
ESTRUCTURAS DE DATOS USADAS EN EL PIA

1. Listas:

- Las listas se utilizan para almacenar datos ordenados, como las listas de partidas, kills, asistencias, muertes, tamaños (en la función ``graficar_estadisticas_partidas`` y ``graficar_estadisticas_victorias``).
- Se accede a los elementos de estas listas para plotear gráficas usando ``plt.plot``.
- Se utilizaron listas para almacenar datos como el resultado de la consulta a la API para obtener detalles de las partidas (``partidas_ids``), nombres de campeones utilizados (``champion_names``), número de kills, muertes y asistencias en cada partida (``kills_lista``, ``muertes_lista``, ``asistencias_lista``), y los tamaños de las secciones de la gráfica de tasa de victorias (``sizes``).

2. Diccionarios:

- El módulo ``Counter`` crea un diccionario que mapea elementos únicos a su frecuencia de aparición. Se utiliza para contar la frecuencia de uso de cada campeón (``conteo_campeones`` en la función ``graficar_campeones_usados``).
- Se utilizaron diccionarios para almacenar la información de los jugadores obtenida de la API, como el ID de la cuenta (``data['accountId']``) y el nivel del jugador (``data['summonerLevel']``).

3. Cadenas de Texto:

- Se utilizan cadenas de texto para construir patrones de búsqueda de archivos (``patron_archivo``) y mensajes impresos en la consola.
- Se utilizó una cadena de texto (``detalles_partidas``) para almacenar los detalles de las partidas y poder guardarlos en un archivo de texto al finalizar la función.

ESTRUCTURAS DE DATOS USADAS EN EL PIA

4. Objetos Datetime:

- Se utiliza el objeto ``datetime`` para obtener la fecha y hora actual, que luego se utiliza para agregar una marca de tiempo a los nombres de archivo y a las consultas API.

5. Manejo de Archivos y Directorios:

- Se utilizan funciones de la biblioteca estándar de Python, como ``os.listdir``, ``os.path.isfile``, ``os.path.join``, ``os.makedirs``, para trabajar con archivos y directorios.
- Se utilizó el módulo ``os`` para manejar la creación de carpetas (``os.makedirs``), la obtención de nombres de archivos existentes (``os.listdir``), y la creación de rutas de archivos (``os.path.join``).
- Se utilizó el módulo ``openpyxl`` para trabajar con archivos Excel y guardar detalles de las consultas API en una hoja de cálculo.

6. Gráficos de Matplotlib:

- Se utilizan funciones de la biblioteca ``matplotlib.pyplot`` para crear gráficos de barras (``plt.bar``), gráficos de líneas (``plt.plot``), y gráficos circulares (``plt.pie``).
- Se utilizaron las funciones de la biblioteca ``matplotlib.pyplot`` para generar gráficos de estadísticas de partidas, tasa de victorias y derrotas, y campeones más usados. Los datos para estas gráficas se obtuvieron de las listas mencionadas anteriormente.