30 DE OCTUBRE DE 2025

EJERCICIO FUTBOL PYTHON

ENRIQUE GONZÁLEZ

CAMPUSFP
1º DAM GETAFE

ENRIQUE GONZÁLEZ
1º DAM GETAFE

INDICE

ENRIQUE GONZÁLEZ
1º DAM GETAFE

GUÍA:

MÓDULO 1 — Gestión de equipos

Objetivo: CRUD de equipos.

Modelo (por equipo):

{"id": 1, "nombre": "Tiburones", "ciudad": "Getafe", "activo": True}

Colección en memoria: equipos: list[dict].

Requisitos funcionales

1. **Crear equipo**: pedir nombre y ciudad (no vacíos), generar id único, activo=True.
2. **Listar equipos**: mostrar todos los activos.
3. **Buscar por id**: mostrar ficha o “no encontrado”.
4. **Actualizar datos**: nombre/ciudad por id.
5. **Eliminar equipo**: por id. (Nota: en Módulo 2 bloquea si tiene jugadores). No se elimina el equipo si no que se pasa el campo activo a False.
6. **Menú del módulo** en bucle (while + if/match) que vuelve al menú principal.

Muestra listados en tabla con tabulate.

MÓDULO 2 — Gestión de jugadores (reutiliza Módulo 1)

Objetivo: CRUD de jugadores, vinculados a equipos.

Modelo (por jugador):

{"id": 1, "nombre": "Laura", "posicion": "Delantera", "equipo_id": 1, "activo": True}

Colección: jugadores: list[dict].

Requisitos funcionales

1. **Alta de jugador**: pedir nombre, posicion y pedir al usuario equipo_id.
    - Validar que el equipo **exista** y esté **activo**.
2. **Listar jugadores**: todos o solo los de un equipo_id.
3. **Buscar por id**: mostrar ficha.
4. **Actualizar**: nombre/posicion/equipo_id
5. **Eliminar jugador**: por id.(No se elimina si no que se modifica el campo activo.
6. **Menú del módulo** en bucle, regreso al menú principal.

Cuando listes jugadores, muestra también el **nombre del equipo** (buscando en equipos).

MÓDULO 3 — Calendario y partidos (reutiliza 1+2)

Objetivo: crear y gestionar el calendario de partidos.

Modelo (por partido):

{

"id": 1, "jornada": 3,

"local_id": 1, "visitante_id": 2,

"fecha": "2025-11-22", "hora": "18:30",

"jugado": False, "resultado": None (golesLocal, golesvisitante)

}

Colección: partidos: list[dict].

Requisitos funcionales

1. **Crear partido**: jornada ≥ 1, local_id != visitante_id, ambos equipos **activos**.
    - (Opcional) Validación básica de fecha/hora con datetime.strptime.
2. **Listar partidos**: todos o por jornada; mostrar nombres de equipos.
3. **Reprogramar**: cambiar fecha/hora si jugado == False.
4. **Eliminar partido**: solo si jugado == False. (Si se elimina el partido completo)
5. **Menú del módulo** en bucle, regreso al principal.

Opcional: Evita **duplicar** el mismo enfrentamiento en la **misma jornada**.

MÓDULO 4 — Resultados y clasificación (reutiliza 1+2+3)

Objetivo: registrar resultados, calcular la tabla y ver estadísticas.

Requisitos funcionales

1. **Registrar resultado**: elegir id_partido pendiente y guardar resultado = (gL, gV) (enteros ≥ 0); marcar jugado=True.
2. **Clasificación**:
    - Por equipo calcula: PJ, G, E, P, GF, GC, DG, PTS (3 victoria, 1 empate, 0 derrota).
    - Ordena por PTS desc.
    - Muestra en **tabla**.

3. **Estadísticas por equipo**: dado un equipo_id, devolver un pequeño resumen (por ejemplo: PJ, GF, GC, PTS).
4. **Menú del módulo** en bucle, regreso al principal.

No recalcules partidos ya contados: usa el flag jugado.

Organización mínima (sugerida, simple)

- **app.py** o **main.py** (menú principal que llama a los módulos).
- **utiles.py** (pequeñas utilidades compartidas: generar id, leer enteros/strings, pintar tablas).
- **equipos.py**, **jugadores.py**, **calendario.py**, **ranking.py** (funciones de cada módulo).

No hace falta una jerarquía compleja: **5–6 archivos .py** son suficientes.

Pistas ligeras

- Genera id como max(lista_ids)+1 o con un contador global sencillo en utiles.py.
- Centraliza en utiles.py funciones como:
    - leer_int(mensaje, minimo=None), leer_texto(mensaje).
    - imprimir_tabla(filas, columnas) con tabulate.
- Mantén los **listados** siempre con **columnas fijas** para que la salida sea consistente.

Menú principal (app.py)

Incluye opciones para entrar/salir de cada módulo:

1. Gestión de equipos
2. Gestión de jugadores
3. Calendario de partidos
4. Resultados y clasificación
5. Salir

Controla con while y if/match. No uses break; usa condiciones del bucle.

Criterios de evaluación

1. **Funciones** claras y reutilizadas (no repetir la misma lógica).
2. **Datos en memoria** con **listas/diccionarios/tuplas** coherentes.
3. **Validaciones** mínimas (ids existentes, activos, rangos correctos).
4. **Flujo estable** con menús (while, if/match), sin bloquear.
5. **Salidas legibles** (tablas con tabulate).

6. **Escalado** entre módulos: cada uno se apoya en los previos sin reescribir.

MÓDULO 1: Creación de equipos.

