# Proyecto de clase de Python aplicado

Fundamentos de Programación

**CODE-LEAGUE COMPARADOR**

## 1. Información General

• Nombre estudiantes: Jonathan Parra Landinez , David Santiago Granja Espinosa, Juan Enrique Galeano Vargas y Carol Daniela Muñoz Cubides  
• Curso/Grupo: A  
• Fecha de primera entrega: 16/10/2025  
• Profesor: Pablo Enrique Carreño

## 2. Título del Proyecto

Análisis de datos de la Premier League.

## 3. Descripción del Proyecto

Este proyecto tiene como propósito analizar las estadísticas de la liga inglesa o Premier League, con el fin de extraer información clave sobre el rendimiento de los equipos, partidos y estadísticas de los jugadores. La idea es facilitar el acceso a datos relevantes de forma automatizada, utilizando herramientas de programación en Python.  
  
**Propósito del proyecto:**  
Este proyecto les permite a los usuarios obtener información clave de manera automatizada y eficiente, resolviendo el problema de la inaccesibilidad de interpretar grandes cantidades de datos estadísticos sin las herramientas adecuadas. Además, automatizar la recopilación y el análisis de estadísticas les permite a los usuarios centrarse en el análisis en lugar de la extensa recolección manual de datos. Permitiendoles disfrutar de sus actividades o hobbies como este sin incomodidad alguna.  
 **Público objetivo:**  
Este proyecto está dirigido principalmente a fanáticos del fútbol, analistas deportivos, entrenadores y periodistas, quienes podrán usar el programa con el fin de analizar los resultados obtenidos, seguir las estadísticas de los equipos y jugadores, y generar reportes de manera más sencilla.  
  
**Resultado esperado:**  
Esperamos que este proyecto genere estadísticas las cuales puedan ser visualizadas y de esta manera contribuyan al proceso de identificar patrones, tendencias y predicciones sobre las estadísticas de los jugadores y equipos de la Premier League. Al final, el objetivo es mejorar la comprensión de las estadísticas y proporcionar a los usuarios una herramienta potente que aunque abarque bastante informacion permita hacer análisis deportivos profundos de manera rápida y precisa.

## 4. Objetivos

**General**:

Desarrollar una herramienta que permita analizar las estadísticas deportivas de la Premier League temporada 2024-25 de forma automatizada, que permita obtener, procesar y visualizar datos relevantes sobre los equipos que compiten en esta misma (incluyendo partidos, goles y asistencias). Facilitando a los fanáticos, analistas deportivos, entrenadores y periodistas a tener esta información de manera clara, sencilla y actualizada.

**Específicos**:

• Mostrar las listas, tuplas, diccionarios, para almacenar y manipular grandes cantidades de datos.

• Implementar funciones para tareas específicas como para cargar los datos y procesamiento de estadísticas, calcular estadísticas simples como goles, partidos jugados, ganados, empatados, perdidos, total de goles por equipo, puntos en la temporada y demás, usando también la modularización permitiéndonos organizar el código en diferentes módulos.

• Aplicar los **temas vistos en clase** como condicionales, ciclos y manejo de entradas del usuario.

• Fortalecer la lógica de programación mediante la creación de un programa que responda a consultas dinámicas del usuario.

• Familiarizarse con el uso de Python en entornos reales, conectando conceptos teóricos con un proyecto práctico.

## 5. Requisitos

## Lenguaje de programación: Python 3.10 o superior

## Editor recomendado: Visual Studio Code (o cualquier IDE compatible con Python)

## Librerías utilizadas:

## pandas: para manejar y analizar los datos de la Premier League.

## openpyxl: para permitir que pandas lea archivos Excel (.xlsx).

## Archivo de datos:

## TABLAPREMIER.xlsx (ubicado en la carpeta data/, con una hoja llamada “Hoja2”).

## Contiene columnas como *Equipo*, *Nombre*, *Posición*, *Dorsal*, y estadísticas.

## Modo de ejecución:

## Por consola, usando input() para que el usuario ingrese nombres de equipos o jugadores.

## Requisitos del sistema:

## Sistema operativo Windows, macOS o Linux.

## Python instalado con las librerías mencionadas (pueden instalarse con pip install pandas openpyxl).

## Requisitos futuros (para posibles mejoras que queremos implementar)

## Estos recursos y herramientas pueden implementarse más adelante para ampliar las funcionalidades y profesionalizar el proyecto:

## Librerías adicionales:

## matplotlib o seaborn: para crear gráficas estadísticas de equipos o jugadores.

## tkinter o streamlit: para desarrollar una interfaz gráfica o web interactiva.

## numpy: para realizar cálculos numéricos más eficientes.

## Requisitos de documentación:

## Manual de usuario.

## Comentarios descriptivos en cada función.

## Diagrama de flujo digital o pseudocódigo complementario.

## 6. Diseño del Proyecto

• Arquitectura o estructura del programa: (modularización, funciones, clases, etc.)

