

#>/<#

HACK A BOSS

Proyecto 1
#dsb07rt
enero 2024

Base del proyecto: Breve análisis de datos en la NBA

EXTRACCIÓN DE DATOS

Herramientas utilizadas:

- Web scraping
- APIs

Objetivo principal:

- Automatización de la extracción de datos

Estrategia de implementación:

- Desarrollo de código reutilizable para extracciones consistentes y adaptables

Consideraciones de alcance:

- Evaluación del tiempo y escala en función de las fuentes de datos seleccionadas

TRANSFORMACIÓN Y MANIPULACIÓN DE DATOS

Herramientas clave:

- Pandas
- NumPy

Objetivos principales:

- Limpieza de datos
- Estructuración de datos
- Creación de nuevas variables
- Automatización de la transformación de datos

Resultado esperado:

- Un DataFrame optimizado con variables relevantes, listo para análisis.

CARGA DE DATOS

Plataforma de gestión de datos:

- Airtable

Objetivo principal:

- Automatización del proceso de carga y extracción de datos en Airtable

Metodología:

- Desarrollo de funciones específicas para la gestión eficiente de datos en Airtable.

VISUALIZACIÓN DE DATOS

Herramientas de visualización:

- matplotlib
- seaborn
- . plotly
- folium

Objetivos principales:

- Creación de visualizaciones significativas a partir de los datos procesados

Enfoque:

- Selección de herramientas y gráficos basados en los datos y objetivos de representación.

STORYTELLING DEL PROYECTO

Motivaciones del proyecto:

- Inspiración y propósito detrás del análisis de datos de la NBA

Alcance del proyecto

- Definición de límites y metas del proyecto

Herramientas y tecnologías empleadas

- Repaso de las herramientas y métodos utilizados

Desafíos y soluciones

- Identificación y resolución de retos en cada fase del proyecto

Esquema general del proyecto

- Representación gráfica del flujo de trabajo y procesos

Visualizaciones y resultados finales

- Exhibición de los descubrimientos y visualizaciones más impactantes.



#>/<

**HACK
A BOSS**

Motivaciones del Proyecto

Interés en el deporte y datos de la NBA

El proyecto nace de una combinación de interés por el deporte, especialmente el deporte americano, y la fascinación que tienen los americanos por el conjunto de datos estadísticos en el deporte. Podía haber sido la NFL, la MLB o la NHL. Podían haber sido opciones igual de buenas. Incluso el deporte universitario en Estados Unidos hubiese sido una elección muy interesante. Al final nos decantamos por la NBA. También quiero reconocer ciertas ganas de salir de mi zona de confort, otros deportes me apasionan más, así que hacerlo con el baloncesto también lo veía como un reto personal y una dificultad añadida para comprender el funcionamiento del juego. Por ello, la elección de este deporte para el análisis se debe a su complejidad, dinamismo y, sobre todo, la disponibilidad de datos detallados y bien estructurados en su sitio web oficial. Este sitio web ofrece un enorme repositorio de información que representa una oportunidad única para explorar y analizar múltiples aspectos del juego.

Revelando historias y tendencias ocultas a través de los datos

Una de las principales motivaciones para este proyecto es descubrir cómo los datos pueden contar historias no vistas a simple vista. El análisis de estos datos no solo proporciona estadísticas y hechos, sino que también puede revelar tendencias, patrones y narrativas que están ocultas en los números. Estas historias pueden abarcar desde el rendimiento y la evolución de equipos y jugadores hasta la dinámica y estrategias de juego que definen la NBA. Es la magia de los datos en el deporte.

La importancia de la analítica en los deportes modernos

Finalmente, el proyecto busca destacar la creciente importancia de la analítica en los deportes modernos. En un mundo donde las decisiones estratégicas son cada vez más basadas en datos, entender y aplicar correctamente la analítica se ha convertido en una herramienta clave. En la NBA, como en muchos otros deportes, los análisis basados en datos no solo mejoran el entendimiento del juego, sino que también influyen en la toma de decisiones, desde el scouting de jugadores hasta la formulación de estrategias en la cancha.



#>/<

HACK
A BOSS

Alcance del proyecto

Límites y datos recogidos

El proyecto se centra en la temporada actual de la NBA, analizando datos actualizados hasta la fecha de hoy, 23 de enero de 2024. Hemos recopilado información específica que incluye las victorias y derrotas de los equipos, su clasificación, así como estadísticas detalladas de los jugadores de la temporada en curso. Además, hemos incorporado datos complementarios como equipos, dorsales, edad y país de procedencia de los jugadores.



Selección de fuentes de datos y análisis a realizar

La principal fuente de datos ha sido el sitio web oficial de la NBA, que ofrece una amplia gama de información estadística de temporadas pasadas y actuales. Nuestro enfoque se ha limitado a los datos de la temporada en curso, priorizando estadísticas clave de jugadores como minutos jugados, puntos, rebotes, entre otros. Con estos datos, planeamos realizar varios análisis, como identificar el top 10 de anotadores, el mejor rookie hasta la fecha y el mejor defensor de la liga, entre otros posibles estudios.

Objetivos específicos y desafíos en la extracción de datos

Los objetivos primordiales del proyecto fueron la recolección de datos utilizando herramientas como Selenium y BeautifulSoup para la extracción de datos de la web. A pesar de la aparente simplicidad inicial, nos enfrentamos a la complejidad de la estructura del sitio web de la NBA y a restricciones en el acceso a los datos, lo que nos llevó a ser temporalmente vetados del sitio. Esta experiencia resaltó la necesidad de ajustar nuestros métodos de extracción para evitar bloqueos, lo que implicó en un proceso de captura de datos que toma más de 100 minutos.



Herramientas y tecnologías usadas

Extracción de datos de la NBA

Para la extracción de datos, hemos empleado un conjunto de herramientas y técnicas de programación avanzadas, utilizando Python como lenguaje principal. A continuación, se detalla el proceso paso a paso:

Configuración inicial y acceso a la web:

Usamos Selenium con WebDriver para Chrome (`chromedriver.exe`) para navegar interactivamente por la web.

Accedemos a la página de juegos de la NBA (nba.com/games) y maximizamos la ventana para asegurar una visualización completa.

Implementamos una pausa (`sleep(1)`) para manejar la carga de la página y aceptamos la política de cookies.

NBA Player & Team Roster

ALL ACCESS. ALL THE TIME IS HAPPENING

League Pass Store Trends Sign In

PLAYERS | Franchise | Player Stats | Starting Lineups | Every Game Tracker | Transaction

Name	Year	Position	Weight	Last Refreshed
Przemek Karnowski	2019	6' 9"	243 lbs	May 10
Shane Larkin	2018	6' 0"	242 lbs	Pittsburgh
Ron Artest	2018	6' 0"	235 lbs	New Zealand
Ortak Ayvaz	1775	6' 0"	219 lbs	Turkey
Samuel Allende	2018	7' 0"	219 lbs	Luxembourg
Harold Alexander Wilson	2018	6' 0"	225 lbs	Virginia Tech
Gregory Allen	2018	6' 0"	210 lbs	Duke
Jaylen Miles	2018	6' 0"	205 lbs	Texas
John Alzamora	2018	6' 0"	170 lbs	Georgia Tech
Ryan Anderson	2018	6' 7"	230 lbs	USA

Navegación y selección de datos:

Navegamos a la sección de jugadores utilizando el método `find_element` con XPath y CSS Selectors para interactuar con elementos específicos de la página.

Seleccionamos la opción para listar todos los jugadores de la temporada actual.

Extracción de URLs de Jugadores:

Obtenemos el código fuente de la página actual con `browser.page_source`. Utilizamos BeautifulSoup para parsear el HTML y extraer las URLs de los perfiles de los jugadores.

Recopilación de datos de jugadores:

Definimos una función `extraer_informacion_jugador` que toma el HTML parseado y extrae datos relevantes como nombre, equipo, estadísticas, etc. Recorremos las URLs de los jugadores, obteniendo y procesando cada página individualmente.

Para evitar bloqueos por parte del servidor de la NBA, introducimos un retardo aleatorio entre solicitudes.

Creación del DataFrame y guardado de datos:

Consolidamos todos los datos extraídos en un DataFrame de pandas, estructurando la información en un formato por columnas.

Desarrollamos una función guardar_excel_con_numeracion para guardar los datos en archivos Excel sin sobrescribir información previa.



#>/< **HACK**
A BOSS

Herramientas y tecnologías usadas

Extracción de datos de los equipos de la NBA

En esta sección del proyecto, utilizamos una combinación de técnicas de web scraping y análisis de datos para recopilar información sobre los equipos de la NBA. A continuación, se detalla el proceso:

Configuración y navegación web:

Usamos Selenium con WebDriver para acceder a la página de la NBA y maximizamos la ventana del navegador.

Aceptamos la política de cookies tras una breve pausa para permitir la carga completa de la página.

Navegamos a la sección de equipos de la NBA usando el método `find_element` y `XPath`.

Extracción de URLs de los equipos:

Obtenemos el código fuente de la página de equipos utilizando `browser.page_source`.

Utilizamos BeautifulSoup para parsear el HTML y extraer las URLs de los perfiles de los equipos.

The screenshot shows the NBA official website's homepage. At the top, there are navigation links like 'Home', 'Schedule', 'News', 'Standings', 'Teams', 'Players', 'NBA Flag', 'All-Star', 'Future Stars', 'NBA Fitness', and 'Fantasy'. Below the header, a large banner reads 'ALL ACCESS, ALL THE TIME IS HAPPENING' with the NBA logo. The main content area is titled 'ALL TEAMS' and lists all 30 NBA teams. They are organized into four conferences: Atlantic, Central, Southeast, and Northwest. Each team has a small thumbnail, its name, and a link to its profile page. For example, the Atlantic conference includes the Boston Celtics, Brooklyn Nets, New York Knicks, Philadelphia 76ers, and Toronto Raptors. The Northwest conference includes the Denver Nuggets, Minnesota Timberwolves, Oklahoma City Thunder, Portland Trail Blazers, Golden State Warriors, LA Clippers, Los Angeles Lakers, and Phoenix Suns. The Central conference includes the Chicago Bulls, Cleveland Cavaliers, Detroit Pistons, Indiana Pacers, and Milwaukee Bucks. The Southeast conference includes the Atlanta Hawks, Charlotte Hornets, Miami Heat, Orlando Magic, and Washington Wizards.

