaciones a las respuestas

Evaluación de los resultados

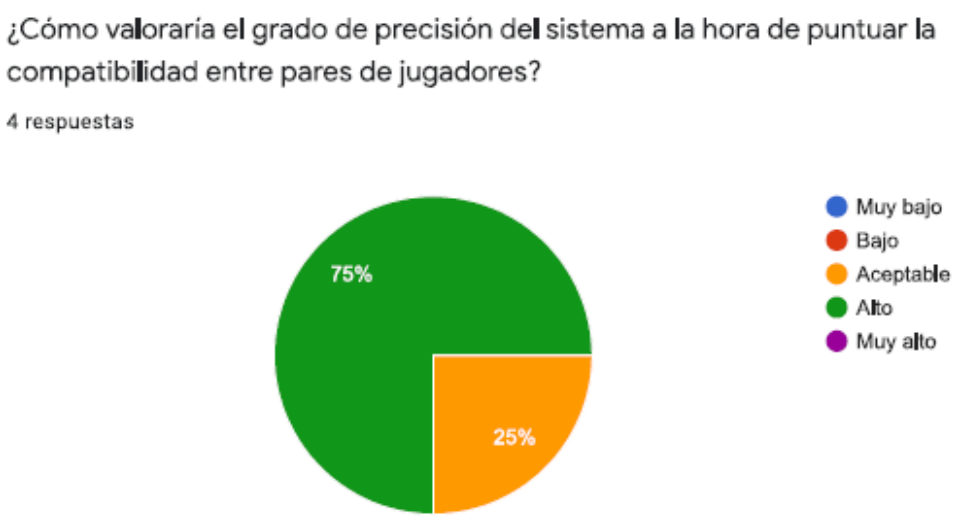

Figura 38. Evaluación de los resultados. Respuestas a la pregunta 1

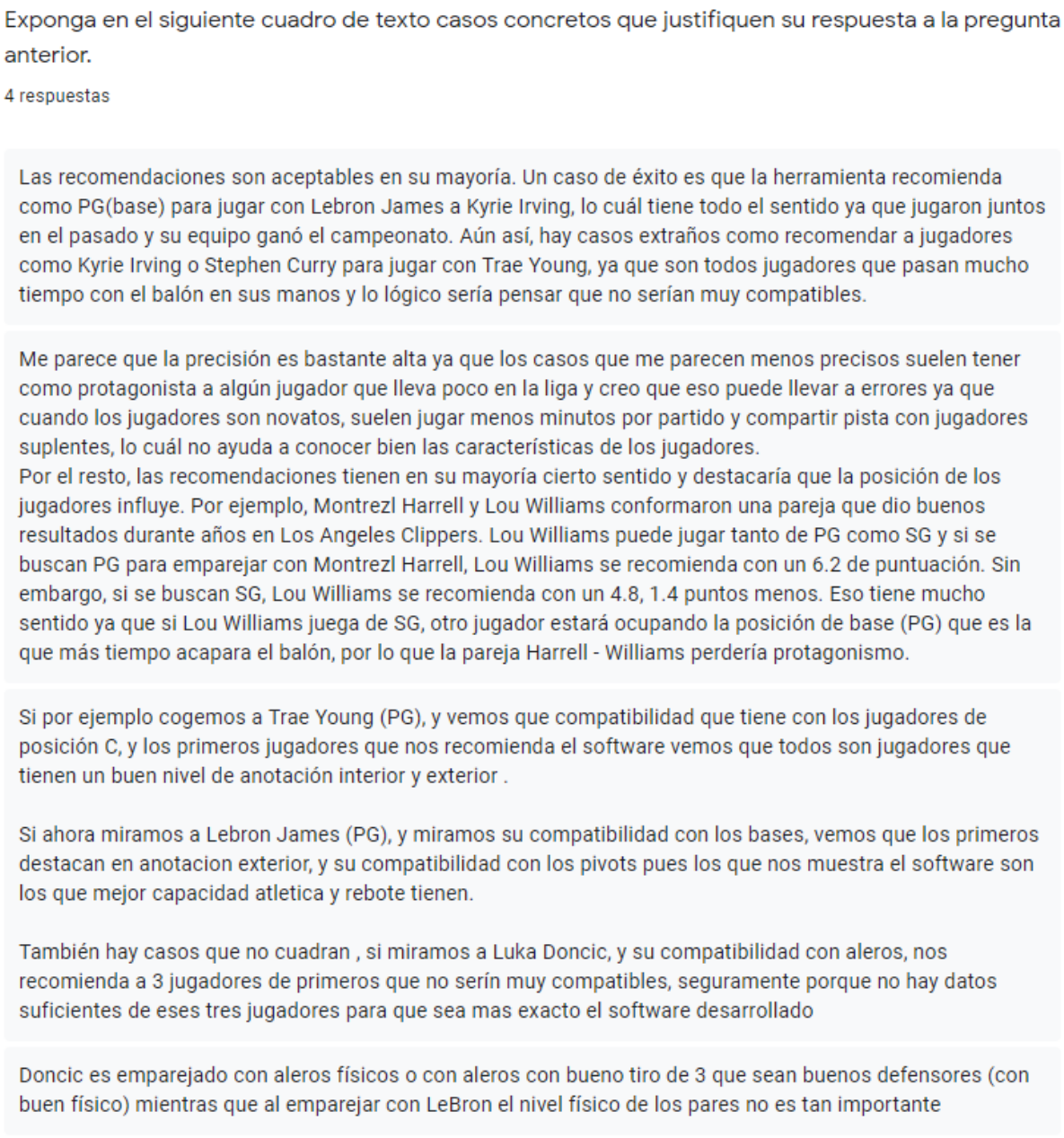

Figura 39. Evaluación de los resultados. Justificaciones a las respuestas

Evaluación de la aplicabilidad

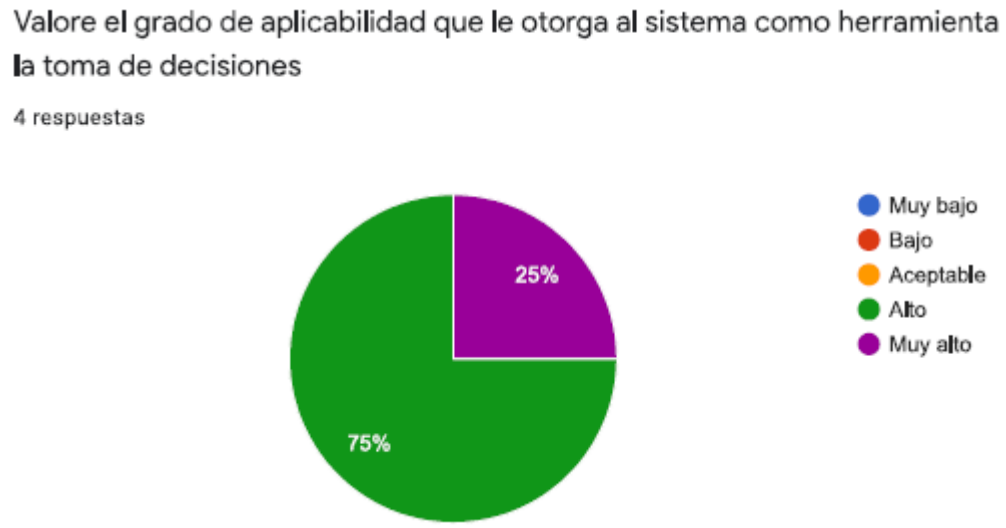

Figura 40. Evaluación de la aplicabilidad. Respuestas a la pregunta 1

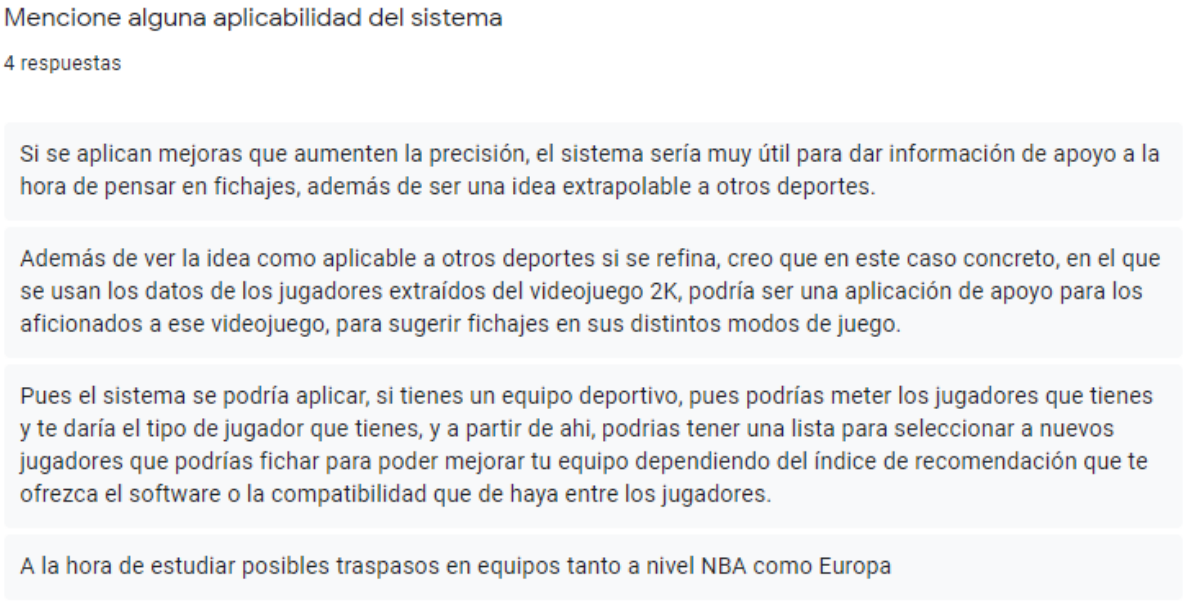

Figura 41. Evaluación de la aplicabilidad. Justificaciones a las respuestas

6. Conclusiones y trabajo futuro

6.1. Conclusiones

En esta sección se presentan las conclusiones derivadas de la realización del trabajo expuesto en esta memoria. Se dividen estas conclusiones en dos apartados, uno dedicado a las conclusiones técnicas y el otro a las personales.

