



Universidad Autónoma de Nuevo León
Facultad de Ciencias Físico Matemáticas
(FCFM)

2do Semestre

Programación Básica

Docente: Perla Marlene Viera Gonzalez

PIA E4

Equipo 8

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

Aula: 403

Grupo: 073

**A entregar el 19 de Mayo del 2024 en Ciudad Universitaria,
San Nicolas, Nuevo Leon**

Documentación.

Desarrollo de las gráficas.

Para la elaboración de las gráficas se realizó en su totalidad con la librería matplotlib.pyplot

- Grafica 1

Primeramente se realizó una gráfica de barras que indicaran la cantidad de victorias, derrotas y empates de cada equipo, estas barras estarían en una sola marca del eje x para ambos equipos, una barra por encima de la otra para que se pueda apreciar mejor sus tamaños.

Para ello primero definimos una lista llamada label, que contendrá los textos para las 3 cosas que se indicaran en el eje x, (victorias, derrotas y empates), después un diccionario llamado Resultados que contiene como llaves los nombres de los equipos y como valor una lista con las cantidades de partidos ganados, perdidos y empates, (en ese orden).

Definimos una figura, (lienzo), y un espacio para gráfica con matplotlib.pyplot y una variable llamada piso igualada a 0, esto es para que cada que inicia una nueva barra en la gráfica se inicie desde abajo.

Para realizar la gráfica iniciamos con un contador $i = 0$, y una lista de colores, (rojo y blanco), para las barras de la gráfica.

Iteramos con la variable equipo y Resultado en el items del diccionario Resultados, después guardamos en una variable llamada gráfica la grafica creada con .bar(), esta gráfica creada con .bar() tendrá en el eje x lo indicado por la lista llamada label, su eje y serán los valores de la llave del diccionario correspondiente, tendrá como piso la variable piso igualada a 0 y los colores serán la lista y el contador. Al piso se le suman los elementos del diccionario, esto para que la siguiente barra puesta en la gráfica este justo por encima de la anteriormente creada, se le suma uno al contador y se indica con un .bar_label() que queremos que nos muestre el valor número que representa la barra de la gráfica creada. Este proceso se repite para el otro elemento del .items del diccionario.

Después de esto definimos el titulo de la gráfica, le ponemos de color de fondo un amarillo claro ya que ambos equipos llevan amarillo en sus escudos. Creamos una leyenda que tenga el mismo fondo amarillo claro y bordes negros que nos indique que equipo representa el color de las barras.

Por último guardamos la gráfica en el directorio donde se esté ejecutando el script con el nombre "Graficas de barras sobrepuestas de los resultados de los equipos", esto en formato .jpg. Y mostramos la gráfica.

- Gráfica 2

La segunda gráfica para crear fueron dos gráficas de líneas que nos indique como se fueron dando los resultados de ambos equipos jornada a jornada.

Para ello definimos una figura con dos filas y una columna, así como dos espacios para gráfica.

Primeramente desarrollamos la primera lista con un `.plot()`, los parámetros que recibe esta lista son:

1. Para el eje x se le pasaran los números del 0 al `len(ListBC)` de 1 en 1, (`ListBC` contiene los resultados de las jornadas en orden, 1 indica victoria, 2 derrota y 3 empate).
2. Para el eje y se le pasara esta `ListBC`.
3. Se indica un marcador `^` para cada punto de la gráfica.
4. Un ancho de línea de 0.2.
5. Un color del marcador en rojo.
6. Un color de la línea en azul.

Después, con la ayuda de la documentación de `matplotlib` desarrollamos una leyenda que contenga una explicación de que 1 representa victoria, 2 representa derrota y 3 representa empate, así mismo que el fondo sea amarillo claro y sus bordes negros.

Indicamos que la gráfica 1 tenga un fondo amarillo claro. Así como indicamos que las marcas en x sean de 2 en 2 hasta llegar a `len(ListBC)` y que las marcas en y sean [1, 2, 3]. Definimos el título de la gráfica como "Resultados del Barcelona y Real Madrid respectivamente". Lo definimos de esta manera ya que si ponemos un título individual para la gráfica 2 se empalmaría con la gráfica 1. Y por último definimos un label en y que indique que la gráfica es correspondiente al Barcelona.

Ahora, para la elaboración de la segunda gráfica se realizó el mismo proceso solo que con los respectivos cambios:

1. El marcador de los puntos de la gráfica es una estrella y se le pone un `width` de 1.5
2. El color de los marcadores es blanco y el color de las líneas es amarillo
3. El fondo del gráfico es gris.
4. Se le define un label en x que indique que los marcadores del eje x son las jornadas, se pone solo en este gráfico por que si se pone en el primer gráfico se empalmaría con este gráfico.
5. Se le define un label en y que indique que la gráfica es correspondiente al Real Madrid.
6. Cambios en la información de los resultados de los partidos correspondientes al Real Madrid.

Por último guardamos la gráfica en el directorio donde se esté ejecutando el script con el nombre “Grafica líneas y puntos que indica resultados jornada a jornada” en formato .jpg

Mostramos la gráfica

- Gráfica 3

Por último, la tercera gráfica fueron dos gráficas de pastel que contendrá los porcentajes de resultados del Real Madrid y Barcelona.