Recopilación de información de equipos:

Definimos una función `extraer_informacion_equipo_bs4` para procesar cada URL y extraer detalles relevantes como el nombre del equipo, registro de victorias y derrotas, puesto, estadísticas clave (PPG, RPG, APG, OPPG), entre otros.

Recorremos cada URL de los equipos, aplicando la función definida para obtener la información deseada.

The screenshot shows a Jupyter Notebook cell with Python code. The code imports necessary modules and defines a function `extraer_informacion_equipo_bs4` that takes a URL as input. Inside the function, it prints the URL, creates an empty list `info_equipo`, and then iterates through several variables to append them to the list: `id_equipo.append(info_equipo['id_equipo'])`, `nombre_equipo.append(info_equipo['nombre_equipo'])`, `victorias_derrotas.append(info_equipo['victorias_derrotas'])`, `puesto.append(info_equipo['puesto'])`, `division.append(info_equipo['division'])`, `ppg.append(info_equipo['ppg'])`, `rpg.append(info_equipo['rpg'])`, `apg.append(info_equipo['apg'])`, and `oppg.append(info_equipo['oppg'])`. Finally, it uses `sleep(uniform(2, 4))` to add a random delay between requests. Below the code, a terminal window shows the browser opening multiple URLs from the NBA stats page, such as `https://www.nba.com/stats/team/16106012738` and `https://www.nba.com/stats/team/16106012750`.

Creación del DataFrame y Almacenamiento de Datos:

Almacenamos la información recolectada en un DataFrame de pandas, estructurando los datos en un formato tabular para un análisis más sencillo.

Implementamos un mecanismo de guardado en archivos Excel utilizando una función que evita la sobrescritura de datos existentes (`guardar_excel_con_numeracion`).



Herramientas y tecnologías usadas

Extracción y descarga de escudos de los equipos de la NBA

Acceso a la web y cookies:

Utilizamos Selenium con WebDriver para Chrome para abrir y maximizar la ventana del navegador.
Accedemos a la página de juegos de la NBA (nba.com/games) y aceptamos la política de cookies para asegurar una navegación fluida.

Navegación a la sección de equipos:

Navegamos a la página de equipos de la NBA mediante Selenium, utilizando el método `find_element` y XPath para seleccionar el enlace correspondiente.

Obtención de URLs de los equipos:

Capturamos el código fuente de la página con `browser.page_source`.
Aplicamos BeautifulSoup para parsear el HTML y extraemos las URLs de los perfiles de los equipos de la NBA.

Función para la extracción y descarga:

Creamos una función `extraer_logos_equipo_bs4` para procesar cada URL y obtener el identificador del equipo y la URL de su logo.
Desarrollamos otra función `descargar_logo` para descargar el logo del equipo basándonos en su URL.

Proceso de extracción y descarga de logos:

Preparamos una carpeta (`logos_equipos`) para almacenar los logos descargados. Si no existe, la creamos.
Recorremos cada URL de equipo, aplicando `extraer_logos_equipo_bs4` para obtener el ID del equipo y la URL de su logo.
Utilizamos `descargar_logo` para guardar cada logo en formato SVG en la carpeta designada, nombrando los archivos según el ID del equipo.

```
browser = webdriver.Chrome()
browser.get("https://nba.com/games")
browser.maximize_window()

#cookies
sleep(3)
browser.find_element(by=By.ID, value="onetrust-accept-btn-handler").click()

#ir a la página de equipos
sleep(2)
browser.find_element(by=By.XPATH, value='//*[@id="nav-ul"]/li[7]/a').click()

html_equipos = browser.page_source
soup = BeautifulSoup(html_equipos, "html.parser")
urls_equipos_nba = [a['href'] for a in soup.find_all('a', class_='_Anchor_anchor_cSc3P_TeamFigureLink_teamFigureLink_uqnh0') if '/stats/team/' in a['href']]

print(urls_equipos_nba)

['/stats/team/1610612738', '/stats/team/1610612751', '/stats/team/1610612752', '/stats/team/1610612755', '/stats/team/1610612761', '/stats/team/1610612741', '/stats/team/1610612742']

import os
import requests
from bs4 import BeautifulSoup

url_base = "https://www.nba.com"
web_equipos_nba = [url_base + equipo_url for equipo_url in urls_equipos_nba]

# Función para extraer la URL del logo de un equipo
def extraer_logos_equipo_bs4(url):
    try:
```



Herramientas y tecnologías usadas

Extracción de estadísticas de jugadores de la NBA utilizando su ID

Acceso a la web y cookies:

Usamos Selenium con WebDriver para Chrome para abrir y maximizar la ventana del navegador.

Accedemos a la página de juegos de la NBA (nba.com/games) y aceptamos la política de cookies para asegurar una navegación fluida.

Navegación a la sección de estadísticas:

Navegamos a la sección de estadísticas de la temporada y luego a la opción de líderes de la temporada.

Extracción de URLs de estadísticas de jugadores:

Obtenemos el código fuente de la página con `browser.page_source`. Utilizamos BeautifulSoup para parsear el HTML y extraer las URLs de las páginas de estadísticas de los jugadores líderes.

Función para la extracción de datos:

Creamos una función `nombre_columnas_estadisticas` para extraer los nombres de las columnas de la tabla de estadísticas. Desarrollamos otra función `extraer_datos_jugadores` para extraer datos específicos de cada jugador, incluyendo el ID del jugador y del equipo, obtenido de los enlaces.

Proceso de extracción de datos de jugadores:

Accedemos a la primera URL de estadísticas de jugadores y ajustamos la vista para mostrar todas las estadísticas y jugadores. Utilizamos `nombre_columnas_estadisticas` y `extraer_datos_jugadores` para obtener los nombres de las columnas y los datos respectivos de cada fila de la tabla de estadísticas.

```
df_estadisticas_puntos_jugadores = pd.DataFrame(datos_jugadores, columns=nombre_columnas)

print(df_estadisticas_puntos_jugadores)
```

[531 rows x 26 columns]

#	Player	TEAM	GP	MIN	PTS	FGM	FGA				
0	Giannis Antetokounmpo	1610612749	41	1444	1278	473	784			
1	Shai Gilgeous-Alexander	1610612760	41	1415	1274	452	823		
2	Luka Doncic	1610612742	36	1327	1288	412	854		
3	Joel Embiid	1610612755	32	1096	1156	386	716		
4	Nikola Jokic	1610612743	43	1443	1122	437	744		
...		
526	Omur Alp Bitim	1610612741	1	3	0	0	0		
527	Pete Nance	1610612739	1	3	0	0	0	2		
528	Ron Harper Jr.	1610612761	1	4	0	0	0	0		
529	Ryan Arcidiacono	1610612752	16	36	0	0	0	5		
530	Usman Garuba	1610612744	2	6	0	0	0	1		
			FG%	3PM	...	DREB	REB	AST	STL	BLK	TOV	PF	EFF	AST/TOV	STL/TOV												
0	60.3	17	361	472	251	56	46	149	119	1487	1.69	0.38	
1	54.9	46	196	238	257	91	32	82	105	1392	3.13	1.11	
2	48.2	139	278	307	334	51	21	141	65	1266	2.37	0.36	
3	53.9	37	283	370	188	37	68	116	92	1321	1.62	0.32	
4	58.7	43	384	513	391	50	40	124	113	1641	3.15	0.40	
...	
526	0.0	0	0	0	0	0	0	0	0	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
527	0.0	0	0	0	0	0	0	0	0	0	-2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
528	0.0	0	0	0	1	0	0	0	2	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
529	0.0	0	7	7	4	1	0	0	4	7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
530	0.0	0	0	0	1	1	0	0	1	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Creación del DataFrame y almacenamiento de datos:

Almacenamos los datos en un DataFrame de pandas, proporcionando una estructura tabular para su análisis. Implementamos un mecanismo de guardado en archivos Excel utilizando una función que evita la sobreescritura de datos existentes (`guardar_excel_con_numeracion`).

Aspectos clave del proceso:

La extracción del ID único del jugador y del equipo es crucial para relacionar estos datos con otros conjuntos de datos, permitiendo un análisis más profundo y conexiones entre diferentes aspectos estadísticos.



#>/<

HACK
A BOSS

Herramientas y tecnologías usadas

Manipulación y actualización de datos con Pandas y NumPy

Funcionalidad general de las funciones creadas:

Las funciones desarrolladas tienen como objetivo actualizar y consolidar los datos de equipos y jugadores de la NBA a partir de múltiples archivos Excel. Se han creado dos funciones, que hacen lo mismo, pero para cada uno de los datos ya sean equipos o jugadores.

Obtener archivos ordenados (obtener_archivos_ordenados):

Identifica y ordena archivos Excel dentro de un directorio específico basándose en un prefijo común.

Utiliza expresiones regulares para ordenar los archivos según su numeración, asumiendo que el número más alto corresponde al archivo más reciente.

Actualizar datos (actualizar_datos_nba_modificado y actualizar_datos_estadísticas):

Crea un DataFrame vacío para almacenar los datos consolidados.

Itera sobre cada archivo Excel y lo carga en un DataFrame (df_actual).

Reemplaza los valores 'None' con pd.NA para manejar adecuadamente los datos faltantes.

Utiliza un bucle para fusionar (merge) cada DataFrame nuevo con el DataFrame final (df_final) basándose en la columna clave (ID Equipo o id_jugador).

En cada iteración, actualiza el DataFrame final con los datos más recientes, conservando los datos antiguos cuando no hay datos nuevos.

Utiliza pd.merge y lambda para actualizar y limpiar los datos.

* Cómo podemos ir actualizando los datos de los jugadores y equipos a lo largo de la temporada, la idea es incorporar esos datos últimos a la tabla, pero también hemos contemplado que puede haber un error en la descarga de datos por bs4 por ello si no hay actualización de datos que deje el antiguo sin modificar.

Preparación del directorio y archivos:

Establece la ruta del directorio donde se almacenan los archivos Excel y define el prefijo del archivo.

