****

**Universidad Autónoma de Nuevo León**

**Facultad de Ciencias Físico Matemáticas**

**(FCFM)**

**2do Semestre**

**Programación Básica**

**Docente: Perla Marlene Viera Gonzalez**

# PIA E2

**Equipo 8**

|  |  |
| --- | --- |
| **Nombre** | **Matricula** |
| Ángel Ernesto García Ramírez | 2049907 |
| Aldo Manuel Gómez Rendon | 1996008 |

# Aula: 403

# Grupo: 073

**A entregar el 7 de Mayo del 2024 en Ciudad Universitaria, San Nicolas, Nuevo Leon**

**Documentación.**

**Extracción y limpieza de los datos.**

Para extraer los datos hicimos el llamado a la api que fuera necesaria según el caso:

1. Para los datos del Barcelona se uso :

https://api.football-data.org/v4/teams/81/matches?competitions=PD

1. Para los datos del Real Madrid se usó:

<https://api.football-data.org/v4/teams/86/matches?competitions=PD>

A ambas se les aplico el filtro ?competitions=PD para que la información de los partidos que nos mostrara la api fuera únicamente de los partidos que se han jugado en la Primera División Española.

Esta URL nos devuelve un diccionario con tres llaves, las cuales son:

1. filters: Contiene filtros que se utilizaron para la búsqueda, algunos predefinidos y otros indicados por nosotros como competitions.
2. resultSet: Un pequeño resumen de los partidos, la cantidad, la competencia, el primer partido y cuando será el último, los que ya se han jugado y los resultados, aunque los resultados están mal contados por lo que es necesario hacer el conteo por nuestra parte.
3. matches: contiene una lista con varios diccionarios que contienen información relacionada al partido, lo que más nos importa es que en esta llave tenemos los partidos y los resultados.

La llave resultSet nos serviría para nuestro problema planteado, sin embargo, el conteo de victorias, empates y derrotas esta mal hecho por lo que hay que hacerlo nosotros.

Antes de comenzar a explicar el como se realizo el conteo hay que especificar que se definen 4 variables en 0, las cuales son:

1. Vict{Siglas del equipo}: Contendrá las victorias del equipo
2. Derr{Siglas del equipo}: Contendrá las derrotas del equipo
3. Emp{Siglas del equipo}: Contendrá los empates del equipo.
4. SinJug{Siglas del equipo}: Contendrá los partidos sin jugar del equipo.

Únicamente nos interesa la llave matches, por lo que, iteramos con x,y en el JSON que nos devuelve la URL y colocamos un if cuando la llave sea igual a “matches”.

Como el value de la llave matches es una lista de diccionarios, hay que iterar entre cada elemento de esta lista con un for con ***i*** para asi obtener el resultado que buscamos.

Los resultados que buscamos obtener son los resultados de los partidos, estos están en un diccionario que contiene una llave llamada “score”, y el value es otro diccionario con diversas llaves, pero la que nos interesa es la llave “winner” que nos indica el ganador, por lo que guardamos al ganador en una variable de la sig. manera:

Gan = ResBC["matches"][i]["score"]["winner"](En este caso es para los partidos del Barcelona por eso el BC después de RES)

Sin embargo, nos indican únicamente si gano el equipo local, el visitante, se empato o no se ha jugado el partido, por lo que hay que conocer a los equipos locales y visitantes de cada partidos.

La información sobre quien es el equipo local y quien el equipo visitante de cada partido esta en la llave “homeTeam” y “awayTeam” respectivamente, estas llaves contienen otro diccionario con diversas llaves, pero la llave que nos importa es “name”, por lo que, para conocer los equipos visitantes se realizó lo siguiente:

Casa = ResBC["matches"][i]["homeTeam"]["name"]

Visit = ResBC["matches"][i]["awayTeam"]["name"] (En este caso es para los partidos del Barcelona por eso el BC después de RES)

Ya una vez conociendo a los equipos locales y visitantes, asi como el ganador podemos iniciar el conteo de los resultados.

Esto lo hicimos con manejo de excepciones y utilizando el módulo re, creando un objeto con cada valor que podría tomar la variable Gan que contendría a quien gano el partido, si se empato o si no se ha jugado, esto nos lo indica tomando la variable alguno de los siguientes valores

1. AWAY\_TEAM: Gano el visitante.
2. HOME\_TEAM: Gano el local.
3. DRAW: Se empato el partido.
4. None: No se ha jugado el partido

Fuimos utilizando manejo de excepciones con cada caso de los objetos que se crearían, en caso de que una sección del try correspondiente a uno de los casos hubiera un error, se pasaría a la sección del except que contendría al otro caso. Si no se ha jugado un partido no es necesario crear un objeto que tome un valor, ya que, únicamente se llegaría a la sección del código correspondiente a los partidos no jugados, si no se cumplen ninguna de las otras tres secciones del código correspondientes a los otros dos casos.

