**Proyecto de clase de Python aplicado.**

Fundamentos de Programación

**Nombre del Proyecto:Data League**

1. Información General

* Nombre estudiantes:

BRYAN STEVEN BELTRAN BARBOSA

KIVEN STEVE PIRAJAN DIAZ

MANUEL ALEJANDRO RINCON GOMEZ

NICOLAS ALEXANDER MEJÍA ROJAS

* Curso / Grupo:GRUPO A
* Fecha de entrega:17/10/2025
* Profesor:Pablo Enrique Carrreño Hernandez

**2. Título del Proyecto**

DATA LEAGUE

**3. Descripción del Proyecto**

Este proyecto consiste en la creación de un conjunto de datos de la Premier League correspondiente a la temporada 2023-2024, utilizando Python para su exploración y análisis. El propósito principal es ofrecer una herramienta que permita visualizar el desarrollo de la temporada regular, así como generar predicciones para futuras ediciones de la liga.

Está dirigido a aficionados del fútbol, analistas deportivos y personas interesadas en el comportamiento de la liga inglesa, quienes podrán acceder a los datos de forma clara y ordenada. A través de una tabla interactiva, el usuario puede consultar información relevante, que por defecto se muestra ordenada por puntaje de mayor a menor, aunque también tiene la opción de reorganizarla según sus intereses o necesidades específicas.

El resultado esperado es una experiencia accesible e informativa, que facilite el análisis de rendimiento de los equipos y permita sacar conclusiones o generar predicciones basadas en datos reales de la temporada.

**4. Objetivos**

**General:**

* Analizar las tendencias de la Premier League y su enfoque a los resultados obtenidos durante la temporada 23-24

**Específicos:**

* Implementar funciones, estructuras iterativas (como bucles for y while) y condicionales (if, else) para facilitar la modularidad, flexibilidad y reutilización del código.
* Utilizar librerías apropiadas junto con listas, tuplas y matrices para organizar y mostrar visualmente la información en forma de tablas claras y estructuradas.
* Documentar el código de manera detallada mediante comentarios, de modo que personas con conocimientos previos en programación puedan comprender el funcionamiento y la lógica detrás de cada sección del código sin dificultad

**5. Requisitos**

* Utilización de Python y todas sus funciones integradas
* Se necesita Python o Visual Studio Code para poder ejecutar el programa

**6. Diseño del Proyecto**

* **Arquitectura o estructura del programa:** (modularización, funciones, clases, etc.)
* **Diagrama de flujo:**
* 
* **Interfaz (si aplica):**

**7. Desarrollo**

* Explicación paso a paso de cómo se desarrolló el proyecto

**1.Primero, usamos algunas librerías para manejar datos, limpiar la pantalla y mostrar tablas bonitas.**

import pandas as pd # Para trabajar con tablas de datos

import os # Para limpiar la pantalla

import time # Para hacer pausas

from tabulate import tabulate # Para imprimir tablas con estilo

**2.Creamos una función que limpia la consola, para que todo se vea ordenado.**

def limpiar\_Pantalla():

if os.name == 'nt': # Si usas Windows

os.system('cls')

else: # Si usas Linux o Mac

os.system('clear')

**3.Aquí le preguntamos al usuario cuántos equipos quiere ingresar, y vamos pidiendo los datos uno por uno: partidos jugados, ganados, goles, tarjetas, etc.**

Además, calculamos cosas importantes como:

Diferencia de goles (goles a favor menos goles en contra

Puntos (3 por victoria, 1 por empate)

Porcentaje de victorias

def datos\_Equipos(datos):

cantidad = int(input("¿Cuántos equipos vas a ingresar? "))

for i in range(cantidad):

print(f"Ingresando datos del equipo {i + 1}")

nombre = input("Nombre del equipo: ")

PJ = int(input("Partidos jugados: "))

PG = int(input("Partidos ganados: "))

PE = int(input("Partidos empatados: "))

PP = int(input("Partidos perdidos: "))

GF = int(input("Goles a favor: "))

GC = int(input("Goles en contra: "))

DG = GF - GC

Puntos = PG \* 3 + PE

Prom\_Victorias = (PG / PJ) \* 100

TA = int(input("Tarjetas amarillas: "))

TR = int(input("Tarjetas rojas: "))

Dt = input("Entrenador: ")

Estadio = input("Estadio: ")

Temporada = "2024/2025"

datos.append((nombre, PJ, PG, PE, PP, GF, GC, DG, Puntos, Prom\_Victorias, TA, TR, Dt, Estadio, Temporada))

limpiar\_Pantalla()

**4.Para que todo se vea lindo y ordenado, usamos pandas para organizar los datos y tabulate para imprimir tablas con bordes y todo centrado.**

def imprimir\_Tabla(df):

print(tabulate(

df,

headers='keys',

tablefmt='fancy\_grid',

showindex=False,

stralign='center',

numalign='center'

))

**5.Por último, mostramos:**

La tabla completa con todos los dato,la tabla ordenada por puntos, de mayor a menor

Equipos con más de 10 puntos

Equipos con más del 50% de victorias

Y una tabla resumida con solo puntos, entrenador y temporada

def generar\_tablas(datos):

columnas = ["EQUIPO","PJ", "PG", "PE", "PP", "GF", "GC", "DG", "PUNTOS",

"% VICTORIAS", "TARJETAS AMARILLAS", "TARJETAS ROJAS",

"ENTRENADOR", "ESTADIO", "TEMPORADA"]

df = pd.DataFrame(datos, columns=columnas)

print("\nTabla original:")

imprimir\_Tabla(df)

print("\nTabla ordenada por PUNTOS (de mayor a menor):")

imprimir\_Tabla(df.sort\_values(by='PUNTOS', ascending=False))

print("\nEquipos con más de 10 puntos:")

imprimir\_Tabla(df[df['PUNTOS'] > 10])

print("\nEquipos con porcentaje de victorias mayor al 50%:")

imprimir\_Tabla(df[df['% VICTORIAS'] > 50])

print("\nTabla con columnas seleccionadas (PUNTOS, ENTRENADOR, TEMPORADA):")

imprimir\_Tabla(df[['PUNTOS', 'ENTRENADOR', 'TEMPORADA']])

Fragmentos de código relevantes comentados

* Descripción de las funciones principales

**8. Pruebas y Resultados**

* Cómo se probó el programa
* Capturas de pantalla o ejemplos de ejecución
* Resultados obtenidos
* Manual de usuario

**9. Conclusiones**

* Lecciones aprendidas
* Dificultades encontradas y cómo se resolvieron
* Posibles mejoras o ideas futuras

**10. Bibliografía / Recursos**

* Sitios web, documentación, libros o videos utilizados, mínimo 10