Obtiene una lista de archivos ordenados utilizando la función obtener_archivos_ordenados.

Actualización de datos y creación de nuevos archivos excel:

Aplicar la función de actualización de datos adecuada (actualizar_datos_nba_modificado o actualizar_datos_estadísticas) para obtener un DataFrame actualizado.

Crea un directorio para almacenar los archivos actualizados si no existe. Guarda el DataFrame actualizado en un nuevo archivo Excel, asegurando que los datos más recientes estén disponibles para análisis futuros.

A	B	C	D	E	F	G	H	I	J	K	L
4	1.61E+09	New York	26 - 17	5th	Eastern	115.2	46.3	24.0	110.6		
5	1.61E+09	Philadelphia	29 - 13	3rd	Eastern	119.8	43.7	25.4	111.5		
6	1.61E+09	Toronto	8 - 26 - 28	12th	Eastern	114.3	43.6	29.6	116.4		
7	1.61E+09	Chicago	21 - 24	9th	Eastern	110.6	43.6	24.4	112.1		
8	1.61E+09	Cleveland	26 - 15	4th	Eastern	114.0	45.1	26.9	109.7		
9	1.61E+09	Detroit	Pis 4 - 39	15th	Eastern	112.4	43.3	26.4	123.0		
10	1.61E+09	Indiana	Pa 24 - 19	7th	Eastern	125.0	40.5	31.1	123.1		
11	1.61E+09	Milwaukee	25 - 12	2nd	Eastern	124.1	44.1	26.4	119.9		
12	1.61E+09	Atlanta	Ha 18 - 25	10th	Eastern	119.7	44.9	25.4	122.4		
13	1.61E+09	Charlotte	110 - 31	13th	Eastern	108.8	43.3	24.9	119.9		
14	1.61E+09	Miami	Heat 24 - 19	6th	Eastern	110.9	41.6	25.7	110.7		
15	1.61E+09	Orlando	M 23 - 21	8th	Eastern	111.4	43.4	24.5	111.2		
16	1.61E+09	Washington	7 - 35	14th	Eastern	115.4	40.0	27.8	124.9		
17	1.61E+09	Denver	Nu 30 - 14	3rd	Western	115.9	44.0	29.3	110.9		
18	1.61E+09	Minnesota	30 - 13	1st	Western	113.4	43.9	26.4	107.8		
19	1.61E+09	Oklahoma	24 - 11	2nd	Western	122.2	40.7	26.9	114.6		
20	1.61E+09	Portland	T 12 - 30	14th	Western	107.5	42.4	22.1	117.1		
21	1.61E+09	Utah	Jazz 22 - 22	10th	Western	117.3	46.7	28.1	118.7		
22	1.61E+09	Golden	St 18 - 22	12th	Western	117.2	46.1	28.2	118.2		
23	1.61E+09	LA Clippers	27 - 14	4th	Western	117.8	43.1	25.8	112.0		
24	1.61E+09	Los Angeles	22 - 22	9th	Western	114.8	43.2	27.9	115.6		
25	1.61E+09	Phoenix	S 25 - 18	5th	Western	116.1	43.3	26.3	114.2		
26	1.61E+09	Sacramento	24 - 18	7th	Western	118.3	43.7	28.7	117.7		
27	1.61E+09	Dallas	Ma 22 - 15	6th	Western	119	42.4	25.4	116.8		
28	1.61E+09	Houston	R 20 - 22	11th	Western	112.5	45.2	24.8	112.1		
29	1.61E+09	Memphis	I 16 - 27	13th	Western	107.8	42.2	24.8	113.3		



Herramientas y tecnologías usadas

Limpieza y fusión de datos de jugadores de la NBA con Pandas

Carga de archivos excel:

Utilizamos pandas.read_excel para cargar dos conjuntos de datos de jugadores desde archivos Excel diferentes (jugadores_nba.xlsx y estadistica_jugadores_nba_actualizado.xlsx).

Inspección inicial de columnas:

Imprimimos los nombres de las columnas de ambos DataFrames para identificar diferencias y similitudes.

Limpieza de columnas innecesarias:

Eliminamos las columnas 'Equipo' de df1 y 'Player' de df2 que no son necesarias para el análisis.

Limpieza y normalización de datos:

Limpiamos y normalizamos datos en varias columnas ('Dorsal', 'Posición', 'PPG', 'RPG', 'APG', 'PIE', 'Altura', 'Peso', 'Edad', 'Experiencia', 'Cumpleaños'), aplicando diferentes técnicas:

Remoción de caracteres no deseados.

Conversión de valores a numéricos.

Uso de expresiones regulares para extraer información específica.

Reemplazo de valores faltantes o incorrectos por None.

Cambio de nombres de columnas para claridad.

Conversión de fechas a un formato más reconocible.

Transformación de datos de acuerdo a necesidades específicas:

Modificamos los datos para adaptarlos a los requisitos del análisis, como cambiar nombres de posiciones al español y estandarizar formatos de altura y peso.

Fusión de DataFrames:

Fusionamos df1 y df2 usando 'id_jugador' como clave. Esto nos permite combinar información relevante de ambos conjuntos de datos en un solo DataFrame.
Manejamos posibles discrepancias en los datos resultantes de la fusión.

Reordenamiento y finalización:

Reordenamos las columnas para una presentación más lógica y coherente de los datos.
Rellenamos valores NaN resultantes de la fusión con 'None' para mantener la consistencia.

Almacenamiento de datos limpios y fusionados:

Guardamos el DataFrame resultante en un archivo Excel nuevo para su uso en análisis posteriores.



Herramientas y tecnologías usadas

Airtable para la gestión de datos.

Configuración de credenciales y URL de Airtable:

Definimos las credenciales de la API de Airtable (API_KEY), la identificación de la base (BASE_ID) y la tabla (TABLE_ID).
Configuramos la URL base de Airtable y los encabezados de la solicitud HTTP con la clave API y el tipo de contenido.

Carga de datos desde archivos excel:

Utilizamos pandas para cargar datos de dos archivos Excel (equipos_nba_actualizado.xlsx y jugadores_completos.xlsx).
Reemplazamos los valores NaN de pandas por None para asegurar la compatibilidad con la API de Airtable.

Preparación de datos para envío:

Preparamos los datos para el envío, creando un diccionario con los registros, donde cada registro es convertido a un diccionario con pandas to_dict().
Limitamos la cantidad de registros a enviar en cada solicitud a 10, debido a las restricciones de la API de Airtable.

Función de envío a Airtable:

Definimos una función enviar_a_airtable para hacer solicitudes POST a la API de Airtable.
Dentro de esta función, manejamos la respuesta para verificar el éxito del envío.

Envío de datos en grupos:

Enviamos los datos en grupos de 10 utilizando un bucle. Esto es para cumplir con la limitación de registros que se pueden enviar en una sola solicitud a Airtable.

Repetición del proceso para diferentes conjuntos de datos:

Repetimos el proceso para dos conjuntos de datos diferentes: uno para los equipos y otro para las estadísticas de los jugadores.

Creación manual de columnas en Airtable:

Un desafío importante fue la necesidad de crear las columnas manualmente en Airtable.
Tras estar intentando subir datos comprobé que no funcionaba, no había manera de hacerlo, por lo que solicité ayuda al tutor que me indicó que había que hacer el nombre de las columnas manualmente.
Se copiaron los títulos del Excel y se pegaron en Airtable para asegurar la correcta correspondencia de los datos.



#>/<

**HACK
A BOSS**

Herramientas y tecnologías usadas

Descarga y manipulación de datos de Airtable con su API y Pandas

Configuración de credenciales y URL de Airtable:

Definimos las credenciales de la API de Airtable (API_KEY), la identificación de la base (BASE_ID) y la tabla (TABLE_ID).
Configuramos la URL base de Airtable y los encabezados de la solicitud HTTP con la clave API y el tipo de contenido.

Establecimiento del endpoint y parámetros:

Establecemos el endpoint de la API de Airtable para acceder a los datos de las tablas de equipos y jugadores.
Configuramos parámetros adicionales para la solicitud HTTP.

Bucle para descargar datos:

Creamos un bucle para obtener datos de Airtable en bloques, manejando el parámetro 'offset' para iterar a través de todos los registros disponibles.

Normalización y creación de DataFrames:

Utilizamos pd.json_normalize para transformar los datos descargados en un formato tabular adecuado para pandas.

pd.json_normalize(datos_jugadores_0bs)														
	id	createdTime	fields.id_jugador	fields.Nombre	fields.Apellido	fields.ID_Equipo	fields.Dorsal	fields.Posición	fields.PPG	fields.RPG	fields.DREB	fields.REB	fields	lyte
0	rec01Djsg9tUsnCmc	2024-01-11T14:33:11.000Z	1630346	Matt	Ryan	1610612740	37	Alero	9.3	2.1	..	25	30	
1	rec02PRowzgg3B0u	2024-01-11T14:33:33.000Z	203468	Cl	McCollum	1610612740	3	Base	20	4.3	..	86	102	
2	rec03JyVwWf30NgyM	2024-01-11T14:32:30.000Z	203465	Jonas	Valentiniuc	1610612740	17	Pivot	14	9.8	..	272	361	
3	rec0CWx0PbzgMmMF	2024-01-11T14:32:47.000Z	1641709	Ausar	Thompson	1610612765	9	Base - Alero	8.3	6.7	..	166	247	
4	rec03ReoF1EE516K	2024-01-11T14:33:39.000Z	1641713	GG	Iacobon	1610612763	45	Alero	1.6	0.6	..	2	3	
..
516	rec0f42oVWwRfKgQo	2024-01-11T14:33:39.000Z	1641706	Brandon	Miles	1610612766	24	Alero	14.3	3.9	..	92	116	
517	rec0f0neUyptwE13	2024-01-11T14:32:51.000Z	1630202	Peyton	Pritchard	1610612758	11	Base	7.5	3.3	..	79	117	
518	rec0jYC80nV9Y78oN	2024-01-11T14:32:25.000Z	203924	Jessie	Grant	1610612757	9	Alero	21.8	3.6	..	93	113	
519	rec0jWRW9yfOEWai	2024-01-11T14:32:37.000Z	1629661	Cameron	Johnson	1610612751	2	Alero	14.9	4.9	..	106	142	
520	rec0v1mjB8a7Mdr	2024-01-11T14:33:42.000Z	1630604	E.J.	Liddell	1610612740	32	Alero	1	0.5	..	2	2	

Almacenamiento inicial de datos:

Guardamos los DataFrames iniciales en archivos Excel para su posterior manipulación y análisis.