CÓDIGO:

main.py

```
#MAIN

#IMPORTAR FUNCIONES Y LIBRERIAS
import tabulate
import rich
from equipos import menu_equipos

#MENU PRINCIPAL
def menu_principal():
  equipos=[]
  opcion = 0
  while opcion != 5:
    print("---- MENÚ PRINCIPAL ----")
    print("1. Gestión de equipos")
    print("2. Gestión de jugadores")
    print("3. Calendarios de partidos")
    print("4. Resultados y clasificación")
    print("5. Salir")
    print("-------------------------")

    try:
      opcion = int(input("Selecciona una opción: "))
    except:
      opcion = 0

    if opcion == 1:
      equipos = menu_equipos(equipos)
    elif opcion == 2:
      print("Módulo jugadores (pendiente)")
    elif opcion == 3:
      print("Módulo calendario (pendiente)")
    elif opcion == 4:
      print("Módulo ranking (pendiente)")
    elif opcion == 5:
      print("Programa finalizado.")
    else:
      print("Opción no válida.")

#EJECUTAR PROGRAMA
if __name__ == "__main__":
  menu_principal()
```

equipos.py

```
#EQUIPOS

from tabulate import tabulate

#DEFINIR VARIABLES
equipos = []
ultimo_id = 0

#FUNCIONES
#MENU DEL MODULO EQUIPOS
def menu_equipos(equipos):
  opcion = 0
  while opcion != 6:
    print("--- MENÚ DE EQUIPOS ---")
    print("1. Crear equipo")
    print("2. Listar equipos")
    print("3. Buscar equipo por ID")
    print("4. Actualizar equipo")
    print("5. Eliminar equipo (dar de baja)")
    print("6. Volver al menú principal")

    try:
      opcion = int(input("Selecciona una opción: "))
    except:
      opcion = 0

    if opcion == 1:
      equipos = crear_equipo(equipos)
    elif opcion == 2:
      listar_equipos(equipos)
    elif opcion == 3:
      buscar_equipo(equipos)
    elif opcion == 4:
      equipos = actualizar_equipo(equipos)
    elif opcion == 5:
      equipos = eliminar_equipo(equipos)
    elif opcion == 6:
      print("Volviendo al menú principal...")
    else:
      print("Opción no válida.")
  return equipos


#CREAR EQUIPO
def crear_equipo(equipos):
  print("--- Crear equipo nuevo ---")
  nombre = input("Nombre del equipo: ").strip()
  ciudad = input("Ciudad del equipo: ").strip()
```

```
    if nombre == "" or ciudad == "":
        print("El nombre y ciudad no pueden estar vacíos!")
        return equipos

    # GENERAR ID ÚNICO (selecciona el mayor y le suma 1)
    nuevo_id = 1
    for e in equipos:
        if e["id"] >= nuevo_id:
            nuevo_id = e["id"] + 1

    # DICCIONARIO
    equipo = {
        "id": nuevo_id,
        "nombre": nombre,
        "ciudad": ciudad,
        "activo": True
    }

    equipos.append(equipo)
    print("El equipo ha sido creado.\n")
    return equipos

#LISTAR EQUIPOS ACTIVOS
def listar_equipos(equipos):
    print("--- Lista de equipos activos ---")
    activos = []
    for e in equipos:
        if e["activo"]:
            activos.append([e["id"], e["nombre"], e["ciudad"]])
    if len(activos) == 0:
        print("No hay equipos activos todavía.")
    else:
        print(tabulate(activos, headers=["ID", "Nombre", "Ciudad"],
tablefmt="grid"))

#BUSCAR EQUIPOS POR ID
def buscar_equipo(equipos):
    print("--- Buscar equipos por su ID ---")
    try:
        id_buscar = int(input("Introduce el ID del equipo a buscar: "))
    except:
        print("ID inválido.")
        return

    encontrado = None
    for e in equipos:
        if e["id"] == id_buscar:
```

```
            encontrado = e

    if encontrado == None:
        print("Equipo no encontrado.")
    else:
        datos = [[encontrado["id"], encontrado["nombre"],
encontrado["ciudad"], encontrado["activo"]]]
        print(tabulate(datos, headers=["ID", "Nombre", "Ciudad", "Activo"],
tablefmt="grid"))


#ACTUALIZAR DATOS
def actualizar_equipo(equipos):
    print("--- Actualizar datos equipo ---")
    try:
        id_actualizar = int(input("Introduce el ID del equipo a modificar:
"))
    except:
        print("ID de equipo no válido")
        return equipos

    equipo = None
    for e in equipos:
        if e["id"] == id_actualizar:
            equipo = e

    if equipo == None:
        print("Equipo no encontrado.")
        return equipos

    print("Nota: Si dejas la casilla en blanco, no se cambiará.")
    nuevo_nombre = input("Nuevo nombre para el equipo: ").strip()
    nueva_ciudad = input("Nueva ciudad para el equipo: ").strip()

    if nuevo_nombre != "":
        equipo["nombre"] = nuevo_nombre
    if nueva_ciudad != "":
        equipo["ciudad"] = nueva_ciudad

    print("El equipo ha sido actualizado!\n")
    return equipos


#ELIMINACIÓN DE EQUIPOS
def eliminar_equipo(equipos):
    print("--- Eliminar equipo ---")
    try:
        id_eliminar = int(input("Introduce el ID del equipo a eliminar: "))
    except:
        print("ID inválido.")
```

```
    return equipos

  encontrado = None
  for e in equipos:
    if e["id"] == id_eliminar:
      encontrado = e

  if encontrado == None:
    print("Equipo no encontrado.")
  else:
    encontrado["activo"] = False
    print("El equipo ha sido eliminado correctamente.\n")

  return equipos
```