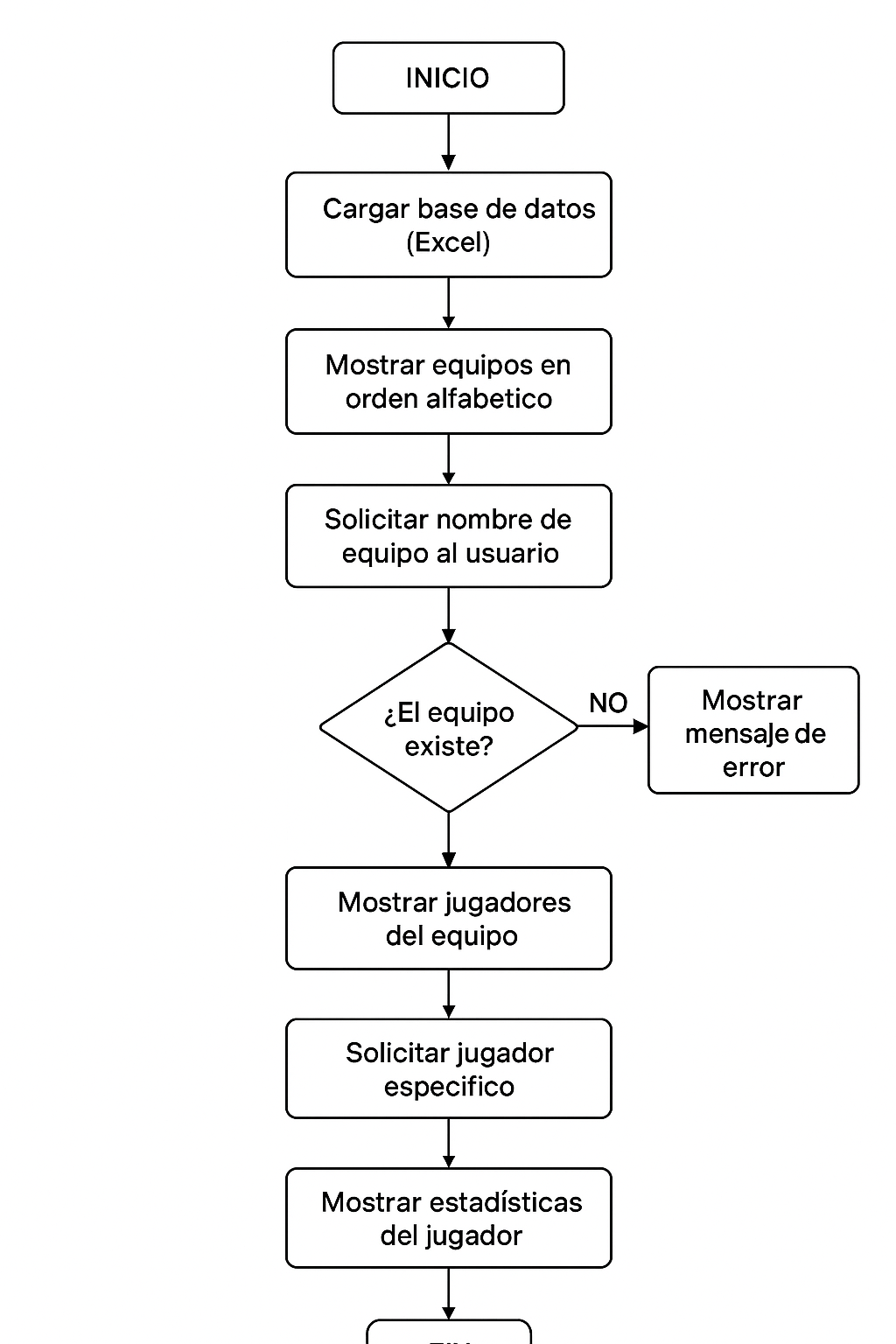
El proyecto implementa una **estructura modular**, dividiendo las tareas principales en **funciones específicas**, lo que facilita la comprensión y mantenimiento del código. Las funciones principales son:

- **Jugadores\_Equipo():** solicita el nombre del equipo y devuelve un DataFrame con los jugadores del equipo.