Limpieza y reestructuración de DataFrames:

Realizamos la limpieza de los DataFrames para eliminar columnas innecesarias ('id', 'createdTime') y para reestructurar los datos a un formato más adecuado.
Utilizamos pd.concat para combinar los DataFrames con los datos normalizados, excluyendo las columnas que no son de interés.

Guardado final de DataFrames:

Guardamos los DataFrames reestructurados y limpios en nuevos archivos Excel, preparados para un análisis más conveniente.



Herramientas y tecnologías usadas

Herramientas de visualización (matplotlib, seaborn, plotly, folium).

Carga de datos desde excel:

Importamos los datos de los equipos y jugadores de la NBA desde archivos Excel previamente guardados (datos_equipos_nba.xlsx y datos_jugadores_nba.xlsx).

Preparación de datos de equipos para gráficos:

Separamos la columna 'Victorias-Derrotas' en dos columnas independientes 'Victorias' y 'Derrotas'.

Convertimos los valores de estas columnas a enteros para facilitar su visualización.

Creación de gráfico de barras con Matplotlib:

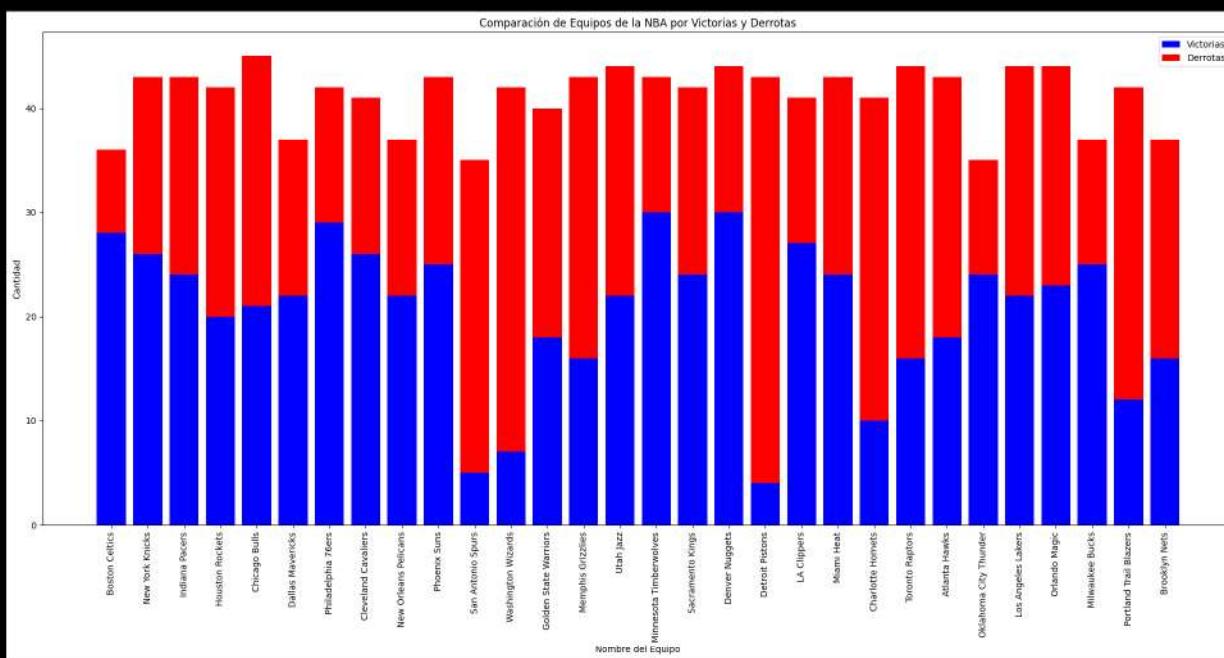
Comparación de equipos por victorias y derrotas:

Utilizamos Matplotlib para crear un gráfico de barras apiladas que muestra las victorias y derrotas de cada equipo.

Configuramos el tamaño del gráfico, los colores de las barras, etiquetas, leyenda y rotación de las etiquetas del eje x para una mejor visualización.

Representación del gráfico:

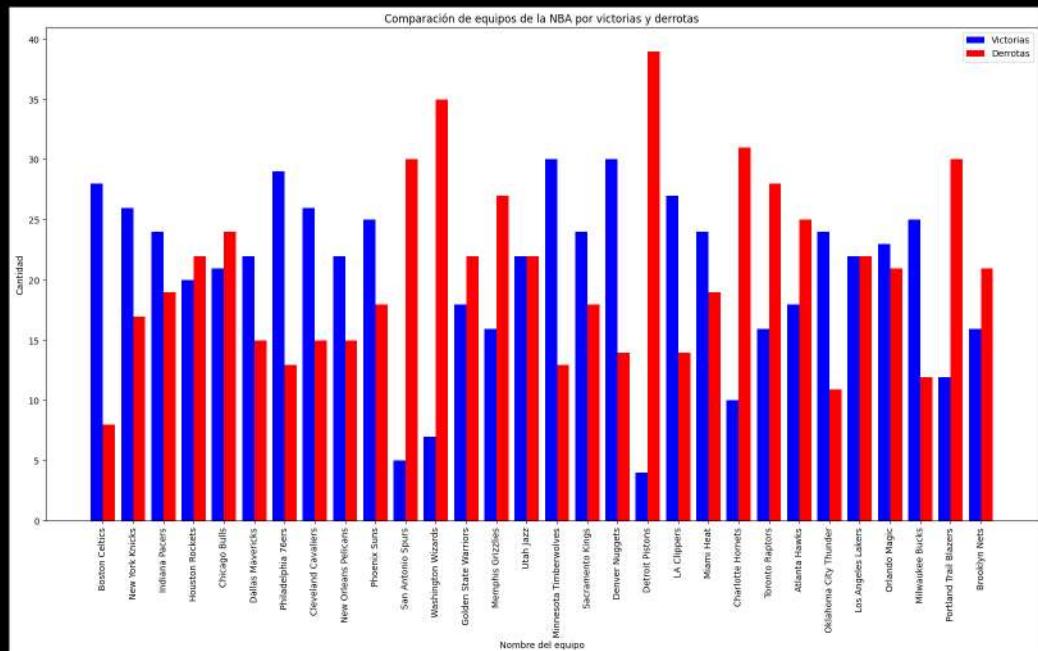
Representamos los datos de los primeros cinco equipos como muestra. Por ejemplo, el gráfico mostraría que los Boston Celtics tienen 28 victorias y 8 derrotas, los New York Knicks 26 victorias y 17 derrotas, y así sucesivamente. La barra azul representa las victorias y la roja las derrotas, mostrando una comparación clara del rendimiento de cada equipo.



Herramientas y tecnologías usadas

Herramientas de visualización (matplotlib, seaborn, plotly, folium).

En este gráfico, se presenta una variante del gráfico anterior de barras para visualizar las victorias y derrotas de los equipos de la NBA, utilizando Matplotlib. A diferencia del gráfico anterior, aquí se opta por barras adyacentes en lugar de apiladas para comparar las victorias y derrotas



Dimensiones y ancho de barras:

Se establece el tamaño del gráfico mediante plt.figure(figsize=(20, 10)). Se define un ancho para las barras (ancho_barra = 0.4) que determina cuán gruesas serán en el gráfico.

Posicionamiento de las barras:

Utilizamos np.arange para generar una secuencia de posiciones basada en el número de equipos. Esto sirve como base para ubicar las barras en el eje X.

Las barras de victorias se posicionan ligeramente a la izquierda del centro de cada equipo (posiciones - ancho_barra / 2), mientras que las de derrotas se colocan a la derecha (posiciones + ancho_barra / 2).

Creación de barras para victorias y derrotas:

Las barras de victorias se dibujan en azul y las de derrotas en rojo, utilizando el método plt.bar.

Se asigna el mismo ancho a ambas barras y se alinean de acuerdo a las posiciones calculadas.

Ventajas de este enfoque:

Claridad en la comparación: Al colocar las barras de victorias y derrotas una al lado de la otra para cada equipo, se facilita la comparación directa entre ambas.

Estética y legibilidad: Este método puede ser más estético y fácil de leer, especialmente cuando hay una gran diferencia en los valores de victorias y derrotas.



>/<

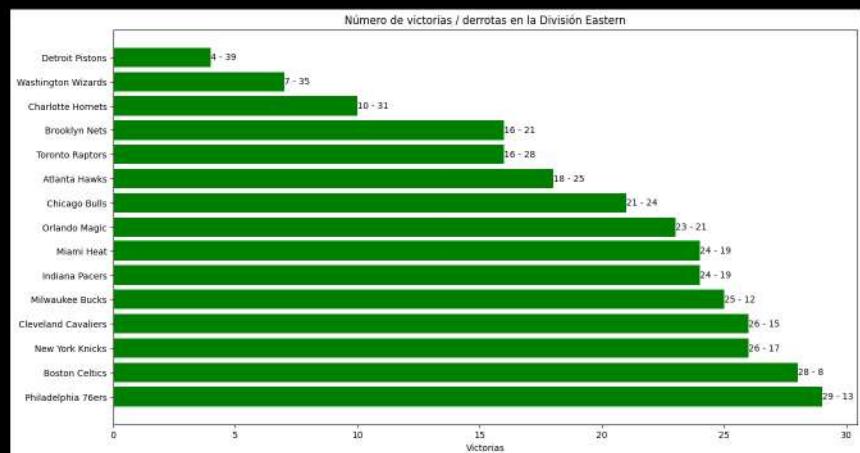
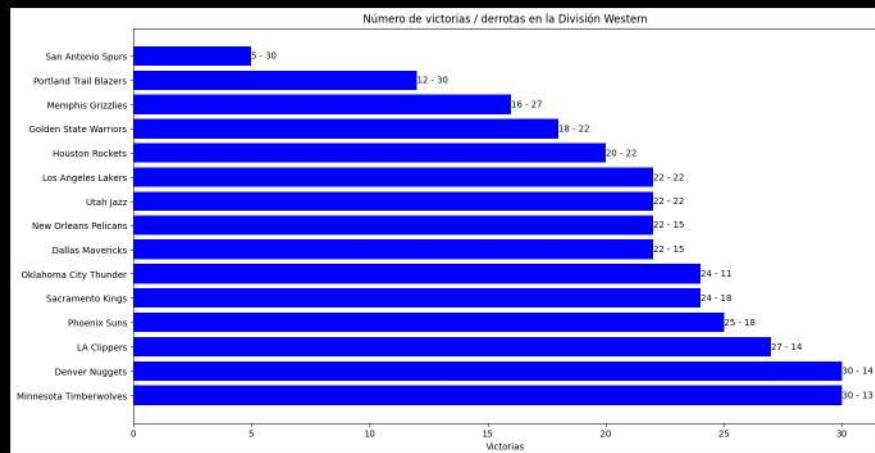
**HACK
A BOSS**

Herramientas y tecnologías usadas

Herramientas de visualización (matplotlib, seaborn, plotly, folium).

