

Trabajo Práctico Número 2. Algoritmos y programación 1

Estructuras utilizadas:

Archivos:

“usuarios.csv” y “transacciones.csv”:

Diccionarios:

“equipos”: respuesta de la api, equipos de la Liga Argentina en 2023

“fixtures”: respuesta de la api, partidos de la Liga Argentina en 2023

“usuarios”:

Clave: “id_usuario”

Datos: los mismos que el archivo de usuarios.csv (y sin el id usuario)

“transacciones”:

Clave: “id_usuario”

Datos: listas de la forma [fecha:str,tipo:str,importe:float]

Librerías:

Requests, os, passlib, matplotlib, csv y tempfile.

Flujo del programa y explicación sobre las líneas/secciones más importantes del código:

- Importo librerías, y el módulo “Utilidades”.
- Declaro constantes la url de la api sin endpoint, la clave mediante la que hago consultas y la api.

Main:

- Cargo el contenido de los archivos en diccionarios.
- Permito al usuario iniciar sesión, registrarse o salir. La función de iniciar sesión se fija que los datos ingresados por el usuario correspondan a un usuario existente y sean correctos. La función de registrarse, permite al usuario crear una cuenta, ingresando un mail

que no exista ya en el diccionario, lo guarda en el diccionario y luego en el archivo de usuarios. La opción de salir hace que el resto de código no se ejecute, haciendo True el booleano de finalizar

- Si no eligió salir, realizo las consultas de equipos y partidos (fixtures) a la api, con la función consultar api, que recibe el endpoint y los parámetros de la consulta y devuelve la respuesta de la api.
- En el caso de los partidos/fixtures, guardo además para cada partido, la cantidad de veces que pagaría al usuario en caso de ganar, sin considerar el equipo que tenga win or draw, con la función obtener cantidad de veces, que me devuelve un random entre 1 y 4.
- En un ciclo while, que ejecuta hasta que el booleano finalizar sea True, ejecuto el resto de código y llamadas a funciones que voy a mencionar a continuación.
- Muestro en menú, y permito al usuario elegir una opción, si la opción es 0, finalizar se vuelve True y guardo (reescribo) en los archivos los diccionarios que usé en el programa.
- Si la opción es 1, si el diccionario de equipos tiene información, mostraré el plantel de un equipo ingresado por el usuario perteneciente a la lista de equipos que le permito ver. Para este caso, realizo una consulta a la api de jugadores (de liga argentina en 2023) filtrando ese equipo elegido.
- Si la opción es 2, el usuario me ingresa un año del cual quiere ver la tabla de posiciones, y consulto a la api las posiciones filtrando por liga y temporada y las guardo en un diccionario de posiciones. Si posiciones tiene información, las muestro al usuario.
- Si la opción es 3, y el diccionario equipos tiene información, le permito al usuario elegir uno para mostrar la información de su estadio y ver su escudo, la función de mostrar información estadio y escudo lo que hace es recorrer equipos y si el id coincide con el del equipo solicitado, me guardo la información del estadio y el equipo para luego mostrar lo que pide el enunciado. El escudo lo muestro guardando la imagen (que me da la api mediante una url) temporalmente con la librería tempfile, y la borro como ultima instrucción de la función.
- Si la opción es 4, y el diccionario de equipos tiene información, permito al usuario ingresar un equipo del que desee ver un gráfico de goles por intervalos de minutos, consulto con la api las estadísticas del equipo. Me guardo en un diccionario los goles por minuto y en una lista las claves de dicho diccionario que son los intervalos de minutos

y en otra lista guardo la cantidad de goles. Esas listas son las que uso para realizar el gráfico con matplotlib.

- Si la opción es 5, permito al usuario ingresar un monto de dinero y aumentárselo al dinero actual de su cuenta (en el diccionario) mediante la función modificar dinero usuario. Me debe ingresar la fecha actual para además guardar dicho deposito en el diccionario de transacciones.
- Si la opción es 6, busco el usuario que tenga el atributo de cantidad total apostada mas grande, y muestro su nombre y dicha cantidad.
- Si la opción es 7, creo un diccionario de ganado por usuario, con clave el id del usuario y dato la cantidad de veces que ganó. Recorro transacciones filtrando por tipo de transacción “Gana” y sumo 1 por cada transacción del tipo “Gana” para cada usuario. Luego veo cual es el usuario que más ganó y muestro su nombre, mail y dicha cantidad de veces ganadas.
- Si la opción es 8, si el usuario tiene dinero disponible y si los diccionarios equipos y fixtures tienen información, el usuario podrá apostar, mediante la función llamada apostar. Le permito que elija un equipo para ver los partidos del mismo, y le muestro quien es local y quien es visitante para cada fecha (únicamente de los partidos que sean posteriores a la fecha actual). Luego, el usuario me ingresa la fecha de un partido por el cual apostará. Consulto para dicha fecha quien tiene el win or draw, calculo la cantidad de veces que pagaría si gana en base a la cantidad de veces base que tengo guardada en fixtures y el win or draw. Guardo esta cantidad y se la muestro al usuario para que sepa cuanto paga cada uno en caso de que gane. El usuario me ingresa el monto a apostar, si le alcanza el dinero, le descuento el dinero en el diccionario de usuarios. Modifico en el usuario la cantidad total apostada (le sumo el monto de la nueva apuesta) y actualizo la fecha de su última apuesta con la fecha actual. Simulo el partido con un random. Si el usuario gana (su apuesta y el resultado simulado son iguales), le sumo a su dinero actual (en el diccionario) lo apostado más lo ganado, siendo lo ganado igual a lo apostado por la cantidad de veces que paga el equipo, en caso de empate es la mitad. Si el usuario pierde, no recupera el dinero apostado, ni recibe nada de dinero. Guardo en el diccionario de transacciones con la fecha actual la transacción, asigno “Gana” o “Pierde” dependiendo del resultado, si gana guardo el monto positivo ganado sin contar lo apostado y si pierde guardo el monto apostado negativo.

- Como mencioné anteriormente, cuando el usuario decide salir, antes de cerrar el programa reescribo los archivos de usuarios y transacciones, con la información de los diccionarios.

Observaciones:

A la hora de consultar los jugadores opté por consultarlos cada vez que el usuario pida ver un plantel, filtrando por equipo, ya que si quisiera guardarme correctamente todos los jugadores de la Liga Argentina 2023 en un diccionario como hago con equipos y fixtures, debería guardar todas las páginas de la respuesta de la api, y por la limitación de 100 consultas diarias opté por solucionarlo así finalmente.

En mi diccionario de fixtures agrego un atributo por cada partido, que es la cantidad de veces base que pagaría al usuario que apueste por ese partido y gane. Luego cuando el usuario elija uno para apostar, consulto el win or draw para pagarle menos si apuesta y gana por el equipo que lo tiene en true. (teniendo en cuenta que el win or draw está asociado a winner, y que puede ser true o false para winner, considero que si winner lo tiene en false, entonces el otro equipo lo tiene en true)

En el modulo de utilidades tengo funciones para el ingreso y validació de int, float, fecha y email.