Conclusiones técnicas

A nivel técnico, el objetivo de la realización de este proyecto era el de conocer nuevas tecnologías, aprender nuevos conceptos y ahondar en otros. Este objetivo se ha cumplido con creces ya que se ha podido profundizar en el conocimiento de MongoDB, la librería Scikit-learn o técnicas como el web scraping. Todo esto adquiere todavía más valor al tratarse de tecnologías o herramientas con mucha aceptación en el mundo profesional.

Además, el proyecto ha servido para conocer más a fondo tareas menos familiares, como la búsqueda de conjuntos de datos y el cruce y transformación de estos, así como el despliegue de software.

Conclusiones personales

A nivel personal, ya hablando en primera persona una vez finalizado el proyecto, puedo decir que ha sido duro. Duro no por su complejidad, sino por el contexto. Compaginar vida laboral y académica se vuelve una odisea cuando toca abordar un proyecto que requiere bastante implicación. Sin embargo, considero que todo el esfuerzo ha merecido la pena.

Como se ha mencionado anteriormente, la asimilación de nuevas tecnologías y la profundización en algunos conceptos han sido parte del proyecto y esto ha propiciado que algunas tareas hayan requerido la inversión de más tiempo del esperado. Estos hechos son los que aportan la experiencia para mejorar a nivel personal en aspectos como la estimación de tareas y la planificación.

Finalmente, reconozco que me hubiera gustado disponer de más tiempo para refinar algunas fases del desarrollo con el objetivo de mejorar los resultados. De todos modos, considero que los objetivos iniciales se han cumplido y se han aprendido lecciones que van más allá de lo técnico.