Seguimos con las victorias y derrotas de los equipos, pero ahora los hemos separado por divisiones, la Western y la Eastern.

Además hemos puesto en cada barra el número de victorias - derrotas de cada equipo.



>/<

**HACK
A BOSS**

Herramientas y tecnologías usadas

Herramientas de visualización (matplotlib, seaborn, plotly, folium).

Visualización geográfica de equipos de la NBA con Folium

Web Scraping de la Wikipedia:

Realizamos web scraping en una página de Wikipedia para obtener nombres de equipos, ciudades y pabellones de la NBA.

Utilizamos BeautifulSoup para parsear el contenido HTML y extraer los datos requeridos.

Preparación de DataFrames con datos de equipos:

Creamos DataFrames para equipos de conferencias Oeste y Este, y los combinamos en un solo DataFrame.

Normalización de nombres de equipos:

Ajustamos los nombres de los equipos para que coincidan con los datos de Foursquare, limpiando y estandarizando los nombres.

Actualización de datos de equipos:

Comparamos y actualizamos el DataFrame principal de equipos con la información de ciudades y pabellones obtenida.

Configuración de credenciales de Foursquare:

Definimos las credenciales para acceder a la API de Foursquare.

Obtención de coordenadas geográficas:

Desarrollamos una función para buscar la latitud y longitud de cada pabellón utilizando la API de Foursquare.

Iteramos sobre los equipos para obtener sus coordenadas geográficas correspondientes, guardándolas en un nuevo DataFrame.

Configuración del mapa base:

Iniciamos un mapa base con Folium, centrado en una ubicación general de los Estados Unidos.

Adición de Marcadores para Cada Equipo:

Añadimos marcadores al mapa para cada equipo, utilizando sus coordenadas geográficas y etiquetando con el nombre del equipo.

Guardado del Mapa:

Guardamos el mapa en un archivo HTML (mapa_equipos_nba.html), permitiendo su visualización en un navegador.



#>/<

**HACK
A BOSS**

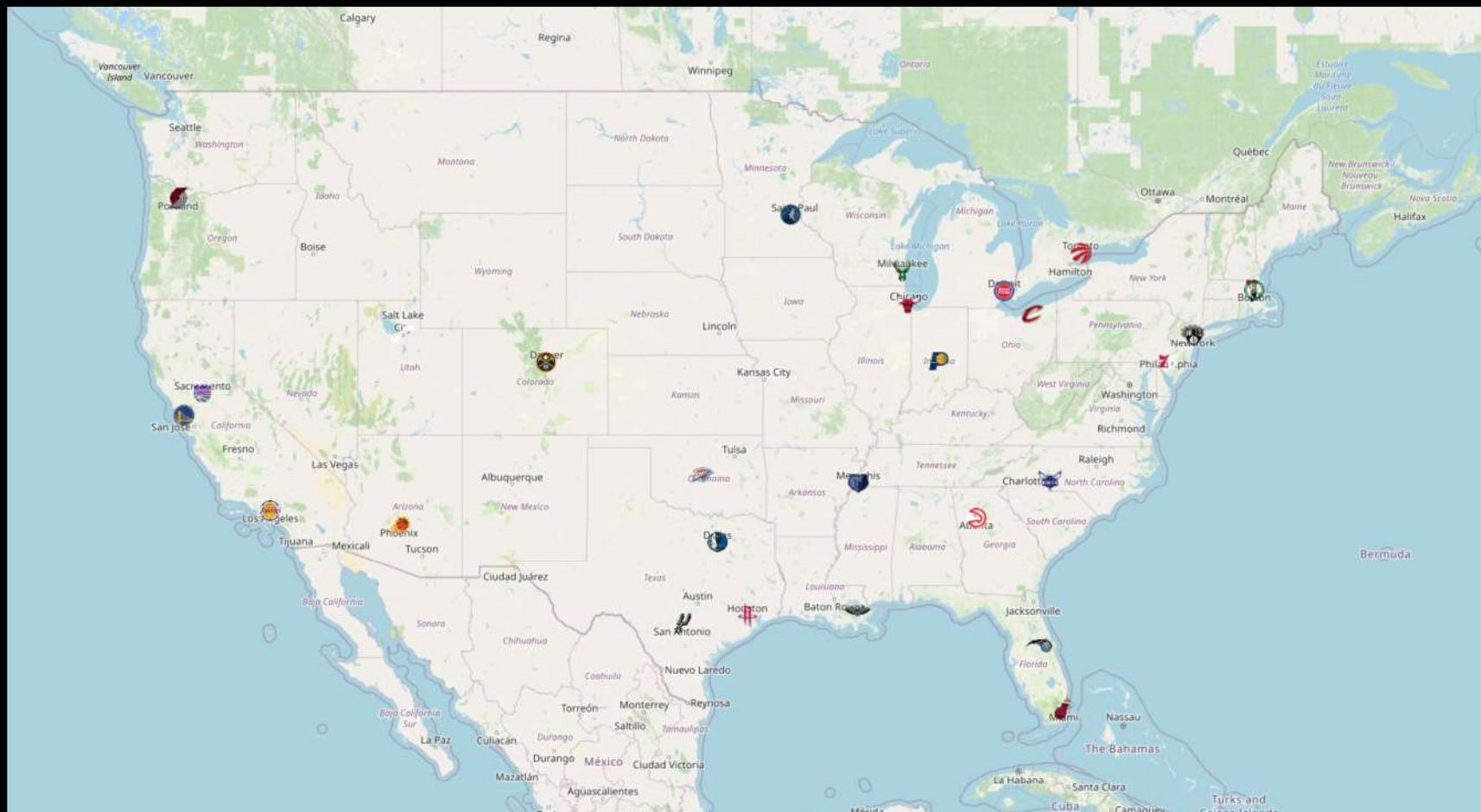
Herramientas y tecnologías usadas

Herramientas de visualización (matplotlib, seaborn, plotly, folium).

Visualización geográfica de equipos de la NBA con Folium con los escudos de los equipos que ya nos habíamos descargado con BS4

Folium con marcadores con logos personalizados:

Itero sobre cada fila del DataFrame que contiene los datos de los equipos, incluyendo sus coordenadas geográficas y el ID del equipo. Utilizo la clase CustomIcon de Folium para crear un ícono personalizado para cada marcador, utilizando el logo correspondiente del equipo. Configuro el tamaño del ícono y establezco la ruta al archivo PNG del logo de cada equipo. Añado marcadores al mapa con estos iconos personalizados, etiquetando cada marcador con el nombre del equipo.



Herramientas y tecnologías usadas

Herramientas de visualización (matplotlib, seaborn, plotly, folium).

Visualización de los Top 10 de jugadores en diferentes categorías de estadísticas de juego

Preparación de mis datos

Unificación de nombres de jugadores:

Primero, combiné las columnas 'Nombre' y 'Apellido' en mi DataFrame para formar una nueva columna 'Nombre Completo'. Esto me proporcionó una identificación clara de cada jugador en los gráficos.

Elección de categorías estadísticas:

Elegí una variedad de categorías estadísticas importantes como puntos (PTS), triples (3PM), tiros libres (FTM), y varias categorías de rebotes y asistencias, entre otras.

Elaboración de gráficos de barras para cada categoría:

Utilicé un bucle para recorrer cada categoría estadística.

Para cada una, seleccioné los 10 mejores jugadores con la función nlargest de pandas y creé un gráfico de barras utilizando Seaborn.

En los gráficos, mostré el nombre completo del jugador en el eje Y y su rendimiento estadístico en el eje X.

