Recuperación de Información y Recomendaciones en la Web 2024

Maria Paula Gomez - 50868943 Gianni Testa - 4891674

1 de julio de 2024

1. Introducción

En la actualidad, los fanáticos de la NBA no solo disfrutan viendo los partidos, sino que también participan activamente en apuestas relacionadas con los jugares y equipos. Sportradar y la NBA anunciaron el lanzamiento de una funcionalidad mejorada relacionada con las apuestas dentro de NBA League Pass, impulsada por emBET, la solución over-the-top (OTT) de Sportradar. El objetivo de esta funcionalidad es integrar contenido de apuestas deportivas, como *spreads* (puntos de dispersión), $over - unders$ (totales) y *moneyline* (líneas de dinero), en la plataforma de transmisión en vivo de la NBA, mejorando la experiencia de apuestas en vivo. Los usuarios podrán optar por ver y seleccionar apuestas directamente en NBA League Pass y ser dirigidos a FanDuel o DraftKings, los socios oficiales de apuestas de la NBA, cuando estén listos para realizar una apuesta. La integración de emBET busca hacer que las apuestas en vivo sean más atractivas e inmersivas para los fanáticos de la NBA.

2. Problema

Pensando en las personas dispuestas a realizar estas apuestas directamente en la aplicación de streaming en vivo de la NBA (NBA League Pass), se ve la necesidad de disponibilizar una herramienta web que proporcione estadísticas detalladas donde se puedan aplicar filtros por jugadores o equipos, así como también filtrar por rango de tiempo y equipos enfrentados. De esta manera, se podrían obtener estadísticas específicas basadas en los filtros aplicados, proporcionando información útil a la hora de tomar decisiones en las apuestas.

3. Enfoque de la Solución

La propuesta es generar una aplicación web que disponga estadísticas detalladas de los jugadores de la NBA, incluyendo datos como la cantidad de triples realizados e intentados, partidos ganados, puntos, minutos jugados, robos de balón, rebotes, tiros libre, entre otros. La aplicación ofrecerá diversos filtros que permitirán organizar y visualizar la información de manera eficiente, tales como filtros por temporadas, equipos contrarios, y más. Esto permitirá a los usuarios tomar decisiones informadas al realizar apuestas en vivo.

4. Diseño

Para poder llevar a cabo la implementación de la solución se dispone el diseño de una aplicación web que se compone de 3 sistemas: Front-end, Back-end y base de datos, donde se distinguen dos sub-componentes a nivel del Back-end que son: una REST API para poder hacer las consultas pertinentes a las estadísticas y los datos necesarios para la misma, y un script con el que se realizará el scrapping para obtener los datos de la fuente de datos.

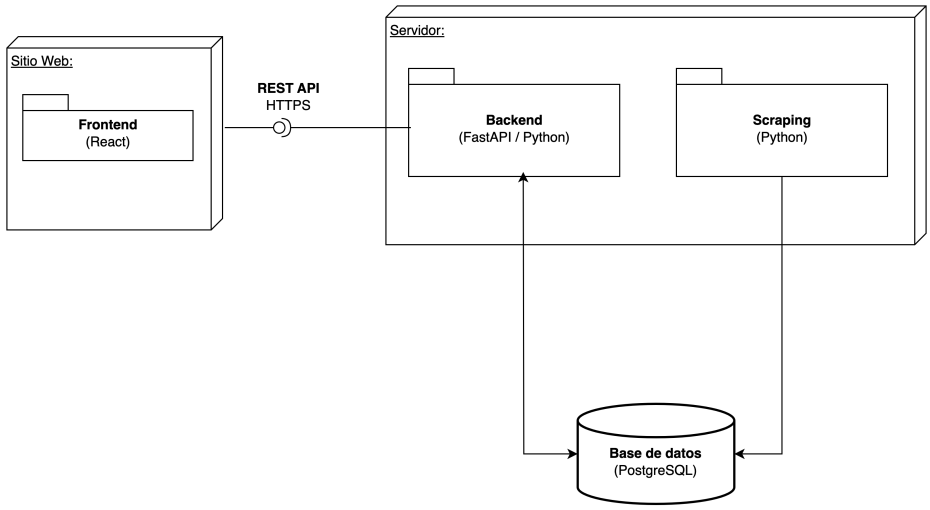

Figura 1: Diseño del sistema

Las tecnologías utilizadas son:

- **Front-end:** ReactJS.
- **Back-end:** FastAPI para la REST API, y Python para el script de scraping utilizando las librerías BeautifulSoap y Playwright.
- **Base de datos:** PostgreSQL.