6.2. Líneas de trabajo futuro

Dadas las restricciones de tiempo para este proyecto, se han desarrollado funcionalidades con un comportamiento limitado. A continuación, se mencionan aspectos susceptibles de ser tratados en un futuro:

- **Mejoras en la implementación actual**
  - Clustering más preciso: Actualmente se clasifican los jugadores de cada una de las 5 posiciones en 4 clústeres. Una mejora consistiría en utilizar para cada posición el número de clústeres más conveniente, el que ayudase a clasificar con mayor precisión a los jugadores, además de valorar la posibilidad de utilizar diferentes atributos de entrada para cada posición o incluso aplicar otro algoritmo distinto a KMeans. Esta mejora contribuiría a conseguir una mayor fiabilidad a la hora de obtener recomendaciones de pares de jugadores a través de la herramienta.
- **Ideas para nuevas funcionalidades**
  - Recomendar conjuntos de un mayor número de jugadores: Una funcionalidad a añadir que supondría un gran valor añadido sería la de recomendar conjuntos de hasta 5 jugadores, que es el máximo que puede tener un equipo en pista.
  - Visionar clústeres en la interfaz: Un añadido importante a la interfaz de la herramienta sería una sección dónde el usuario pueda consultar los clústeres y ver a cuál pertenece cada jugador.
  - Recomendaciones configurables desde la interfaz: Relacionada con las dos anteriores, esta funcionalidad consistiría en dar soporte a que desde la interfaz de la herramienta se pueda configurar el tamaño del conjunto de jugadores compatibles a recomendar (desde 2 como actualmente, hasta un máximo de 5). Además, si la clasificación de jugadores por defecto no le satisface, también se podría permitir al usuario seleccionar el número de clústeres en los que quiere clasificar a los jugadores, aunque debe ser consciente de las repercusiones que eso puede llegar a tener en la precisión de las recomendaciones.
  - Automatización de procesos: Actualmente el clustering que se aplica sobre las características de los jugadores y el análisis de los datos de los partidos, así como la carga de esa información en una base de datos, son procesos que se inician de forma manual. Un valor añadido a este sistema sería la automatización de esas tareas.

7. Cumplimiento RGPD

Para el presente trabajo se han utilizado dos conjuntos de datos. El conjunto de datos relativo a los partidos de la NBA se ha obtenido de la plataforma Kaggle (Schmadamco, 2021), y se trata de un dataset público con los datos jugada a jugada de todos los partidos de la NBA que no contiene ningún dato de carácter personal.

En cuanto al conjunto de datos acerca de los atributos de los jugadores, éste se ha obtenido haciendo uso de la técnica de web scraping sobre el sitio web 2kratings (2K Ratings, 2017), como ya se ha descrito en el apartado 4.2. Teniendo en cuenta que la información publicada por la entidad es de acceso público, no se está violando ninguna ley o norma que prohíba el uso de técnicas de web scraping para la obtención de información pública. Además, ningún dato de los recabados es de carácter sensible.

8. Bibliografía

2K Ratings. (2017, 17 septiembre). *NBA 2K22 Ratings*. Recuperado 18 de noviembre de 2021, de https://www.2kratings.com/

Bioinformatics Laboratory, University of Ljubljana. (2016, 11 julio). *Data Mining*. Orange Data Mining - Data Mining. Recuperado 22 de febrero de 2022, de https://orangedatamining.com/

Conde, E. (2021, 10 diciembre). *GitHub - edconde/tfm-visualanalytics-bigdata: Trabajo de fin de máster del Máster en Análisis y Visualización de Datos Masivos*. GitHub. Recuperado 16 de diciembre de 2021, de https://github.com/edconde/tfm-visualanalytics-bigdata