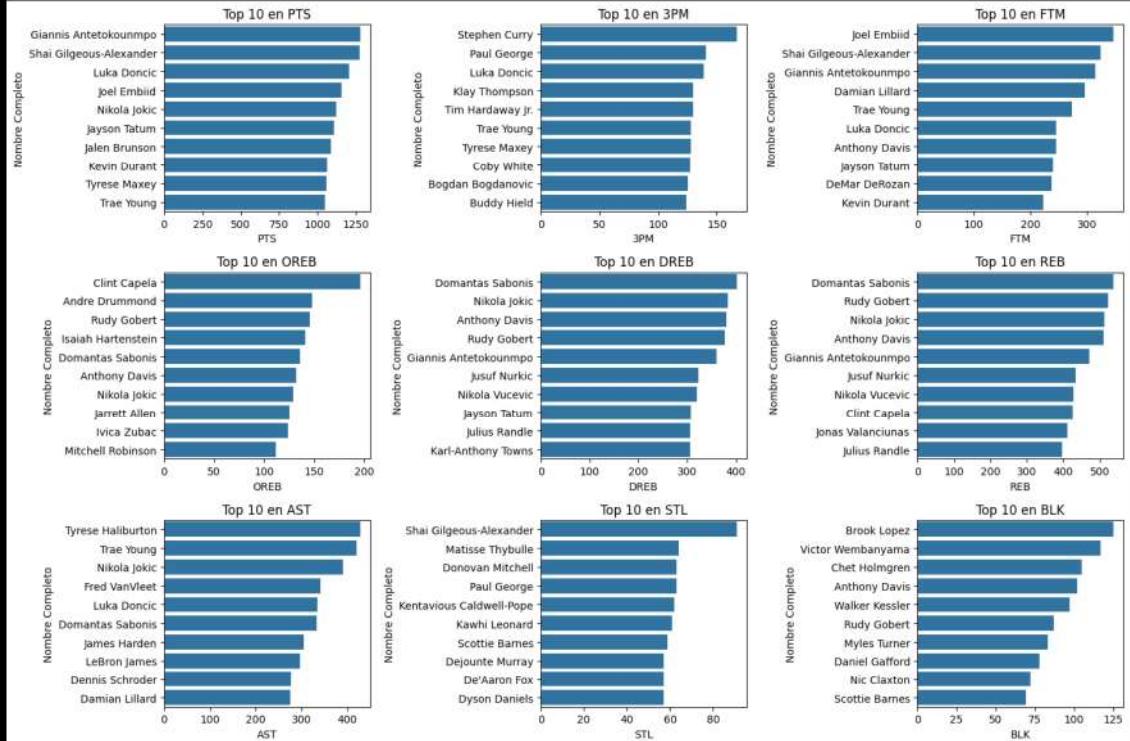
Asigné un título específico a cada gráfico para indicar claramente la categoría representada.

Organización y presentación de los gráficos:

Coloque los gráficos en una disposición de 3x3 con plt.subplot para una presentación organizada.

Aplique plt.tight_layout para ajustar el diseño y garantizar que todo se viera claro y ordenado.

Finalmente, mostré todos los gráficos juntos con plt.show()



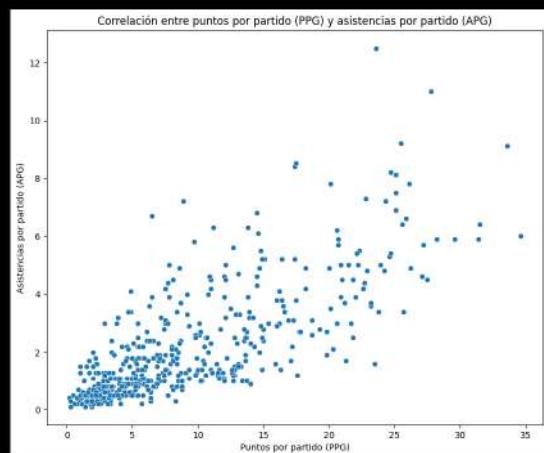
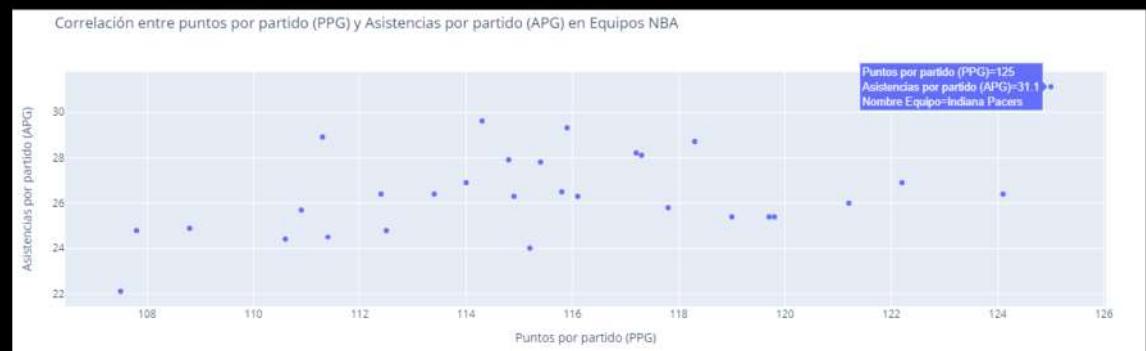
Herramientas y tecnologías usadas

Herramientas de visualización (matplotlib, seaborn, plotly, folium).

Gráficos de correlación de jugadores y equipos respecto a puntos por partido y asistencia por partidos.

He realizado un análisis de correlación entre dos importantes estadísticas del baloncesto: puntos por partido (PPG) y asistencias por partido (APG) tanto para jugadores individuales como para equipos de la NBA.

Los gráficos de dispersión las he creado con Matplotlib y Plotly me permiten visualizar la relación entre estas dos variables.



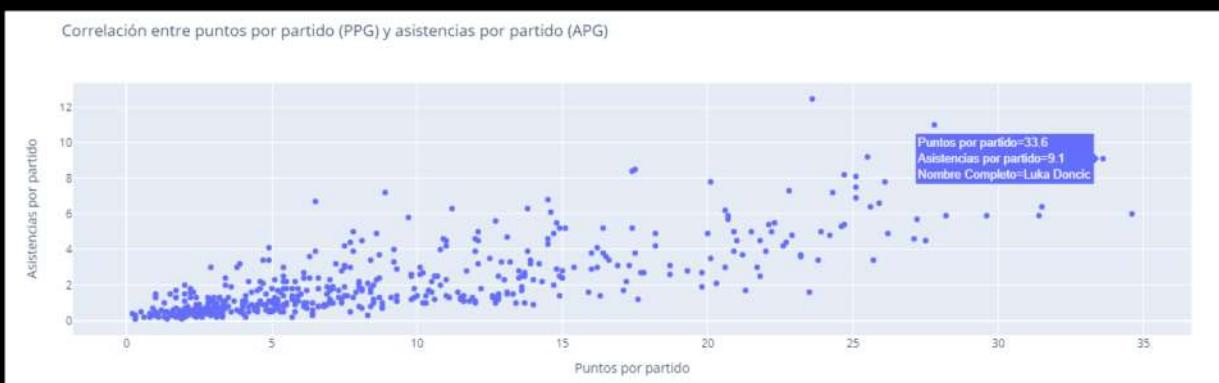
Distribución del gráfico:

El gráfico muestra cómo se distribuyen tanto jugadores como equipos en términos de su capacidad para anotar (PPG) y asistir (APG). Los jugadores y equipos ubicados más a la derecha (mayor PPG) son los principales anotadores del juego, mientras que aquellos más arriba (mayor APG) son destacados en asistencias.

Como observamos hay una tendencia o patrón en el gráfico, como una línea diagonal ascendente, indicando que los jugadores que anotan más puntos también tienden a realizar más asistencias.

Equipos con altos PPG y APG podrían indicar un estilo de juego ofensivo y colaborativo. Podrían estar implementando un juego basado en un juego más dinámico del balón y el trabajo en equipo, lo que les permite tanto anotar como asistir eficientemente.

Los equipos con alta puntuación pero asistencias bajas podrían depender más de jugadores individuales para anotar.



Una correlación positiva fuerte entre PPG y APG en jugadores podría indicar jugadores versátiles que contribuyen significativamente a su equipo en múltiples frentes.

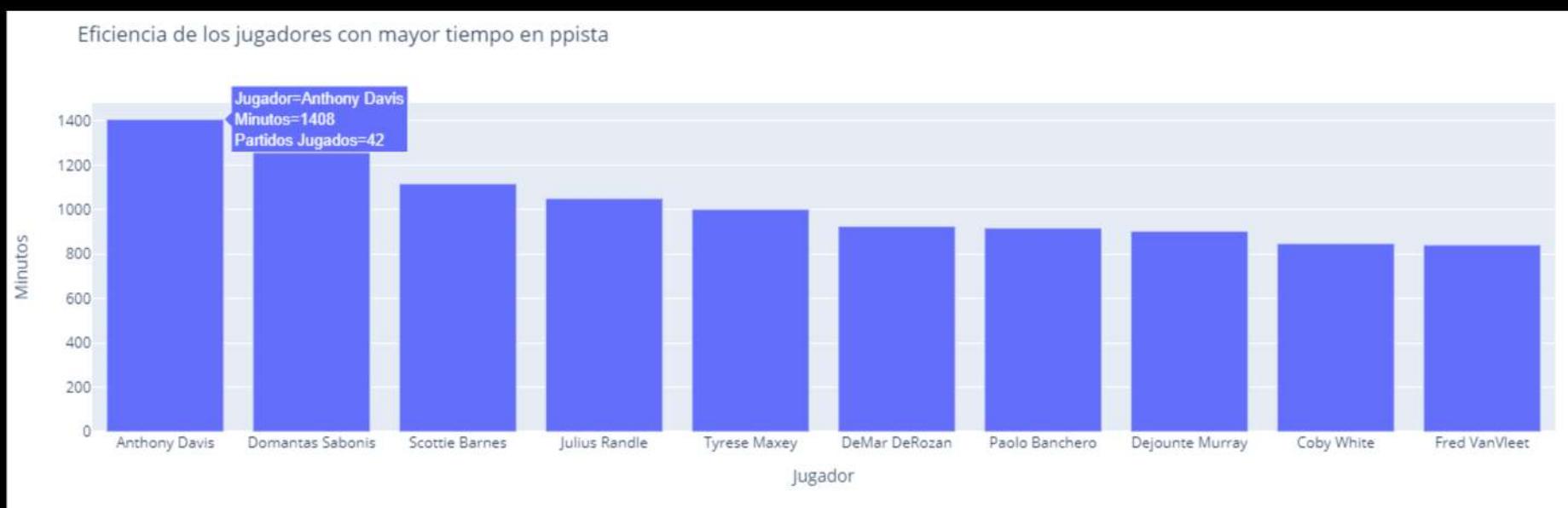
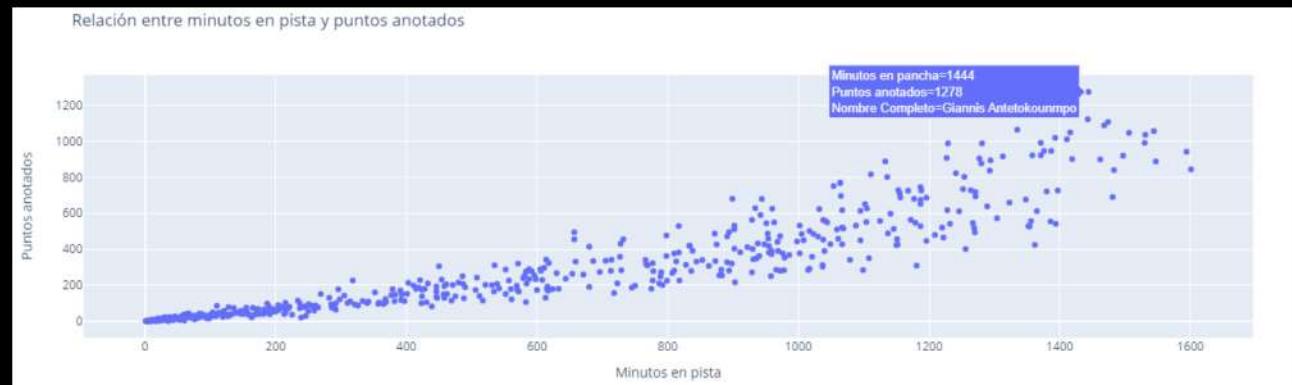
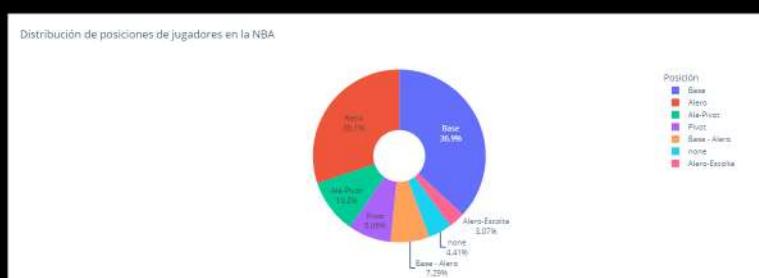
En los equipos, una buena correlación entre PPG y APG podría ser indicativa de un juego de equipo eficiente, donde los jugadores no solo anotan sino que también crean oportunidades de anotación para otros.