Con DRAW y None no hay que realizar comprobaciones sobre para que equipo aplica ese resultado, ya que si empatan, empatan los dos, y si no han jugado el partido no lo han jugado los dos. Por lo que únicamente se le suma 1 al contador que contiene la información de los partidos empatados y no jugados, una vez que hayamos comprobado si se empato o no se ha jugado.

Iniciamos un if se ejecutará si el objeto tiene el valor de AWAY\_TEAM, esto nos indicara que gano el equipo visitante, por lo que dentro de este if, anidamos otro if en que comprobaremos si el Barcelona es el equipo visitante, si esto es así se le sumara uno a las victorias. Si el equipo visitante es cualquier otro valor menos el Barcelona, se le sumara uno a las derrotas.

Replicamos la misma lógica en otro if el cual se ejecutará si el objeto tiene el valor de HOME\_TEAM, como aquí se indica que gano el equipo local, comprobamos si el Barcelona es el equipo local o no y se realiza la suma correspondiente al contador correspondiente.

Ya que ya conocemos la cantidad de partidos ganados, empatados, perdidos y sin jugar de los equipos, (en el caso de esta explicación únicamente del Barcelona), calculamos los porcentajes que representan estas cantidades sobre el total de los partidos jugados, esto dividiendo la cantidad de cada uno de los resultados posibles, (victoria, empate, derrota, sin jugar), entre el total de partidos jugados. Este porcentaje lo almacenamos en una variable.

Por último imprimimos la información obtenida, la cantidad de partidos, ya sea ganados, empatados, perdidos o sin jugar y el porcentaje que representan cada uno de estos sobre el total de los partidos jugados por el equipo.

Para esta explicación se uso como ejemplo para la explicación a el equipo del Barcelona, pero este proceso se repito de la misma manera con el Real Madrid por lo que de esta manera ya tenemos exactamente la información que ocupamos de cada uno de los dos equipos.

**Diseño de la Nueva Estructura de Datos**

Con el objetivo de organizar la información de manera más accesible y estructurada, se diseñó una nueva arquitectura de datos basada en listas y diccionarios propios del lenguaje Python. Esta estructura permite almacenar de forma clara tanto los detalles de cada partido como los resultados globales del equipo analizado, facilitando así el procesamiento y análisis posterior.

**Estructura de Datos**

Se definieron dos niveles de almacenamiento:

* Estructura global (partidos\_analizados)

Se implementó una lista donde se agregan todos los partidos procesados individualmente. Esta lista actúa como un contenedor centralizado de datos, que puede ser reutilizado para análisis más profundos, visualización o exportación.

* Estructura individual por partido

Cada partido se almacena como un diccionario que registra los siguientes campos relevantes:

[fecha: la fecha del encuentro (en formato ISO 8601),

local: nombre del equipo local,

visitante: nombre del equipo visitante,

resultado: resultado del partido desde la perspectiva del equipo analizado (Ganado, Perdido, Empatado o Sin jugar),

equipo: nombre del equipo que se está evaluando.]

Esta estructura permite identificar con claridad el contexto de cada resultado y a qué equipo corresponde.

**Resumen de resultados (resultados)**

Además del registro individual de partidos, se genera un diccionario resumen que acumula las siguientes métricas por equipo:

Número total de victorias,

Número total de derrotas,

Número total de empates,

Número de partidos aún no jugados.

Este resumen es devuelto por la función principal de análisis para un acceso directo a los datos estadísticos agregados del equipo.

**Lógica de Almacenamiento**

Durante el análisis, por cada partido del equipo evaluado, se determina el resultado y se registra en la lista partidos\_analizados junto con los datos clave del evento. Junto con eso el diccionario de resumen se va actualizando con base en el resultado detectado (victoria, derrota, empate o partido pendiente).

**Bibliografía de código**

Para conocer el uso del guardado en .csv nos apoyamos con más información aparte de lo visto en clase las fuentes, son las siguientes:

<https://www.youtube.com/watch?v=iXv4bBG3qKc>