

## Examen parcial

Fechas del cronograma de entrega y corrección:

**Jueves 26 /10** presentación del problema.

**Martes 31/10** Pre entrega y **parcial de programación(Google Form)**, se piden cambios para la clase siguiente

**Jueves 2/11** Pedido final de cambios de estructura , informes y ordenamiento (60 minutos para hacer los cambios),despues defensa de lo entregado de manera oral

Escribe un programa en Python que cargue la información de los jugadores del Dream Team desde un archivo JSON y realice las siguientes tareas, teniendo en cuenta que cada una de ellas deberá de ser realizada por una función diferente:

El programa deberá contar con las siguientes clases: **Equipo, Jugador, Estadísticas**.

La Clase Equipo será la encargada de levantar el archivo JSON y realizar todas las tareas relacionadas a archivos. También se encargará de crear a cada jugador al leer el JSON y agregarlos a una lista de jugadores la cual deberá ser atributo de la clase Equipo.

Cada jugador tendrá como atributos una lista de logros, un nombre, posición y un objeto de clase **Estadistica**.

Cada estadística tendrá 12 atributos los cuales deberán tener sus respectivos getters & setters / properties.

Cada clase deberá tener sus respectivos Getters & Setters o properties.



- 1) Mostrar la lista de todos los jugadores del Dream Team. Con el formato:  
*Nombre Jugador - Posición. Ejemplo:*  
*Michael Jordan - Escolta*
- 2) Permitir al usuario seleccionar un jugador por su índice (validar con regex) y mostrar sus estadísticas completas, incluyendo temporadas jugadas, puntos totales, promedio de puntos por partido, rebotes totales, promedio de rebotes por partido, asistencias totales, promedio de asistencias por partido, robos totales, bloqueos totales, porcentaje de tiros de campo, porcentaje de tiros libres y porcentaje de tiros triples.
- 3) Después de mostrar las estadísticas de un jugador seleccionado por el usuario, permite al usuario guardar las estadísticas de ese jugador en un archivo CSV. El archivo CSV debe contener los siguientes campos: nombre, posición, temporadas, puntos totales, promedio de puntos por partido, rebotes totales, promedio de rebotes por partido, asistencias totales, promedio de asistencias por partido, robos totales, bloqueos totales, porcentaje de tiros de campo, porcentaje de tiros libres y porcentaje de tiros triples.
- 4) Permitir al usuario buscar un jugador por su nombre (validar con regex) y mostrar sus logros, como campeonatos de la NBA, participaciones en el All-Star y pertenencia al Salón de la Fama del Baloncesto, etc.
- 5) Calcular y mostrar el promedio de puntos por partido de todo el equipo del Dream Team, ordenado por nombre de manera ascendente.
- 6) Permitir al usuario ingresar el nombre de un jugador (validar con regex) y mostrar si ese jugador es miembro del Salón de la Fama del Baloncesto.
- 7) Calcular y mostrar el jugador con la mayor cantidad de rebotes totales.

Recuerda utilizar el archivo JSON proporcionado para cargar la información de los jugadores. Puedes utilizar las bibliotecas estándar de Python, como json, para leer el archivo JSON y procesar la información.

- Todas las funciones deberán estar correctamente documentadas.
- Las funciones y las variables deberán estar claramente nombradas y seguir las reglas de estilos utilizadas en la materia.
- Todos los puntos deberán poder ser accedidos a través de un menú de opciones.
- En todos los casos que se lean o se escriba archivos se deberán controlar las excepciones

Exportar a csv.

Ejemplo

Jugador	Puntos	Rebotes	Asistencias	Robos
Michael Jordan	1	1	1	2
Magic	2	3	4	4

NOTA:

#Utilizar los jugadores como objetos

#Para sacar un 10 , se debe realizar la interfaz en pygame, pero no es necesaria para aprobar el parcial.