**- Solicitar\_Jugadores(Jugadores**): muestra la lista de jugadores (nombre, posición, dorsal) y permite seleccionar un jugador para ver sus estadísticas.

Estas funciones permiten separar la lógica de entrada/validación, el filtrado de datos y la presentación de resultados. En proyectos futuros se recomendaría agregar clases para representar entidades (Equipo, Jugador) si la complejidad crece.

**DIAGRAMA DE FLUJO INICIAL:**



## 7. Desarrollo

• Explicación paso a paso de cómo se desarrolló el proyecto:

1. Se importó la librería pandas y se cargó la hoja de Excel en un DataFrame.

2. Se obtuvieron los equipos únicos con BASE\_DATOS["Equipo"].unique() y se ordenaron con sorted(). (Funciones que obtuvimos a parir de las librerías)

3. Se definieron las funciones Jugadores\_Equipo() y Solicitar\_Jugadores(Jugadores) para manejar la interacción y filtrado.

4. Se imprimieron en consola los jugadores del equipo seleccionado y se permitió consultar estadísticas por jugador.

• Fragmentos de código relevantes comentados:

**# IMPORTAMOS LIBRERIA PANDAS**  
import pandas as pd   
  
**# Cargar datos desde Excel (Hoja2)**BASE\_DATOS = pd.read\_excel('data/TABLAPREMIER.xlsx', sheet\_name='Hoja2')  
  
**# Obtener equipos únicos**  
Nombre\_Equipos = BASE\_DATOS["Equipo"].unique()  
for Orden\_Equipo in sorted(Nombre\_Equipos):  
 print("-", Orden\_Equipo)  
 **# Función para obtener jugadores de un equipo**  
def Jugadores\_Equipo():  
 Solicitud\_Equipo = input("Ingrese el nombre del equipo que desea ver sus jugadores: ")  
 if Solicitud\_Equipo in Nombre\_Equipos:  
 Jugadores = BASE\_DATOS[BASE\_DATOS["Equipo"].str.lower() == Solicitud\_Equipo.lower()]  
 return Jugadores   
 else:  
 print("El equipo ingresado no se encuentra en la base de datos, revisa la ortografia e intentalo de nuevo :)")  
 **# Función para mostrar jugadores y solicitar uno**  
def Solicitar\_Jugadores(Jugadores):  
 if len(Jugadores) > 0:  
 print("Los JUGADORES del equipo son: ")  
 for \_, Jugador in Jugadores.iterrows():  
 print("-", Jugador["Nombre"], "(", Jugador["Posicion"], ")", "#", Jugador["Dorsal"])   
 Solicitud\_Jugador = input("Ingrese el nombre completo del jugador que desea ver sus estadisticas: ")  
 if Solicitud\_Jugador in BASE\_DATOS["Nombre"]:  
 Estadisticas\_Jugador = BASE\_DATOS[BASE\_DATOS["Nombre"].str.lower() == Solicitud\_Jugador.lower()]

• Descripción de las funciones principales (explicación detallada):

Jugadores\_Equipo():  
 - Solicita al usuario el nombre del equipo.  
 - Valida si el equipo existe en la lista de equipos únicos.  
 - Filtra el DataFrame para devolver las filas correspondientes al equipo seleccionado.  
 - Retorna un DataFrame con los jugadores del equipo o muestra un mensaje de error si no existe.  
  
Solicitar\_Jugadores(Jugadores):  
 - Recibe un DataFrame con los jugadores de un equipo.  
 - Verifica que el DataFrame no esté vacío.  
 - Recorre las filas e imprime nombre, posición y dorsal de cada jugador.  
 - Solicita al usuario el nombre de un jugador y filtra sus estadísticas en BASE\_DATOS.

• Proyecciones a futuro y mejoras planteadas:

- Implementar una interfaz gráfica para mejorar usabilidad y proyección de la información al usuario

- Exportar resultados a Excel para compartir o un tipo de documento compartible

- Agregar visualizaciones (gráficas de rendimiento, mapa de calor de estadísticas).