Primero definimos unas variables que utilizaremos mas adelante en la elaboración de los gráficos:

1. PorBar es una lista que contendrá los porcentajes de las victorias, derrotas y empates, en ese orden, del Barcelona.
2. PorRM es una lista que contendrá los porcentajes de las victorias, derrotas y empates, en ese orden, del Real Madrid.
3. separar es una tupla que contiene tres números (0.1, 0, 0), esto para indicar que solamente el primer slice de la gráfica de barras se separe de los demás.

Con esto definido procedemos a realizar la gráfica.

Definimos una figura con una fila y dos columnas asi como dos espacios para gráficas.

Primeramente definimos el color de fondo de la figura definida anteriormente en negro, esto por que nos ayudara mas adelante con los colores de las slices del gráfico.

Ahora definimos el primer gráfico usando .pie(), los parámetros pasados a este gráfico son los siguientes:

1. Como datos pasamos los porcentajes de los resultados.
2. Como label pasamos los mismos labels que definimos para el gráfico 1.
3. Indicamos autopct=“%1.1f%%”, esto nos hace que se muestre el porcentaje que representa el slice del 100% de la gráfica, nos muestra este número con 1 decimal.
4. textprops al cual le pasamos un diccionario que contiene el color que contendrá el texto en el gráfico asi como weight = “bold” que indica que se mostrará en negritas.
5. Los colores de cada slice del gráfico, los cuales son rojo, azul y el amarillo claro que hemos venido usando para los anteriores gráficos, (#ffffb5).
6. Indicamos la separación del primer slice.
7. El gráfico tendrá una sombra.
8. Y que el gráfico se girara 90 grados de su posición original para una que tenga una mejor posición.

Ya con esto definimos el título de este primer gráfico como “Porcentaje de resultados del Barcelona”, y que se muestre con un color blanco.

Ya desarrollado el primer gráfico, pasamos al desarrollo del segundo, el cual sigue el mismo procedimiento solo que con los siguientes cambios:

1. Los colores de los slice son blanco, naranja y azul, el naranja esta pasado con el siguiente código Hex #e39d3f.
2. Cambios en la información de los porcentajes correspondientes al Real Madrid.

Con el gráfico realizado pasamos a asignarle el título “Porcentaje de resultados del Real Madrid” y que el título se muestre en blanco.

Guardamos la gráfica en el directorio donde estemos ejecutando el script con el nombre “Grafica de Pastel con los porcentajes de los resultados del Barcelona y Real Madrid” en formato .jpg

Y por último mostramos esta gráfica.

Únicamente se realizaron estas tres gráficas debido a que realizar una cuarta únicamente sería redundante y no aportaría absolutamente nada a la resolución de la problemática inicial del problema, con estas tres gráficas es mas que suficiente para realizar la comparación de los resultados del Barcelona y Real Madrid de la actual liga española, como solo se realizaron estos tres gráficos fue que se enfoco en darle un aspecto más trabajado a las gráficas.

Almacenamiento de los datos en Excel

- ***Lectura y apertura de archivos locales:***

El script inicia su ejecución identificando y accediendo a los archivos generados previamente. Estos archivos contienen datos valiosos como el nombre de los equipos, los resultados de los partidos (victoria, empate, derrota), y el estado de cada encuentro (jugado o pendiente).

El script garantiza que la apertura de estos archivos se realice de manera segura, usando mecanismos que previenen errores comunes como rutas incorrectas, archivos vacíos o datos mal formateados. Esto es esencial para asegurar la integridad del proceso y evitar interrupciones durante la ejecución.

- ***Interpretación y transformación de datos:***

Una vez abiertos los archivos, el script toma el contenido y lo transforma desde su forma original (habitualmente texto plano) a estructuras de datos internas más eficientes, como listas, diccionarios u objetos según la lógica del programa.

Esta transformación permite que los datos sean manipulados, filtrados o reorganizados de acuerdo con las necesidades analíticas. Por ejemplo, se pueden agrupar partidos por equipo, calcular estadísticas de victorias y derrotas, o preparar resúmenes de rendimiento.

- ***Validación de datos cargados:***

Parte crucial del avance es la verificación de que todos los datos esperados hayan sido correctamente cargados. El script realiza comprobaciones para asegurarse de que no faltan registros, que los formatos de los datos coinciden con lo esperado y que no hay errores de lectura. Esta etapa es fundamental para evitar problemas en las fases posteriores de análisis.

- ***Optimización del acceso local:***

Al trabajar sobre archivos locales en lugar de depender constantemente de la API externa, el proyecto mejora notablemente en velocidad y eficiencia. Esto significa que se pueden realizar múltiples pruebas, ajustes y análisis sin incurrir en costos adicionales ni en riesgos de conectividad o límites de uso de la API. Además, se evita la repetición innecesaria de procesos costosos de descarga, enfocándose solo en la reutilización y análisis.

- *Preparación para análisis avanzados:*

Este segundo script no solo carga los datos, sino que los organiza de manera que queden listos para análisis más profundos:

El avance asegura que los datos estén correctamente alineados para ser integrados en módulos futuros del proyecto, que podrían incluir visualización gráfica, exportación a otros formatos (Excel, CSV, bases de datos), o conexión con interfaces de usuario.

Referencia del código:

Youtube **Cómo Leer Archivos Excel con Python | Cargar Datos de Excel (.xlsx) a Python | Paso a Paso :**

<https://www.youtube.com/watch?v=n16Jolz5FQs>