La aplicación fue organizada con una arquitectura en capas, donde la capa mas de arriba consiste en la aplicación web react y luego las posteriores conforman el backend. A continuación las capas definidas:

- **Presentación:** Capa responsable de la UI (interfaz de usuario) de la aplicación, se incluyen todos los elementos visuales con los cuales interactúan los usuarios proporcionando una experiencia más atractiva.
- **Servicios** Capa responsable de alojar los servicios o APIs, recibe las solicitudes desde la capa de presentación e interactúa con la capa de negocio para proporcionar las respuestas. Se implementó utilizando una arquitectura RESTful
- **Negocio** Capa responsable de la lógica de la aplicación, recibe las solicitudes de la capa de servicios, las procesa y resuelve aplicando las reglas de negocio, se comunica con la capa de datos de ser necesario.
- **Datos** En esta capa se almacenan y gestionan los datos de la aplicación

4.1. Diagrama entidad-relación

A continuación se detalla el diagrama entidad relación definido en la base de datos.

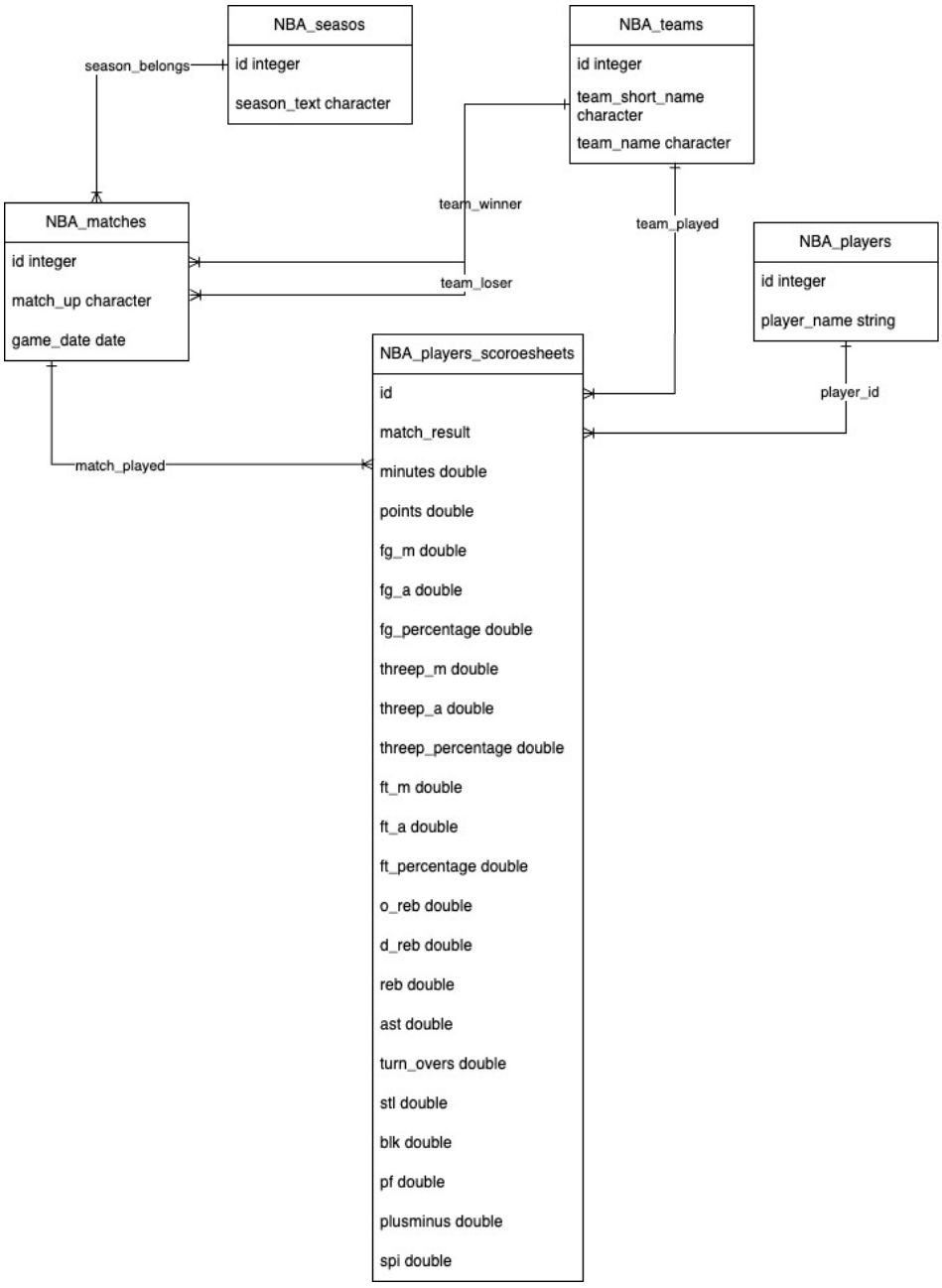

Figura 2: Diagrama entidad-relación

5. Implementación

Para la implementación construimos el backend utilizando Python y las liberrías FastApi para exponer la api REST. El scrapping fue realizado utilizando las librerías BeautifulSoup y Playwright. BeautifulSoap es una librería de python que permite extraer documentos HTML Y XML. Esta biblioteca crea un árbol con todos los elementos html extraídos de la página sobre el cual podemos iterar buscando lo que se desee. Playwright es una librería que permite automatizar las interacciones con un navegador, podemos por ejemplo contectarnos con distintos navegadores y acceder a la página deseada desde python de forma sincrónica o asincrónica.