Driblab. (2021, 3 febrero). *Jugadores similares: Cómo reemplazar a Modric, van Dijk, Sancho o Enzo Pérez*. Recuperado 18 de noviembre de 2021, de https://www.driblab.com/es/actualidad/jugadores-similares-como-reemplazar-a-modric-van-dijk-sancho-o-enzo-perez

ESPN Internet Ventures. (2009, 12 febrero). *NBA Trade Machine*. ESPN Deportes. Recuperado 17 de noviembre de 2021, de http://www.espn.com/nba/tradeMachine

Fanspo Inc. (2021, 11 junio). *Trade Machine & Cap Manager*. Fanspo. Recuperado 17 de noviembre de 2021, de https://fanspo.com/nba/trade-machine

Geron, A. (2020). Capítulo 9. Técnicas de aprendizaje no supervisado. En *Aprende Machine Learning con Scikit-Learn, Keras y Tensorflow* (2.ª ed., pp. 251–263). Anaya Multimedia.

Google. (2015, 5 marzo). *Angular*. Angular Docs. Recuperado 27 de diciembre de 2021, de https://angular.io/docs

Microsoft. (2021, 3 noviembre). *Visual Studio Code - Code Editing. Redefined*. Visual Studio Code. Recuperado 18 de noviembre de 2021, de https://code.visualstudio.com/

MongoDB, Inc. (2009, 5 febrero). *MongoDB Documentation*. MongoDB. Recuperado 18 de noviembre de 2021, de https://docs.mongodb.com/

Mongoose ODM v6.1.3. (2012, 7 agosto). Mongoose. Recuperado 27 de diciembre de 2021, de https://mongoosejs.com/

OpenJS Foundation. (2015, 15 marzo). *Express - Infraestructura de aplicaciones web Node.js*. Express. Recuperado 27 de diciembre de 2021, de https://expressjs.com/es/

OpenJS Foundation. (2020, 19 mayo). *Documentación*. Node.js. Recuperado 27 de diciembre de 2021, de https://nodejs.org/es/docs/

Orange Data Mining. (2015). *MDS — Orange Visual Programming 3 documentation*. Orange Visual Programming. Recuperado 22 de febrero de 2022, de https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/unsupervised/mds.html

Python Software Foundation. (2021a, mayo 3). *Browse the docs online or download a copy of your own*. Python.Org. Recuperado 18 de noviembre de 2021, de https://www.python.org/doc/

Python Software Foundation. (2021b, septiembre 8). *Beautifulsoup4*. PyPI. Recuperado 18 de noviembre de 2021, de https://pypi.org/project/beautifulsoup4/

Schmadamco. (2021, 22 enero). *NBA Play-by-Play Data 2015–2021*. Kaggle. Recuperado 18 de noviembre de 2021, de https://www.kaggle.com/schmadam97/nba-playbyplay-data-20182019

scikit-learn developers (BSD License). (2011a, noviembre 8). *2.3. Clustering*. Scikit-Learn. Recuperado 16 de diciembre de 2021, de https://scikit-learn.org/stable/modules/clustering.html

scikit-learn developers (BSD License). (2011b, noviembre 8). *sklearn.cluster.KMeans*. Scikit-Learn. Recuperado 18 de noviembre de 2021, de https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

Sports Reference LLC. (2020, 22 julio). *Stathead Basketball. The Most Powerful Research Tools in Sports*. Stathead. Recuperado 17 de noviembre de 2021, de https://stathead.com/basketball/

Sports Reference LLC. (2021, 20 septiembre). *Basketball Reference. Basketball Stats and History*. Basketball Reference. Recuperado 18 de noviembre de 2021, de https://www.basketball-reference.com

StatsBomb Services Ltd. (2020, 18 noviembre). *Doppelgängers: Encontrando jugadores similares*. StatsBomb. Recuperado 18 de noviembre de 2021, de https://statsbomb.com/es/2020/11/doppelgangers-encontrando-jugadores-similares/

The pandas development team. (2021, 17 octubre). *pandas documentation*. Pandas. Recuperado 18 de noviembre de 2021, de https://pandas.pydata.org/docs/