COMPROBACIÓN:

```
---- MENÚ PRINCIPAL ----
1. Gestión de equipos
2. Gestión de jugadores
3. Calendarios de partidos
4. Resultados y clasificación
5. Salir
------------------------
Selecciona una opción: 1
--- MENÚ DE EQUIPOS ---
1. Crear equipo
2. Listar equipos
3. Buscar equipo por ID
4. Actualizar equipo
5. Eliminar equipo (dar de baja)
6. Volver al menú principal
Selecciona una opción: 1
--- Crear equipo nuevo ---
Nombre del equipo: KikesFC
Ciudad del equipo: Leganes
El equipo ha sido creado.

--- MENÚ DE EQUIPOS ---
1. Crear equipo
2. Listar equipos
3. Buscar equipo por ID
4. Actualizar equipo
5. Eliminar equipo (dar de baja)
6. Volver al menú principal
Selecciona una opción: 1
--- Crear equipo nuevo ---
Nombre del equipo: John AndersonFC
Ciudad del equipo: Matrix
El equipo ha sido creado.

--- MENÚ DE EQUIPOS ---
1. Crear equipo
2. Listar equipos
3. Buscar equipo por ID
4. Actualizar equipo
5. Eliminar equipo (dar de baja)
6. Volver al menú principal
Selecciona una opción: 2
--- Lista de equipos activos ---
+------+-----------------+----------+
|   ID | Nombre          | Ciudad   |
+======+=================+==========+
|    1 | KikesFC         | Leganes  |
+------+-----------------+----------+
|    2 | John AndersonFC | Matrix   |
+------+-----------------+----------+
```

```
--- MENÚ DE EQUIPOS ---
1. Crear equipo
2. Listar equipos
3. Buscar equipo por ID
4. Actualizar equipo
5. Eliminar equipo (dar de baja)
6. Volver al menú principal
Selecciona una opción: 3
--- Buscar equipos por su ID ---
Introduce el ID del equipo a buscar: 2
+------+----------------+----------+----------+
|   ID | Nombre         | Ciudad   | Activo   |
+======+================+==========+==========+
|    2 | John AndersonFC | Matrix   | True     |
+------+----------------+----------+----------+
--- MENÚ DE EQUIPOS ---
1. Crear equipo
2. Listar equipos
3. Buscar equipo por ID
4. Actualizar equipo
5. Eliminar equipo (dar de baja)
6. Volver al menú principal
Selecciona una opción: 4
--- Actualizar datos equipo ---
Introduce el ID del equipo a modificar: 1
Nota: Si dejas la casilla en blanco, no se cambiará.
Nuevo nombre para el equipo: KikinioBrasilFC
Nueva ciudad para el equipo: Sao Paolo
El equipo ha sido actualizado!

--- MENÚ DE EQUIPOS ---
1. Crear equipo
2. Listar equipos
3. Buscar equipo por ID
4. Actualizar equipo
5. Eliminar equipo (dar de baja)
6. Volver al menú principal
Selecciona una opción: 2
--- Lista de equipos activos ---
+------+----------------+-----------+
|   ID | Nombre         | Ciudad    |
+======+================+===========+
|    1 | KikinioBrasilFC | Sao Paolo |
+------+----------------+-----------+
|    2 | John AndersonFC | Matrix    |
+------+----------------+-----------+
```

```
--- MENÚ DE EQUIPOS ---
1. Crear equipo
2. Listar equipos
3. Buscar equipo por ID
4. Actualizar equipo
5. Eliminar equipo (dar de baja)
6. Volver al menú principal
Selecciona una opción: 5
--- Eliminar equipo ---
Introduce el ID del equipo a eliminar: 2
El equipo ha sido eliminado correctamente.

--- MENÚ DE EQUIPOS ---
1. Crear equipo
2. Listar equipos
3. Buscar equipo por ID
4. Actualizar equipo
5. Eliminar equipo (dar de baja)
6. Volver al menú principal
Selecciona una opción: 2
--- Lista de equipos activos ---
+------+-----------------+-----------+
|   ID | Nombre          | Ciudad    |
+======+=================+===========+
|    1 | KikinioBrasilFC | Sao Paolo |
+------+-----------------+-----------+
--- MENÚ DE EQUIPOS ---
1. Crear equipo
2. Listar equipos
3. Buscar equipo por ID
4. Actualizar equipo
5. Eliminar equipo (dar de baja)
6. Volver al menú principal
Selecciona una opción: 6
Volviendo al menú principal...
---- MENÚ PRINCIPAL ----
1. Gestión de equipos
2. Gestión de jugadores
3. Calendarios de partidos
4. Resultados y clasificación
5. Salir
------------------------
```

MÓDULO 2: Creación de jugadores

CÓDIGO:

main.py

Ha sido añadida la variable “jugadores=[]” y “jugadores = menu_jugadores(jugadores, equipos)” en la opcion 2 del menú.