Herramientas y tecnologías usadas

Herramientas de visualización (matplotlib, seaborn, plotly, folium).

Otros gráficos realizados



Y entonces... se me ocurrió hacer esto



La idea

Como tenía los datos estadísticos de los jugadores decidí otorgarles valores numéricos del 1 al 100 para ver su destreza.

La idea era hacer algo de gamificación, pero de forma rápida. He pretendido distribuir los puntos en base a los que están participando, es decir, el mejor anotando en puntos tiene un 10. Desde ese dato he distribuido los puntos entre todos los jugadores hasta llegar al peor.

Así con todas las estadísticas.

Además he querido aplicar la distribución de los puntos por los quartiles, es decir, los que están por encima del q3 en puntos tienen una mayor puntuación, no se han distribuido de forma normalizada utilizando la distancia de la mediana para hacerlo.

Quizás sea un poco farragoso de explicar, pero tiene su lógica en la distribución de los puntos.

Y el otro desafío que me propuse, porque lo lógico hubiese sido hacerlo con Photoshop, era el diseño de la tarjeta. Estuve investigando y vi que podía hacerlo con Python y me lo tomé como un reto llevarlo a cabo.

Para el futuro no descarto hacer diseño para cada uno de los jugadores mezclando Photoshop y Python para la automatización de los datos. Me pareció muy interesante este desafío.



#>/<

**HACK
A BOSS**

Proyecto puntuación / tarjetas NBA

Desarrollo de un sistema de puntuación para jugadores de la NBA

Preparación de datos y cálculo de estadísticas:

Importación de datos:

Importé datos de jugadores y equipos de la NBA desde archivos Excel que previamente había guardado.

Cálculo de estadísticas normalizadas:

Creé una copia del DataFrame de jugadores para trabajar sin modificar los datos originales.

Calculé las medias por partido jugado (GP) para una variedad de estadísticas como minutos, puntos, rebotes, asistencias, etc.

Eliminé a los jugadores que no habían jugado ningún partido para evitar divisiones por cero.

Normalización de estadísticas:

Normalicé las estadísticas calculadas para escalarlas en un rango de 0 a 10, utilizando el mínimo y máximo de cada estadística.

Ajuste por cuartiles:

Ajusté las puntuaciones normalizadas basándome en los cuartiles.

Incrementé las puntuaciones de aquellos jugadores cuyas estadísticas estaban en el cuartil superior (Q3) y disminuí las de los que estaban en el cuartil inferior (Q1).

Este paso estaba destinado a valorar más a los jugadores que superaban ciertos umbrales estadísticos.

Ajuste por varianza:

Realicé un ajuste adicional basado en la varianza de cada estadística.

Modifiqué las puntuaciones para aquellos jugadores cuyas estadísticas variaban significativamente de la media.

Este paso me ayudó a identificar y ajustar las puntuaciones de jugadores con rendimientos estadísticos particularmente inconsistentes.

Cálculo de la puntuación total:

Media de puntuaciones:

Calculé la media de todas las puntuaciones normalizadas y ajustadas para cada jugador, obteniendo así una puntuación total que refleja su rendimiento general en todas las estadísticas consideradas.

Escalado a formato tipo FIFA:

Escalé la puntuación total a un formato de 1 a 100, similar a las valoraciones en el videojuego FIFA, para facilitar la comparación y la interpretación.

Visualización de los mejores jugadores:

Finalmente, ordené y presenté los jugadores según su puntuación total, permitiéndome ver quiénes eran los jugadores mejor valorados según mi sistema de puntuación.

ID	NOMBRE	APellido	Puntuación Total
402	Shai	Gilgeous-Alexander	9.509625
418	Giannis	Antetokounmpo	9.508024
241	Kevin	Durant	9.057219
***	***	***	***
286	Alex	Fudge	8.500000
307	Spurs	Loyd	8.500000
289	JaVale	Garcia	8.500000
292	Jamaree	Bouyea	8.500000
528	E.J.	Liddell	8.500000

[521 rows x 3 columns]

```
# Ahora vamos a calcular la media de puntos que han obtenido y ver cuál es el mejor jugador según nuestros cálculos de la NBA
columnas_puntuaciones = [stat + '_por_GP_normalizado' for stat in estadísticas]
df_puntuaciones_finales['Puntuación_Total'] = df_puntuaciones_finales[columnas_puntuaciones].mean(axis=1)

# Y quiero expresar el resultado en formato tipo FIFA que es valoración de 1 a 100
df_puntuaciones_finales['Puntuación_Total'] = (df_puntuaciones_finales['Puntuación_Total'] * 10).round(0)

print(df_puntuaciones_finales[['Número', 'Apellido', 'Puntuación_Total']].sort_values(by='Puntuación_Total', ascending=False))
```

ID	NOMBRE	APellido	Puntuación Total
379	Joel	Embiid	83.98
106	Iuka	Doncic	83.28
418	Giannis	Antetokounmpo	78.91
27	Jayson	Tatum	74.62
218	Nikola	Jokić	74.26
***	***	***	***
57	Dayron	Skapetonov	5.00
75	Grundberg	Ritme	5.00
262	Tay	Huff	5.00
399	Malcolm	Gazalon	5.00
***	***	***	***



Proyecto puntuación / tarjetas NBA

Desarrollo de un sistema de puntuación para jugadores de la NBA

Esta idea me permite crear una especie de sistema de clasificación para catalogar a los jugadores de la NBA, combinando varias estadísticas clave en una sola puntuación. Este sistema no solo me da una visión general del rendimiento de los jugadores, sino que también me permite identificar a los jugadores más destacados en distintas áreas del juego.

Puedo ver quienes son los mejores ofensivamente, defensivamente, etc...

Asignación de valores monetarios ficticios a jugadores de la NBA

Otra de las cosas que se me ha ocurrido agregarles a los jugadores de la NBA es un valor económico. Es ficticio. Muchas veces somos más conscientes de las cosas cuando le ponemos un valor, entonces, utilizando ese principio se me ha ocurrido ponerles un valor económico.

Establecimiento de valores máximos y mínimos:

Definí un valor máximo de 300 millones de dólares y un mínimo de 5 millones, con una puntuación máxima de 100 para escalar los valores monetarios. No podía tener un jugador con valor infinito, había que poner un máximo.

Escalar valor monetario:

Desarrollé una función `escalar_valor_monetario` para asignar a cada jugador un valor monetario basado en su puntuación total, utilizando una escala lineal entre los valores mínimo y máximo.

Ajustes por cuartiles y varianza igual que había hecho con la distribución de puntos estadísticos a los jugadores:

Realicé ajustes adicionales en los valores monetarios basándome en los cuartiles y la varianza de las puntuaciones totales.

Incrementé el valor para los jugadores en el cuartil superior y lo disminuí para aquellos en el cuartil inferior.

También ajusté los valores en función de la varianza para reflejar la consistencia en el rendimiento del jugador.

Aplicación de la función de ajuste:

Aplicué la función `ajustar_valor_monetario` para obtener los valores monetarios finales para cada jugador.

Visualización de los valores monetarios:

Finalmente, ordené y presenté los jugadores según su valor monetario ajustado, ofreciendo una visión de su "valor" en un contexto ficticio basado en su rendimiento.

Significado y uso de este enfoque:

Este enfoque me permite visualizar el valor de los jugadores de una manera novedosa y entretenida, similar a cómo los videojuegos clasifican y valoran a los jugadores.

Aunque los valores son ficticios, proporcionan una perspectiva interesante sobre cómo el rendimiento estadístico podría traducirse en valor monetario en un contexto hipotético.

	Nombre	Apellido	Puntuacion_Total	Valor_Monetario
379	Joel	Embiid	83.90	257.5050
166	Luka	Doncic	83.28	255.6760
418	Giannis	Antetokounmpo	78.91	242.7845
27	Jayson	Tatum	74.62	230.1290
218	Nikola	Jokic	74.26	229.0670
..
236	Chris	Livingston	6.98	5.0000
239	Aleksej	Pokusevski	8.30	5.0000
451	Jacob	Toppin	6.19	5.0000
253	Damian	Jones	8.31	5.0000
520	E.J.	Liddell	8.25	5.0000

[521 rows x 4 columns]



#>/<

HACK
A BOSS

Proyecto puntuación / tarjetas NBA

Desarrollo de un sistema de puntuación para jugadores de la NBA

Con los datos de puntuaciones y valor monetario de los jugadores de la NBA ya calculados, decidí crear gráficos de radar para visualizar las habilidades ofensivas de los tres mejores jugadores. Estos gráficos me resultan útiles para mostrar el rendimiento de los jugadores en múltiples categorías en un solo vistazo, o al menos estoy tan acostumbrado a verlos en los video juegos deportivos que me parecen muy entendibles. Aquí está cómo lo hice:

Elección de categorías estadísticas:

Selecioné categorías clave como porcentajes de tiros de campo (FG%), triples (3P%) y tiros libres (FT%), rebotes, asistencias, robos, bloqueos, pérdidas y faltas personales.