En términos generales la implementación del scrapper consiste en conectarse con un navegador de web y navegar hasta la web de la NBA, utilizando las funcionalidades de *Playwright* y luego se extrae la información de las páginas HTML con la ayuda de $Beautiful Soup$.
Los datos fueron tomados directamente desde la página de la NBA y la lógica para implementar el scrapper fue basado en lo expuesto en este post de Medium.com, que nos sirvió de base y guía para la recolección de los datos desde la página oficial de la NBA.

Para el módulo de interacción la base de datos se utilizó SQLAlchemy, el cual provee un kit de harramientas SQL y un ORM (Object-Relational Mapping). Se utilizo el ORM provisto, el cual permite abstraerse de realizar consultas SQL en crudo, facilitando las consultas mas complejas.
Para poder realizar operaciones de búsqueda mas rápidas se agregaron índices en las claves mas relevantes:

Por último el frontend fue implementada utilizando ReactJS. React es una biblioteca de JavaScript, que permite la creación de interfaces de usuario interactivas, facilitando su desarrollo.

6. Funcionalidades y uso

A continuación se describen las funcionalidades de la aplicación web

- Listado de jugadores con paginado
- Filtro de jugadores por equipo en el listado
- Filtros en estadísticas:
  - Estadísticas por temporada
  - Estadísticas por equipo adversario
  - Estadísticas por equipo al cual pertenece
  - Estadísticas por partidos ganados o perdidos

7. Interfaz de usuario

A continuación se presentan algunas imágenes de la interfaz de usuario implementada.
En la figura [3] se presenta la página principal, al ingresar a la web es lo primero que vemos.