jugadores.py

```
#JUGADORES

from tabulate import tabulate

#FUNCIONES
#MENÚ MODULO JUGADORES
def menu_jugadores(jugadores, equipos):
  opcion = 0
  while opcion != 6:
    print("\n===== MENÚ DE JUGADORES =====")
    print("1. Crear jugador")
    print("2. Listar jugadores")
    print("3. Buscar jugador por ID")
    print("4. Actualizar jugador")
    print("5. Eliminar jugador")
    print("6. Volver al menú principal")

    try:
      opcion = int(input("Selecciona una opción: "))
    except:
      opcion = 0

    if opcion == 1:
      jugadores = crear_jugador(jugadores, equipos)
    elif opcion == 2:
      listar_jugadores(jugadores, equipos)
    elif opcion == 3:
      buscar_jugador(jugadores, equipos)
    elif opcion == 4:
      jugadores = actualizar_jugador(jugadores)
    elif opcion == 5:
      jugadores = eliminar_jugador(jugadores)
    elif opcion == 6:
      print("Volviendo al menú principal...")
    else:
      print("Opción no válida.")
  return jugadores

#CREAR JUGADOR
```

```
def crear_jugador(jugadores, equipos):
  print("\n--- Crear jugador nuevo ---")
  nombre=input("Nombre del jugador: ").strip()
  edad=input("Edad del jugador: ").strip()

  if nombre =="" or edad =="":
    print("El nombre y la edad no pueden estar vacíos.")
    return jugadores

  #SELECCIONAR EQUIPO EXISTENTE
  print("\nEquipos disponibles:")
  activos = []
  for e in equipos:
    if e["activo"]:
      activos.append([e["id"], e["nombre"]])
  if len(activos) == 0:
    print("No hay equipos activos, no se puede crear jugador, CREA UN
EQUIPO PRIMERO.")
    return jugadores
  else:
    print(tabulate(activos, headers=["ID", "Nombre"], tablefmt="grid"))

  try:
    id_equipo = int(input("Introduce el ID del equipo al que pertenece:
"))
  except:
    print("ID de equipo no válido.")
    return jugadores

  #VERIFICAR QUE EQUIPO EXISTE Y ESTA ACTIVO
  existe=False
  for e in equipos:
    if e["id"] ==id_equipo and e["activo"]:
      existe=True
  if not existe:
    print("Equipo no encontrado o inactivo.")
    return jugadores

  #GENERAR ID JUGADOR
  nuevo_id=1
  for j in jugadores:
    if j["id"] >=nuevo_id:
      nuevo_id=j["id"] + 1
  jugador= {
    "id": nuevo_id,
    "nombre": nombre,
    "edad": edad,
    "equipo_id": id_equipo,
    "activo": True
```

```
    }

    jugadores.append(jugador)
    print("¡Jugador creado correctamente!\n")
    return jugadores

#LISTAR JUGADORES ACTIVOS
def listar_jugadores(jugadores, equipos):
    print("\n--- Lista de jugadores activos ---")
    activos=[]
    for j in jugadores:
        if j["activo"]:
            nombre_equipo=""
            for e in equipos:
                if e["id"]==j["equipo_id"]:
                    nombre_equipo=e["nombre"]
            activos.append([j["id"], j["nombre"], j["edad"], nombre_equipo])
    if len(activos)==0:
        print("No hay jugadores activos todavía.")
    else:
        print(tabulate(activos, headers=["ID", "Nombre", "Edad", "Equipo"],
tablefmt="grid"))

#BUSCAR JUGADOR POR ID
def buscar_jugador(jugadores, equipos):
    print("\n--- Buscar jugador por ID ---")
    try:
        id_buscar=int(input("Introduce el ID del jugador: "))
    except:
        print("ID inválido.")
        return
    encontrado=None
    for j in jugadores:
        if j["id"]== id_buscar:
            encontrado= j

    if encontrado==None:
        print("Jugador no encontrado.")
    else:
        equipo_nombre =""
        for e in equipos:
            if e["id"]==encontrado["equipo_id"]:
                equipo_nombre=e["nombre"]

        datos=[[encontrado["id"], encontrado["nombre"], encontrado["edad"],
equipo_nombre, encontrado["activo"]]]
        print(tabulate(datos, headers=["ID", "Nombre", "Edad", "Equipo",
"Activo"], tablefmt="grid"))
```

```
#ACTUALIZAR DATOS JUGADOR
def actualizar_jugador(jugadores):
  print("\n--- Actualizar datos del jugador ---")
  try:
    id_actualizar=int(input("Introduce el ID del jugador a modificar: "))
  except:
    print("ID inválido.")
    return jugadores

  jugador=None
  for j in jugadores:
    if j["id"] ==id_actualizar:
      jugador =j
  if jugador ==None:
    print("Jugador no encontrado.")
    return jugadores

  print("Nota: si dejas un campo vacío, no se cambiará.")
  nuevo_nombre = input("Nuevo nombre: ").strip()
  nueva_edad = input("Nueva edad: ").strip()
  if nuevo_nombre != "":
    jugador["nombre"] =nuevo_nombre
  if nueva_edad != "":
    jugador["edad"] =nueva_edad

  print("Jugador actualizado correctamente.\n")
  return jugadores

#ELIMINAR JUGADOR
def eliminar_jugador(jugadores):
  print("\n--- Eliminar jugador ---")
  try:
    id_eliminar=int(input("Introduce el ID del jugador a eliminar: "))
  except:
    print("ID no válido.")
    return jugadores

  encontrado=None
  for j in jugadores:
    if j["id"] ==id_eliminar:
      encontrado =j

  if encontrado ==None:
    print("Jugador no encontrado.")
  else:
    encontrado["activo"] = False
    print("El jugador ha sido eliminado correctamente.\n")
  return jugadores
```

COMPROBACIÓN:

```
===== MENÚ DE JUGADORES =====
1. Crear jugador
2. Listar jugadores
3. Buscar jugador por ID
4. Actualizar jugador
5. Eliminar jugador
6. Volver al menú principal
Selecciona una opción: 1

--- Crear jugador nuevo ---
Nombre del jugador: Kike Ronaldo
Edad del jugador: 22

Equipos disponibles:
+------+----------+
|   ID | Nombre   |
+======+==========+
|    1 | KikeFC   |
+------+----------+
|    2 | CarolFC  |
+------+----------+
Introduce el ID del equipo al que pertenece: 1
¡Jugador creado correctamente!


===== MENÚ DE JUGADORES =====
1. Crear jugador
2. Listar jugadores
3. Buscar jugador por ID
4. Actualizar jugador
5. Eliminar jugador
6. Volver al menú principal
Selecciona una opción: 2

--- Lista de jugadores activos ---
+------+--------------+--------+----------+
|   ID | Nombre       |   Edad | Equipo   |
+======+==============+========+==========+
|    1 | Carolina Jr  |     22 | CarolFC  |
+------+--------------+--------+----------+
|    2 | Kike Ronaldo |     22 | KikeFC   |
+------+--------------+--------+----------+
```

```
===== MENÚ DE JUGADORES =====
1. Crear jugador
2. Listar jugadores
3. Buscar jugador por ID
4. Actualizar jugador
5. Eliminar jugador
6. Volver al menú principal
Selecciona una opción: 3

--- Buscar jugador por ID ---
Introduce el ID del jugador: 2
+------+--------------+--------+----------+----------+
|   ID | Nombre       |   Edad | Equipo   | Activo   |
+======+==============+========+==========+==========+
|    2 | Kike Ronaldo |     22 | KikeFC   | True     |
+------+--------------+--------+----------+----------+

===== MENÚ DE JUGADORES =====
1. Crear jugador
2. Listar jugadores
3. Buscar jugador por ID
4. Actualizar jugador
5. Eliminar jugador
6. Volver al menú principal
Selecciona una opción: 4

--- Actualizar datos del jugador ---
Introduce el ID del jugador a modificar: 1
Nota: si dejas un campo vacío, no se cambiará.
Nuevo nombre: Lorena Messi
Nueva edad: 18
Jugador actualizado correctamente.


===== MENÚ DE JUGADORES =====
1. Crear jugador
2. Listar jugadores
3. Buscar jugador por ID
4. Actualizar jugador
5. Eliminar jugador
6. Volver al menú principal
Selecciona una opción: 2

--- Lista de jugadores activos ---
+------+--------------+--------+----------+
|   ID | Nombre       |   Edad | Equipo   |
+======+==============+========+==========+
|    1 | Lorena Messi |     18 | CarolFC  |
+------+--------------+--------+----------+
|    2 | Kike Ronaldo |     22 | KikeFC   |
+------+--------------+--------+----------+
```

```
===== MENÚ DE JUGADORES =====
1. Crear jugador
2. Listar jugadores
3. Buscar jugador por ID
4. Actualizar jugador
5. Eliminar jugador
6. Volver al menú principal
Selecciona una opción: 5

--- Eliminar jugador ---
Introduce el ID del jugador a eliminar: 1
El jugador ha sido eliminado correctamente.


===== MENÚ DE JUGADORES =====
1. Crear jugador
2. Listar jugadores
3. Buscar jugador por ID
4. Actualizar jugador
5. Eliminar jugador
6. Volver al menú principal
Selecciona una opción: 2

--- Lista de jugadores activos ---
+------+--------------+--------+----------+
|   ID | Nombre       |   Edad | Equipo   |
+======+==============+========+==========+
|    2 | Kike Ronaldo |     22 | KikeFC   |
+------+--------------+--------+----------+

===== MENÚ DE JUGADORES =====
1. Crear jugador
2. Listar jugadores
3. Buscar jugador por ID
4. Actualizar jugador
5. Eliminar jugador
6. Volver al menú principal
Selecciona una opción:
Opción no válida.

===== MENÚ DE JUGADORES =====
1. Crear jugador
2. Listar jugadores
3. Buscar jugador por ID
4. Actualizar jugador
5. Eliminar jugador
6. Volver al menú principal
Selecciona una opción: 6
Volviendo al menú principal...
```

