

# Trabajo Práctico N° 2

[75.40] Algoritmos y Programación I

## *Apuestas*

*La adicción al juego no es un vicio, sino una peligrosa enfermedad*

Facultad de Ingeniería - Universidad de Buenos Aires

23 de mayo - Primer Cuatrimestre de 2023

## Enunciado

Recientemente un grupo de amigos decidió emprender y lanzar su portal de apuestas deportivas “Jugársela”. Como en Argentina el fanatismo por el fútbol es gigante, se decidió arrancar con una primera versión Beta que sólo permite apostar sobre los **partidos de fútbol de la Liga Argentina**. Para esto deciden contratarnos y nos detallan el alcance y los requerimientos funcionales que son necesarios.

## Tecnología

- Utilizar lenguaje Python versión +3.0.
- Autenticar a los usuario mediante un usuario y contraseña (deberá estar encriptada por alguna función de hash, ver librerías al final).
- API a utilizar <https://api-sports.io/>, precisamente <https://dashboard.api-football.com/>.

## Requerimientos

1. El programa solicitará usuario y contraseña, en caso que sea un usuario nuevo permitirá crearlo. Para esta funcionalidad existirá un archivo **usuarios.csv** que contendrá los siguiente campos separados por comas:
  - i. Id Usuario (mail)
  - ii. Nombre Usuario
  - iii. Contraseña (encriptada)
  - iv. Cantidad apostada hasta el momento
  - v. Fecha última apuesta YYYYMMDD
  - vi. Dinero disponible
2. Mostrar el plantel completo de un equipo ingresado por consola, para esto previamente listar todos los equipos de la Liga Profesional correspondiente a la temporada 2023.
3. Mostrar la tabla de posiciones de la Liga profesional, ingresando la temporada que se quiera consultar.
4. Permitir ingresar un equipo de fútbol y mostrar toda la información posible sobre el estadio y su escudo.
5. Realizar un gráfico que muestre los goles y los minutos en los que fueron realizados para un equipo ingresado por teclado (eje x: *minutos jugados*, eje y: *goles realizados*).
6. Permitir a un usuario cargar dinero en su cuenta.

7. Mostrar el usuario que más dinero apostó.
8. Mostrar el usuario que más veces ganó.
9. Apuestas:
  - i. El usuario ingresará un equipo y el programa deberá mostrar un listado del fixture y lo que paga por cada equipo para esa fecha (ver [pagos de apuesta por equipo](#)). El listado también debe mostrar quién es local y quién es visitante (se puede usar L/V).
  - ii. El usuario podrá elegir un partido para apostar.
  - iii. **La apuesta podrá ser Ganador(L)/Empate/Ganador(V)** (sólo una por vez), el usuario ingresará el monto a apostar (de su dinero disponible) y luego se simulará el resultado del partido lanzando un dado (si sale 1 gana el local, si sale 2 empata, si sale 3 gana visitante). El dinero ganado irá al dinero disponible del usuario.
  - iv. Toda transacción ya sea apuesta o ingreso de dinero del usuario debe ser registrada en un archivo CSV, "transacciones.csv", tendrá el siguiente formato:
    - i. Id Usuario (mail)
    - ii. Fecha YYYYMMDD
    - iii. Tipo de resultado (Gana/Pierde/Deposita)
    - iv. Importe (positivo si gana o deposita, negativo si pierde)

## Pagos de apuesta por equipo

Cada partido está relacionado con una predicción que estima la propia API. Esta predicción determina si el equipo gana o empata mediante el campo **win\_or\_draw: true**.

Para calcular el pago de las apuestas se tomará en forma random un número entre (1, 4] que determinará las veces que pagará lo apostado.

El equipo que posea el campo **win\_or\_draw** en **true**, pagará el 10% de las veces que paga la apuesta.

Ejemplo:

(L)Racing Club - (V)Platense (**win\_or\_draw = true para Platense**)

**Sale el nro random 3 y sale el dado 1 (gana local) ⇒**

Si se apuesta, Racing Club ganador ⇒ paga 3 veces lo apostado

Si se apuesta, Racing Club empate ⇒ paga 3 veces lo apostado

Si se apuesta, Platense empate ⇒ paga 0,3 veces lo apostado

Si se apuesta, Platense ganador ⇒ paga 0,3 veces lo apostado

## Documentación Requerida

Para la presentación del Trabajo Práctico, se pide al alumno realizar una breve documentación del mismo que especifique:

- qué estructuras utilizaron
- cuál es el flujo de su programa
- explicación sobre las líneas/secciones más importantes del código
- dificultades/observaciones

Esta información será utilizada por el equipo docente previo a corregir el trabajo por lo cual el alumno puede expresar allí lo que crea conveniente para ayudar en el proceso de corrección.

## Defensa Oral

La entrega del Trabajo Práctico estará acompañada de una defensa oral tanto grupal como individual. Se espera que tanto el alumno como el grupo en su totalidad pueda responder de forma completa las preguntas que llegara a plantear el docente en lo que respecta a la funcionalidad del trabajo, las herramientas provistas por el lenguaje de programación y el diseño del sistema en general.

## Librerías Permitidas

Para la realización del Trabajo Práctico, el alumno puede hacer uso de librerías únicamente avaladas previamente por el equipo docente. Algunas de ellas podrían ser:

- [random](#)
- [colored](#)
- [os](#)
- [passlib](#)

En caso de querer utilizar otra librería, el alumno deberá consultar con el equipo docente para confirmar que esté avalada efectivamente.

## Recomendaciones

Visualización de json: <https://jsonhero.io/>.

## Condiciones de Aprobación

**NO** se debe...

- usar **variables globales**
- usar **ciclos infinitos**
- escribir **código repetido**
- permitir al usuario **ingresar valores/datos erróneos**
- entregar un programa que **no funcione**
- entregar un programa con **errores graves**
- incumplir con la fecha de entrega

**SÍ** se debe...

- **validar el ingreso de datos**, de forma tal que el programa no se rompa ante el ingreso de valores/datos erróneos
- **modularizar** el código
- usar **buenas prácticas de programación**
- entregar un programa que cumpla con la **funcionalidad completa especificada**

## Fecha de Entrega

El Trabajo Práctico cuenta con 2 fechas de entrega: 22/06/2023 y 29/06/2023. Ambas tienen hora límite 18:00 hs. Mientras que la primera fecha tiene posibilidad de recuperación, la segunda no. En esta línea, se espera que el grupo entregue el trabajo completo para la primera fecha, y -en caso de recibir correcciones por parte del equipo docente- tendrán posibilidad de realizar una segunda entrega en la segunda fecha. Esta primera instancia de entrega no restará puntos en caso de re-entregar en la segunda fecha. La segunda fecha definirá la nota del trabajo según corresponda. Quienes decidan entregar únicamente en la segunda fecha (sin pasar por la primera) no tendrán posibilidad de correcciones intermedias y tendrán una nota final del trabajo según corresponda. Tanto la primera entrega como la segunda deben contener la totalidad de la funcionalidad solicitada, no se aceptarán entregas parciales en ninguna de las fechas.

## Importante

La aprobación del Trabajo Práctico N°2 es condición **necesaria** (junto con la aprobación del parcial) para **regularizar la materia**.