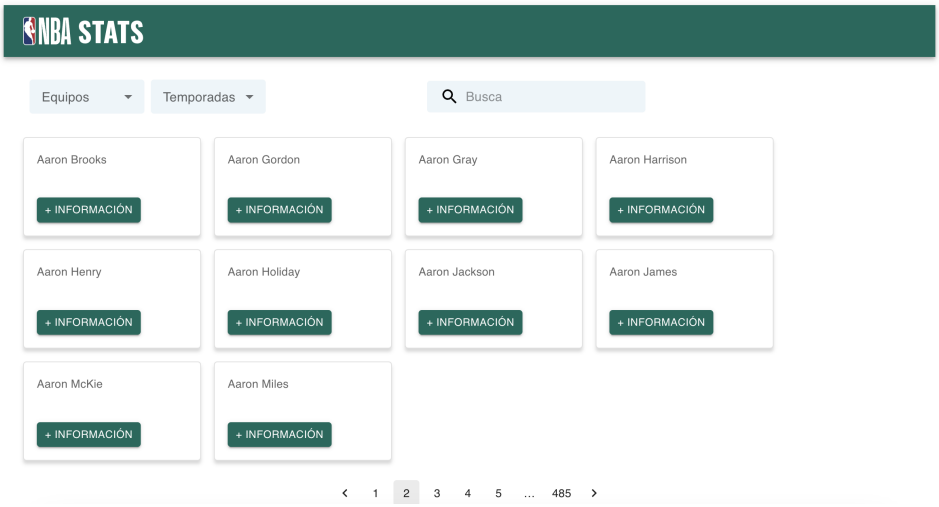

Figura 3: Pagína principal

Luego en la figura [4] podemos ver el buscador por jugador

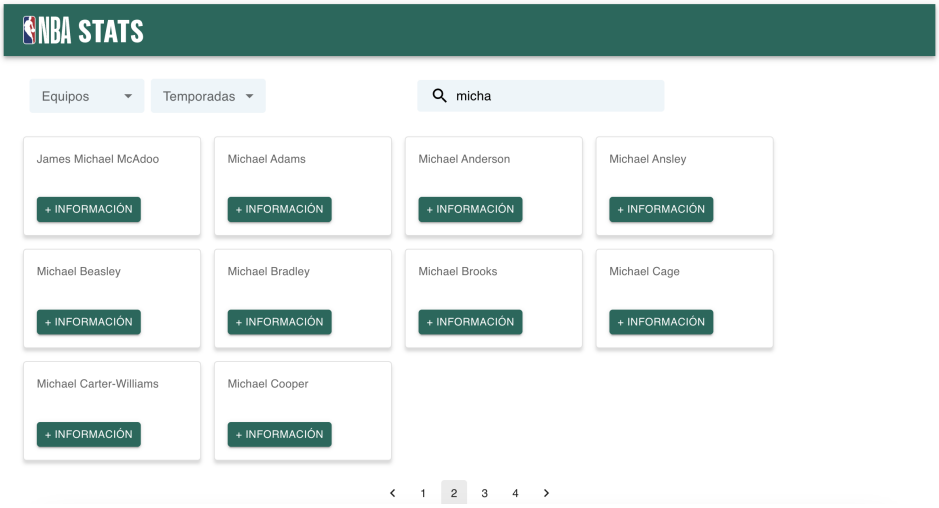

Figura 4: Buscador por jugador

Si seleccionamos el selector de la izquierda, como se visualiza en [5] se realizan los filtrados por equipo y por temporada.