MÓDULO 3: Calendario

CÓDIGO:

```
#CALENDARIO

from tabulate import tabulate
from datetime import datetime
from rich.console import Console
console = Console()

#FUNCIONES
#MENU DEL CALENDARIO
def menu_calendario(partidos, equipos):
  opcion = 0
  while opcion != 5:
    print("\n--- MENÚ DE CALENDARIO ---")
    print("1. Crear partido")
    print("2. Listar partidos")
    print("3. Reprogramar partido")
    print("4. Eliminar partido")
    print("5. Volver al menú principal")

    try:
      opcion = int(input("Selecciona una opción: "))
    except:
      opcion = 0

    if opcion == 1:
      partidos = crear_partido(partidos, equipos)
    elif opcion == 2:
      listar_partidos(partidos, equipos)
    elif opcion == 3:
      partidos = reprogramar_partido(partidos)
    elif opcion == 4:
      partidos = eliminar_partido(partidos)
    elif opcion == 5:
      print("Volviendo al menú principal...")
    else:
      print("Opción no válida.")
  return partidos

#CREAR PARTIDO
def crear_partido(partidos, equipos):
  print("\n--- Crear partido ---")

  #MOSTRAR EQUIPOS ACTIVOS
  activos = []
  for e in equipos:
```

```
    if e["activo"]:
        activos.append([e["id"], e["nombre"]])
    if len(activos) < 2:
        print("No hay suficientes equipos ACTIVOS para crear un partido.")
        return partidos
    print(tabulate(activos, headers=["ID", "Equipo"], tablefmt="grid"))

    try:
        jornada = int(input("Jornada: "))
        if jornada < 1:
            print("La jornada debe ser al menos 1.")
            return partidos
        local_id=int(input("ID del equipo local: "))
        visitante_id=int(input("ID del equipo visitante: "))
    except:
        print("Entrada inválida.")
        return partidos

    if local_id ==visitante_id:
        print("¡Un equipo no puede enfrentarse a sí mismo!")
        return partidos

    #VERIFICAR QUE AMBOS EQUIPOS ESTEN ACTIVOS
    ids_activos = [e["id"] for e in equipos if e["activo"]]
    if local_id not in ids_activos or visitante_id not in ids_activos:
        print("Uno o los dos equipos no existen o están inactivos.")
        return partidos

    #PARA NO DUPLICADOS(SACADO DE UN CHICO DE REDDIT)
    for p in partidos:
        if p["jornada"] == jornada and (
            (p["local_id"] == local_id and p["visitante_id"] == visitante_id)
or
            (p["local_id"] == visitante_id and p["visitante_id"] == local_id)
        ):
            print("Ya existe un partido entre estos equipos en esta jornada.")
            return partidos
    #USO DE DATETIME
    fecha = input("Fecha (YYYY-MM-DD): ").strip()
    hora = input("Hora (HH:MM): ").strip()

    #GENERAR ID ÚNICO PARA EL ENCUENTRO
    nuevo_id = 1
    for p in partidos:
        if p["id"] >= nuevo_id:
            nuevo_id = p["id"] + 1

    partido = {
        "id": nuevo_id,
```

```
        "jornada": jornada,
        "local_id": local_id,
        "visitante_id": visitante_id,
        "fecha": fecha,
        "hora": hora,
        "jugado": False,
        "resultado": None  #(golesLocal, golesVisitante)
    }
    partidos.append(partido)
    print("Partido creado correctamente.\n")
    return partidos

#LISTAR PARTIDOS
def listar_partidos(partidos, equipos):
    print("\n--- Calendario de partidos ---")
    if len(partidos) ==0:
        print("No hay partidos programados.")
        return
    try:
        filtro=input("¿Deseas filtrar por jornada? (S/N): ").strip().upper()
    except:
        filtro="N"

    if filtro=="S":
        try:
            jornada = int(input("Introduce el número de jornada: "))
        except:
            print("Entrada inválida.")
            return
        seleccionados = []
        for p in partidos:
            if p["jornada"]==jornada:
                seleccionados.append(p)
        if len(seleccionados)==0:
            print("No hay partidos en esa jornada.")
            return
        mostrar_tabla(seleccionados, equipos)
    else:
        mostrar_tabla(partidos, equipos)

#REPROGRAMAR O EDITAR PARTIDO
def reprogramar_partido(partidos):
    print("\n--- Reprogramar partido ---")
    try:
        id_partido=int(input("Introduce el ID del partido: "))
    except:
        print("ID inválido.")
        return partidos
```

```
    partido=None
    for p in partidos:
        if p["id"]==id_partido:
            partido=p
    if partido ==None:
        print("Partido no encontrado.")
        return partidos
    #FILTRO PARA PARTIDOS QUE YA HAN SIDO JUGADOS(SON NO EDITABLES)
    if partido["jugado"]:
        print("No se puede reprogramar un partido ya jugado.")
        return partidos
    #USO DATETIME
    nueva_fecha = input("Nueva fecha (YYYY-MM-DD): ").strip()
    nueva_hora = input("Nueva hora (HH:MM): ").strip()
    try:
        datetime.strptime(nueva_fecha, "%Y-%m-%d")
        datetime.strptime(nueva_hora, "%H:%M")
    except:
        print("Formato de fecha/hora no válido.")
        return partidos

    partido["fecha"] = nueva_fecha
    partido["hora"] = nueva_hora
    print("Partido reprogramado correctamente.\n")
    return partidos

#ELIMINAR PARTIDO
def eliminar_partido(partidos):
    print("\n--- Eliminar partido ---")
    try:
        id_eliminar=int(input("Introduce el ID del partido a eliminar: "))
    except:
        print("ID inválido.")
        return partidos

    partido=None
    for p in partidos:
        if p["id"]==id_eliminar:
            partido=p
    if partido==None:
        print("Partido no encontrado.")
        return partidos
    #FILTRO PARA NO ELIMINAR UN PARTIDO YA JUGADO
    if partido["jugado"]:
        print("No se puede eliminar un partido que ya se ha jugado.")
        return partidos

    partidos.remove(partido)
    print("Partido eliminado correctamente.\n")
```

```
  return partidos

#FUNCION DE UTILIDAD: MOSTRAR LA TABLA
def mostrar_tabla(lista, equipos):
  filas=[]
  for p in lista:
    local=visitante=""
    for e in equipos:
      if e["id"] == p["local_id"]:
        local=e["nombre"]
      if e["id"] == p["visitante_id"]:
        visitante=e["nombre"]
    resultado="-"
    #USO DE LIBRERIA RICH(COLORES EN RESULTADOS)
    if p["resultado"] is not None:
      goles_local, goles_visitante = p["resultado"]
      if goles_local > goles_visitante:
        local = f"[green]{local}[/green]"
        visitante = f"[red]{visitante}[/red]"
      elif goles_visitante > goles_local:
        local = f"[red]{local}[/red]"
        visitante = f"[green]{visitante}[/green]"
      resultado = f"{goles_local} - {goles_visitante}"
    filas.append([
      p["id"], p["jornada"], local, visitante, p["fecha"], p["hora"],
      "Sí" if p["jugado"] else "No", resultado
    ])
  #CONSOLE.PRINT PARA MOSTRAR RESULTADO CON RICH
  console.print(tabulate(filas, headers=["ID", "Jornada", "Local",
"Visitante", "Fecha", "Hora", "Jugado", "Resultado"], tablefmt="grid"))
```