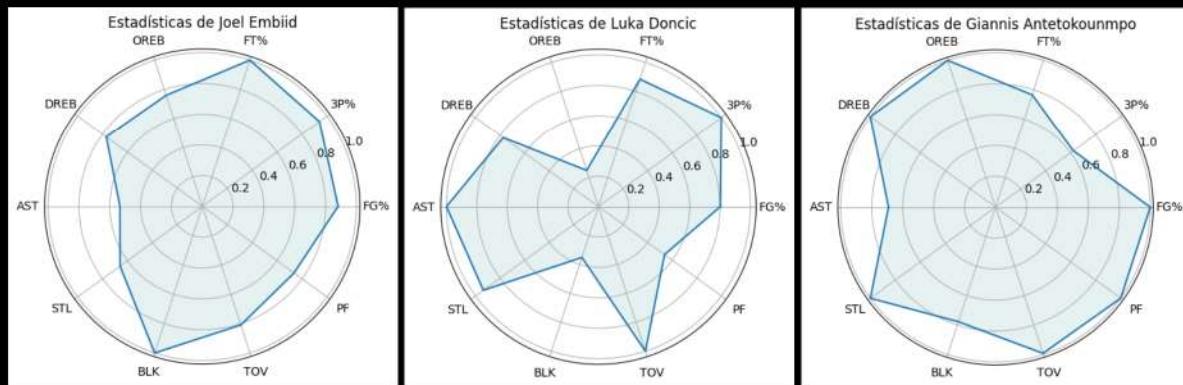
Normalización de estadísticas:

Normalicé las estadísticas para cada jugador en relación con el valor máximo de cada categoría entre los tres mejores jugadores, permitiendo una comparación justa.

Configuración y creación del gráfico:

Configuré un gráfico de radar para cada uno de los tres mejores jugadores. Utilicé un bucle para graficar los valores normalizados de cada jugador en las categorías seleccionadas.

Añadí estilos y colores para mejorar la legibilidad y estética del gráfico.



Presentación de los resultados:

Cada gráfico muestra visualmente cómo cada jugador se compara en las diferentes categorías estadísticas.

Los títulos de los gráficos incluyen el nombre del jugador, lo que facilita la identificación.

Uso de Plotly para gráficos interactivos:

Repetí el proceso de normalización y creación de gráficos de radar utilizando Plotly.



Proyecto puntuación / tarjetas NBA

Desarrollo de un sistema de puntuación para jugadores de la NBA

En la última parte de mi proyecto, realicé un diseño gráfico para crear fichas tipo FIFA para los jugadores de la NBA, integrando los datos y gráficos que había elaborado anteriormente. Esto es lo que hice:

Creación de la carpeta para guardar las fichas:

Primero, creé una carpeta para guardar las fichas de los jugadores si aún no existía.

Selección de los 10 mejores jugadores:

Luego, seleccioné los 10 mejores jugadores basándome en la 'Puntuación _Total' que había calculado previamente.

Normalización y creación de gráficos:

Normalicé las estadísticas para cada jugador y creé un gráfico de radar para cada uno utilizando Plotly, representando sus habilidades en diferentes categorías.

Guardé cada gráfico de radar como un archivo PNG con fondo transparente para usarlos posteriormente en las fichas.

Diseño de las fichas con PIL (Python Imaging Library):

Configuración de fuentes y textos:

Configuré las fuentes y tamaños para el texto que aparecería en las fichas. Preparé los textos para cada categoría estadística y otros detalles del jugador como el nombre, edad, país y dorsal.

Integración de elementos gráficos y texto:

Comencé con una imagen de fondo que serviría como la base de la ficha. Colocqué los gráficos de radar y los logos de los equipos en sus posiciones correspondientes en la ficha. Utilicé PIL para dibujar el texto con las estadísticas, la puntuación total y el valor monetario de cada jugador en la ficha.

Guardado de las fichas finales:

Una vez completada la integración de todos los elementos, guardé cada ficha como un archivo PNG.



Utilicé este diseño rápido realizado en Photoshop para crear la ficha de los jugadores de la NBA

Para el futuro se podrían implementar en el diseño imágenes de montajes gráficos de los jugadores con su foto, además de otros elementos gráficos para hacerla más atractiva.

Pero lo que quería era ver la capacidad de diseño que podía hacer con Python y la librería PIL que no la conocía.



#>/<

**HACK
A BOSS**

Top 10 de jugadores en tarjetas NBA



76

Joel Embiid

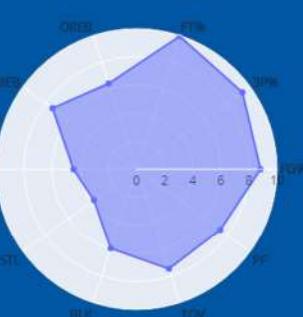
Edad: 29 / Dorsal: 21
País: Cameroon

Puntos: 10.0 Asistencias: 6.3
Triples: 4.0 Robos: 6.2
Tiros Libres: 10.0 Bloqueos: 7.6
Rebotes Of.: 6.6 Pérdidas: 10.0
Rebotes Def.: 10.0 Faltas: 9.2

Puntos Totales: 83.9

Precio del Jugador: \$257.50M

Puntos Ataque / Defensa



The radar chart displays Embiid's performance across seven categories: Pts (10.0), 3P (4.0), FT (10.0), REB (6.6), STL (7.6), BLK (6.2), and TOV (9.2). The chart is set against a scale from 0 to 10.

Información sobre las tarjetas NBA:

Todos los datos de la tarjeta están realizados con Python.

A todas las tarjetas se le ha agregado el escudo del equipo del jugador.

Nombre y apellidos.

Edad y dorsal.

Así como su país de nacimiento.

Además se le ha agregado el gráfico de radar con los puntos NBA otorgados con nuestros cálculos estadísticos.

Y en la parte media/baja de la tarjeta se han introducido esos puntos.

La puntuación de este jugador en puntos por partido es de 10.

En Triples su puntuación es 4.

En tiros libres es un 10.

En rebotes ofensivos 6.6

En rebotes defensivos 10

En asistencias 6.3

En robos 6.2

En bloqueos tiene una puntuación de 7.6

En pérdidas tiene un 10

Y en faltas tiene un 9.2

Con ello, este jugador tiene una puntuación de 83.9 siendo el jugador más valioso de los calculados.

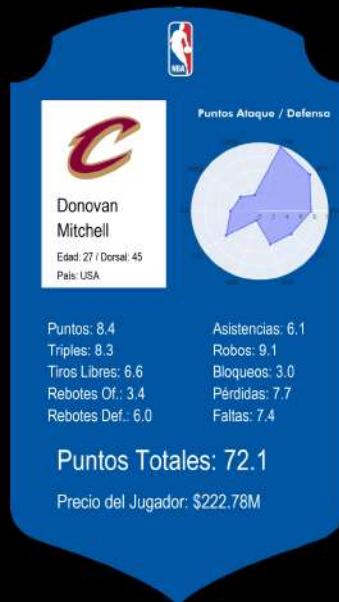
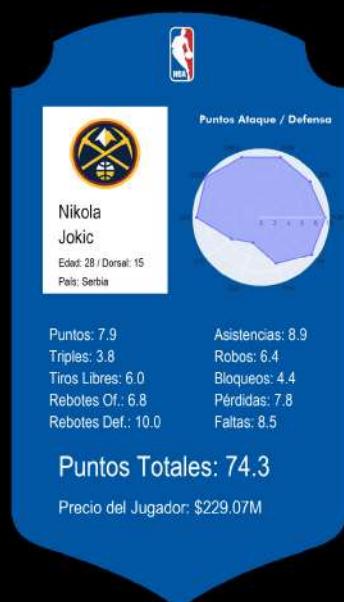
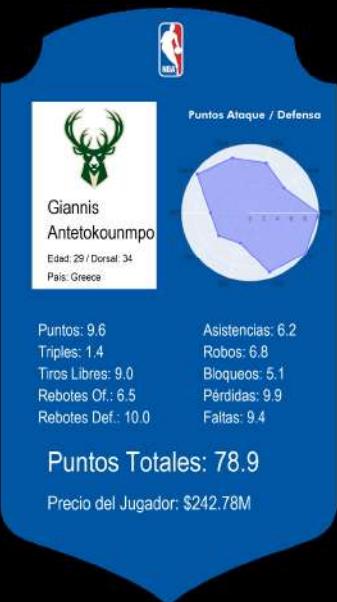
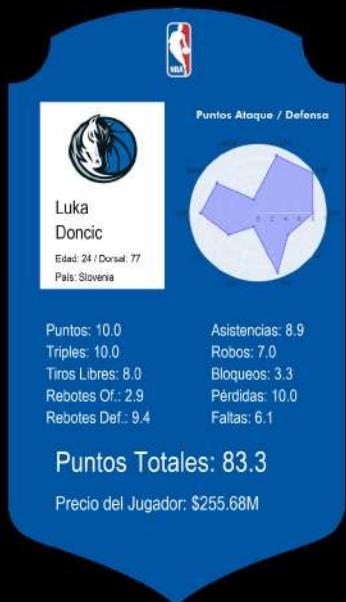
Y su valor económico, ficticio, es de 257, 50 millones de dólares.



#>/<

**HACK
A BOSS**

Top 10 de jugadores en tarjetas NBA



>/<

**HACK
A BOSS**

#>/<#

HACK A BOSS

Proyecto 1
#dsb07rt
enero 2024

Bootcamp Data Science & AI