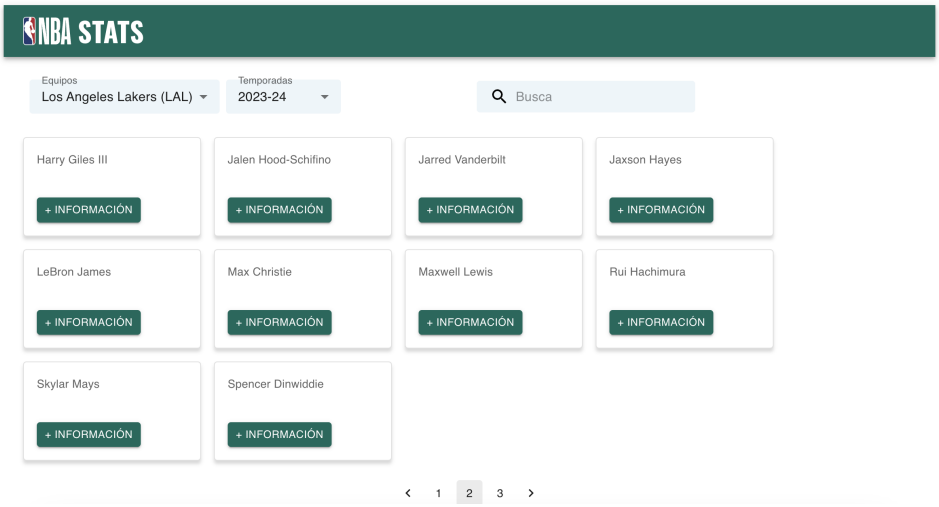

Figura 5: Juadores por equipo y/o por temporada

Luego dando en el botón de + *Informacion* pasamos al modal con información específica del jugador, aquí se detallan todas las estadísticas del mismo, y se seleccionan los filtros a aplicar. Además se incluye un glosario para saber a que hace referencia cada columna. Esto lo podemos ver en las figuras [6] y [7] donde se muestra un caso sin aplicar filtro, y otro con filtros aplicados respectivamente

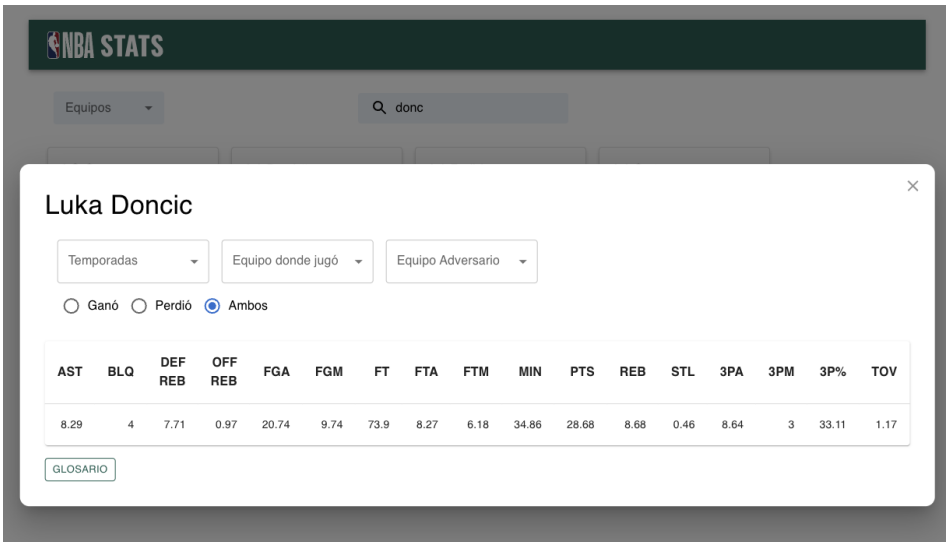

Figura 6: Estadísticas jugador sin filtro

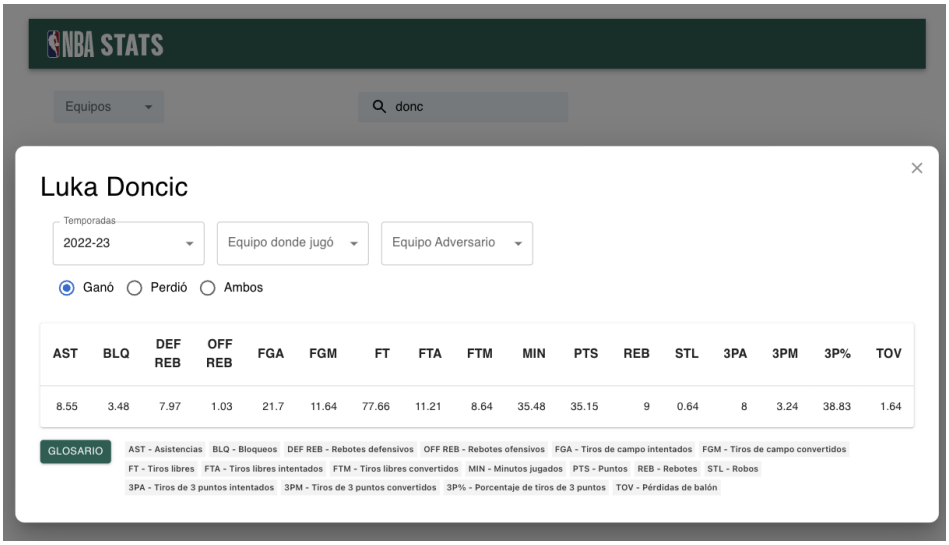

Figura 7: Estadísticas jugador con filtros

8. Evaluación y resultados

Luego de procesar todas las estadísticas disponibles, se obtuvieron los siguientes resultados:

- 1.312.181 de datos de estadísticas.
- 64.938 partidos.
- 4.847 jugadores.
- 74 equipos.
- 78 temporadas, que van desde la 1946-47 hasta la 2023-2024.