COMPROBACIÓN:

```
--- MENÚ DE CALENDARIO ---
1. Crear partido
2. Listar partidos
3. Reprogramar partido
4. Eliminar partido
5. Volver al menú principal
Selecciona una opción: 1

--- Crear partido ---
+------+-----------+
|   ID | Equipo    |
+======+===========+
|    1 | KikeFC    |
+------+-----------+
|    2 | CarolFC   |
+------+-----------+
|    3 | ArturoFC  |
+------+-----------+
|    4 | RobertoFC |
+------+-----------+
Jornada: 1
ID del equipo local: 1
ID del equipo visitante: 2
Fecha (YYYY-MM-DD): 2025-10-20
Hora (HH:MM): 15:30
Partido creado correctamente.
```

He añadido 3 partidos mas y una jornada extra:

```
--- MENÚ DE CALENDARIO ---
1. Crear partido
2. Listar partidos
3. Reprogramar partido
4. Eliminar partido
5. Volver al menú principal
Selecciona una opción: 2

--- Calendario de partidos ---
¿Deseas filtrar por jornada? (S/N): N
+------+-----------+----------+-------------+------------+--------+----------+-------------+
|   ID |   Jornada | Local    | Visitante   | Fecha      | Hora   | Jugado   | Resultado   |
+======+===========+==========+=============+============+========+==========+=============+
|    1 |         1 | KikeFC   | CarolFC     | 2025-10-20 | 15:30  | No       | -           |
+------+-----------+----------+-------------+------------+--------+----------+-------------+
|    2 |         1 | ArturoFC | RobertoFC   | 2025-10-21 | 17:30  | No       | -           |
+------+-----------+----------+-------------+------------+--------+----------+-------------+
|    3 |         2 | CarolFC  | ArturoFC    | 2025-11-04 | 14:30  | No       | -           |
+------+-----------+----------+-------------+------------+--------+----------+-------------+
|    4 |         2 | KikeFC   | RobertoFC   | 2025-11-05 | 16:30  | No       | -           |
+------+-----------+----------+-------------+------------+--------+----------+-------------+

--- MENÚ DE CALENDARIO ---
1. Crear partido
2. Listar partidos
3. Reprogramar partido
4. Eliminar partido
5. Volver al menú principal
Selecciona una opción: 2

--- Calendario de partidos ---
¿Deseas filtrar por jornada? (S/N): S
Introduce el número de jornada: 1
+------+-----------+----------+-------------+------------+--------+----------+-------------+
|   ID |   Jornada | Local    | Visitante   | Fecha      | Hora   | Jugado   | Resultado   |
+======+===========+==========+=============+============+========+==========+=============+
|    1 |         1 | KikeFC   | CarolFC     | 2025-10-20 | 15:30  | No       | -           |
+------+-----------+----------+-------------+------------+--------+----------+-------------+
|    2 |         1 | ArturoFC | RobertoFC   | 2025-10-21 | 17:30  | No       | -           |
+------+-----------+----------+-------------+------------+--------+----------+-------------+

--- MENÚ DE CALENDARIO ---
1. Crear partido
2. Listar partidos
3. Reprogramar partido
4. Eliminar partido
5. Volver al menú principal
Selecciona una opción: 3

--- Reprogramar partido ---
Introduce el ID del partido: 1
Nueva fecha (YYYY-MM-DD): 2025-10-23
Nueva hora (HH:MM): 12:30
Partido reprogramado correctamente.
```

```
--- MENÚ DE CALENDARIO ---
1. Crear partido
2. Listar partidos
3. Reprogramar partido
4. Eliminar partido
5. Volver al menú principal
Selecciona una opción: 2

--- Calendario de partidos ---
¿Deseas filtrar por jornada? (S/N): N
+------+-----------+----------+-------------+------------+--------+----------+-------------+
|   ID |   Jornada | Local    | Visitante   | Fecha      | Hora   | Jugado   | Resultado   |
+======+===========+==========+=============+============+========+==========+=============+
|    1 |         1 | KikeFC   | CarolFC     | 2025-10-23 | 12:30  | No       | -           |
+------+-----------+----------+-------------+------------+--------+----------+-------------+
|    2 |         1 | ArturoFC | RobertoFC   | 2025-10-21 | 17:30  | No       | -           |
+------+-----------+----------+-------------+------------+--------+----------+-------------+
|    3 |         2 | CarolFC  | ArturoFC    | 2025-11-04 | 14:30  | No       | -           |
+------+-----------+----------+-------------+------------+--------+----------+-------------+
|    4 |         2 | KikeFC   | RobertoFC   | 2025-11-05 | 16:30  | No       | -           |
+------+-----------+----------+-------------+------------+--------+----------+-------------+

--- MENÚ DE CALENDARIO ---
1. Crear partido
2. Listar partidos
3. Reprogramar partido
4. Eliminar partido
5. Volver al menú principal
Selecciona una opción: 4

--- Eliminar partido ---
Introduce el ID del partido a eliminar: 3
Partido eliminado correctamente.


--- MENÚ DE CALENDARIO ---
1. Crear partido
2. Listar partidos
3. Reprogramar partido
4. Eliminar partido
5. Volver al menú principal
Selecciona una opción: 2

--- Calendario de partidos ---
¿Deseas filtrar por jornada? (S/N): N
+------+-----------+----------+-------------+------------+--------+----------+-------------+
|   ID |   Jornada | Local    | Visitante   | Fecha      | Hora   | Jugado   | Resultado   |
+======+===========+==========+=============+============+========+==========+=============+
|    1 |         1 | KikeFC   | CarolFC     | 2025-10-23 | 12:30  | No       | -           |
+------+-----------+----------+-------------+------------+--------+----------+-------------+
|    2 |         1 | ArturoFC | RobertoFC   | 2025-10-21 | 17:30  | No       | -           |
+------+-----------+----------+-------------+------------+--------+----------+-------------+
|    4 |         2 | KikeFC   | RobertoFC   | 2025-11-05 | 16:30  | No       | -           |
+------+-----------+----------+-------------+------------+--------+----------+-------------+
```

MÓDULO 4: Ranking

CÓDIGO:

Menú principal final quedaria asi:

```
#MAIN

#IMPORTAR FUNCIONES Y LIBRERIAS
import tabulate
import rich
from equipos import menu_equipos
from jugadores import menu_jugadores
from calendario import menu_calendario
from ranking import menu_resultados

#MENU PRINCIPAL
def menu_principal():
  equipos=[]
  jugadores=[]
  partidos=[]
  opcion = 0
  while opcion != 5:
    print("---- MENÚ PRINCIPAL ----")
    print("1. Gestión de equipos")
    print("2. Gestión de jugadores")
    print("3. Calendarios de partidos")
    print("4. Resultados y clasificación")
    print("5. Salir")
    print("-------------------------")

    try:
      opcion = int(input("Selecciona una opción: "))
    except:
      opcion=0

    if opcion == 1:
      equipos=menu_equipos(equipos)
    elif opcion == 2:
      jugadores=menu_jugadores(jugadores, equipos)
    elif opcion == 3:
      partidos=menu_calendario(partidos, equipos)
    elif opcion == 4:
      partidos=menu_resultados(partidos, equipos)
    elif opcion == 5:
      print("Programa finalizado.")
    else:
      print("Opción no válida.")

#EJECUTAR PROGRAMA
```

```
if __name__ == "__main__":
  menu_principal()
```