No solo el básquetbol como deporte si no también la NBA como liga han ido evolucionando con los años y es por eso que a medida que empezamos a ver datos de años anteriores, algunas estadísticas no existen. Por ejemplo, la diferenciación entre rebotes ofensivos y defensivos comienza en la temporada 1973-74, junto a los robos y los bloqueos. Las pérdidas se empiezan a contabilizar en la temporada 1977-78 y la existencia de los tiros de 3 puntos (triples) comienza en la temporada 1979-80.

9. Problemas encontrados

Los dos problemas principales que tuvimos durante el proceso de desarrollo están asociados al proceso de scraping.

- **Procesamiento de los datos:** Nos costo bastante tiempo en horas poder recolectar los datos, que son un total de 1.312.181 estadísticas. nos llevo al rededor de 3 días poder recolectar correctamente todos los datos, primero con algunos errores en el casteo de datos inexistentes, y también por el siguiente problema.
- **Timeouts:** Para recolectar los datos de cada una de las temporadas , se hacía scraping sobre una tabla con las estadísticas de esa temporada, y el tiempo de respuesta de la aplicación web de la NBA variaba mucho entre una temporada y otra, por lo que para muchas temporadas estuvimos intentando varias veces hasta poder determinar un timeout que no cierre el navegador antes de que se lograran cargar los datos.

Otro problema a resolver mas hacia el final del desarrollo eran los tiempos de respuesta, ya que las consultas tardaban bastante tiempo, y se logró solucionar fácilmente agregando índices tanto simples como de tuplas de las consultas que pueden generarse.

10. Conclusiones

En conclusión se logró implementar una aplicación que cumple con los objetivos planteados inicialmente. Se desarrolló un sisitema completo y con potencial crecimiento para la recolección, almacenamiento, procesamiento y visualización de datos estadísticos de la NBA.

Consideramos que se logró un diseño flexible y escalable, sobre el cual no se tendrían mayores dificultades para incorporar nuevas funcionalidades, como por ejemplo nuevos filtros.

Este trabajo permitió tener un acercamiento a lo que es el web scrapping, lo cual desconocíamos, y por consecuente a conocer algunas herramientas que permiten realizarlo. Además de enfrentarnos al desafío de transformar estos datos recuperados en HTML y, a través de limpieza y transformación, insertarlos en una base de datos estructurada.

11. Trabajo a Futuro

Existen varias posibilidades a incorporar a futuro.

En primer lugar se podrían incluir estadísticas generales a nivel equipo, ya que para el alcance del aplicativo creímos mas pertinente trabajar la información recuperada sobre las jugadores, la cual ya era bastante extensa. Sería una buena incorporación la recuperación de datos sobre equipos e incorporar filtros específicos para estos datos.

Otro punto en cuanto a los datos, en el alcance del proyecto restringimos los datos solo a estadísticas de los juegos de temporada regular, por lo que un trabajo a futuro sería agregar las estadísticas de los juegos de playoffs y su diferenciación con los de temporada regular.

Por otro lado se podrían incluir otros tipos de ordenamiento, como ordenar a los jugadores con mayor valor en cierta estadística, por ejemplo: mayor cantidad de puntos, mayor cantidad de asistencias, mayor cantidad de rebotes, entre otros.

Respecto al rendimiento, si ampliamos el conjunto de datos con información de equipos, se podría mejorar el motor de búsqueda incorporando una herramienta mas potente como Elasticsearch la cual permitiría realizar búsquedas mas veloces y realizar filtrados mas complejos.

Por último, a efectos de tener datos actualizados periódicamente se podría incorporar un *job* que ejecute cada cierto tiempo el módulo que se encargara de recuperar los datos de la web de la NBA e insertarlos en la base de datos.