ranking.py

```
#RANKING

from tabulate import tabulate

#FUNCIONES
#MENU PRINCIPAL RANKING
def menu_resultados(partidos, equipos):
  opcion = 0
  while opcion != 4:
    print("\n--- MENÚ DE RESULTADOS Y CLASIFICACIÓN ---")
    print("1. Registrar resultado")
    print("2. Mostrar clasificación general")
    print("3. Ver estadísticas por equipo")
    print("4. Volver al menú principal")
    try:
      opcion = int(input("Selecciona una opción: "))
    except:
      opcion=0
    if opcion == 1:
      partidos = registrar_resultado(partidos, equipos)
    elif opcion == 2:
      calcular_clasificacion(partidos, equipos)
    elif opcion == 3:
      estadisticas_equipo(partidos, equipos)
    elif opcion == 4:
      print("Volviendo al menú principal...")
    else:
      print("Opción no válida.")
  return partidos

#REGISTRAR RESULTADO
def registrar_resultado(partidos, equipos):
  print("\n--- Registrar resultado de un partido ---")
  #MOSTRAR !!SOLO!! PARTIDOS PENDIENTES
  pendientes = [p for p in partidos if not p["jugado"]]
  if len(pendientes) == 0:
    print("No hay partidos pendientes de resultado.")
    return partidos

  filas = []
  for p in pendientes:
    local=visitante= ""
    for e in equipos:
      if e["id"] == p["local_id"]:
```

```
            local=e["nombre"]
        if e["id"] == p["visitante_id"]:
            visitante=e["nombre"]
    filas.append([p["id"], p["jornada"], local, visitante, p["fecha"],
p["hora"]])
  print(tabulate(filas, headers=["ID", "Jornada", "Local", "Visitante",
"Fecha", "Hora"], tablefmt="grid"))

  try:
    id_partido=int(input("Introduce el ID del partido para registrar
resultado: "))
  except:
    print("ID inválido.")
    return partidos

  partido=None
  for p in partidos:
    if p["id"] == id_partido:
      partido=p

  if partido is None:
    print("Partido no encontrado.")
    return partidos

  if partido["jugado"]:
    print("Este partido ya tiene resultado registrado.")
    return partidos
  #GANA LOCAL O VISITANTE
  try:
    gL = int(input("Goles equipo local: "))
    gV = int(input("Goles equipo visitante: "))
  except:
    print("Entrada inválida. Usa números enteros pringao.")
    return partidos

  if gL < 0 or gV < 0:
    print("Los goles deben ser numeros enteros o cero.")
    return partidos

  partido["resultado"]= (gL, gV)
  partido["jugado"]=True
  print("El resultado ha sido registrado correctamente.\n")
  return partidos

#CLASIFICACIÓN GENERAL
def calcular_clasificacion(partidos, equipos):
  print("\n--- Clasificación general ---")
  #DICCIONARIO CON STATS
  tabla={}
```

```
  for e in equipos:
    if e["activo"]:
      tabla[e["id"]] = {
        "equipo": e["nombre"],
        "PJ": 0, "G": 0, "E": 0, "P": 0,
        "GF": 0, "GC": 0, "DG": 0, "PTS": 0
      }
  #BUCLE RECORRE LOS PARTIDOS JUGADOS
  for p in partidos:
    if p["jugado"] and p["resultado"] is not None:
      gL, gV = p["resultado"]
      local = p["local_id"]
      visitante = p["visitante_id"]

      if local in tabla and visitante in tabla:
        #PARTIDOS JUGADOS
        tabla[local]["PJ"] += 1
        tabla[visitante]["PJ"] += 1
        #GOLES
        tabla[local]["GF"] += gL
        tabla[local]["GC"] += gV
        tabla[visitante]["GF"] += gV
        tabla[visitante]["GC"] += gL
        #DIFERENCIA DE GOLES
        tabla[local]["DG"] = tabla[local]["GF"] - tabla[local]["GC"]
        tabla[visitante]["DG"] = tabla[visitante]["GF"] -
tabla[visitante]["GC"]
        #RESULTADO
        if gL > gV:
          tabla[local]["G"] += 1
          tabla[visitante]["P"] += 1
          tabla[local]["PTS"] += 3
        elif gV > gL:
          tabla[visitante]["G"] += 1
          tabla[local]["P"] += 1
          tabla[visitante]["PTS"] += 3
        else:
          tabla[local]["E"] += 1
          tabla[visitante]["E"] += 1
          tabla[local]["PTS"] += 1
          tabla[visitante]["PTS"] += 1
  #LISTA POR PUNTUAJE
  lista_ordenada=sorted(tabla.values(), key=lambda x: x["PTS"],
reverse=True)
  filas=[]
  for t in lista_ordenada:
    filas.append([
      t["equipo"], t["PJ"], t["G"], t["E"], t["P"],
      t["GF"], t["GC"], t["DG"], t["PTS"]
```

```
    ])
  print(tabulate(
    filas,
    headers=["Equipo", "PJ", "G", "E", "P", "GF", "GC", "DG", "PTS"],
    tablefmt="grid"
  ))

#ESTADÍSTICAS DE EQUIPO
def estadisticas_equipo(partidos, equipos):
  print("\n--- Estadísticas por equipo ---")
  try:
    id_equipo = int(input("Introduce el ID del equipo: "))
  except:
    print("Entrada inválida.")
    return
  equipo=None
  for e in equipos:
    if e["id"] == id_equipo:
      equipo=e
  if equipo is None:
    print("Equipo no encontrado.")
    return

  PJ = G = E = P = GF = GC = PTS = 0 #SUGERIDO POR COPILOT
  for p in partidos:
    if p["jugado"] and p["resultado"] is not None:
      gL, gV=p["resultado"]
      if p["local_id"] == id_equipo:
        PJ += 1
        GF += gL
        GC += gV
        if gL > gV:
          G += 1
          PTS += 3
        elif gV > gL:
          P += 1
        else:
          E += 1
          PTS += 1
      elif p["visitante_id"] == id_equipo:
        PJ += 1
        GF += gV
        GC += gL
        if gV > gL:
          G += 1
          PTS += 3
        elif gL > gV:
          P += 1
        else:
```

```
            E += 1
            PTS += 1
    DG = GF - GC
    filas = [[equipo["nombre"], PJ, G, E, P, GF, GC, DG, PTS]]
    print(tabulate(
        filas,
        headers=["Equipo", "PJ", "G", "E", "P", "GF", "GC", "DG", "PTS"],
        tablefmt="grid"
    ))
```

COMPROBACIÓN:

```
---- MENÚ PRINCIPAL ----
1. Gestión de equipos
2. Gestión de jugadores
3. Calendarios de partidos
4. Resultados y clasificación
5. Salir
-------------------------
Selecciona una opción: 4

--- MENÚ DE RESULTADOS Y CLASIFICACIÓN ---
1. Registrar resultado
2. Mostrar clasificación general
3. Ver estadísticas por equipo
4. Volver al menú principal
Selecciona una opción: 1

--- Registrar resultado de un partido ---
+------+-----------+-----------+-------------+------------+--------+
|   ID |   Jornada | Local     | Visitante   | Fecha      | Hora   |
+======+===========+===========+=============+============+========+
|    1 |         1 | KikeFC    | CarolFC     | 2025-11-22 | 13:00  |
+------+-----------+-----------+-------------+------------+--------+
|    2 |         1 | RobertoFC | EstrellaFC  | 2025-11-23 | 13:00  |
+------+-----------+-----------+-------------+------------+--------+
|    3 |         2 | KikeFC    | EstrellaFC  | 2025-11-30 | 12:30  |
+------+-----------+-----------+-------------+------------+--------+
|    4 |         2 | CarolFC   | RobertoFC   | 2025-11-31 | 12:30  |
+------+-----------+-----------+-------------+------------+--------+
Introduce el ID del partido para registrar resultado: 1
Goles equipo local: 4
Goles equipo visitante: 1
El resultado ha sido registrado correctamente.


--- MENÚ DE RESULTADOS Y CLASIFICACIÓN ---
1. Registrar resultado
2. Mostrar clasificación general
3. Ver estadísticas por equipo
4. Volver al menú principal
Selecciona una opción: 1

--- Registrar resultado de un partido ---
+------+-----------+-----------+-------------+------------+--------+
|   ID |   Jornada | Local     | Visitante   | Fecha      | Hora   |
+======+===========+===========+=============+============+========+
|    2 |         1 | RobertoFC | EstrellaFC  | 2025-11-23 | 13:00  |
+------+-----------+-----------+-------------+------------+--------+
|    3 |         2 | KikeFC    | EstrellaFC  | 2025-11-30 | 12:30  |
+------+-----------+-----------+-------------+------------+--------+
|    4 |         2 | CarolFC   | RobertoFC   | 2025-11-31 | 12:30  |
+------+-----------+-----------+-------------+------------+--------+
Introduce el ID del partido para registrar resultado: 1
Este partido ya tiene resultado registrado.
```

```
--- MENÚ DE RESULTADOS Y CLASIFICACIÓN ---
1. Registrar resultado
2. Mostrar clasificación general
3. Ver estadísticas por equipo
4. Volver al menú principal
Selecciona una opción: 2

--- Clasificación general ---
+------------+------+-----+-----+-----+------+------+------+-------+
| Equipo     |   PJ |   G |   E |   P |   GF |   GC |   DG |   PTS |
+============+======+=====+=====+=====+======+======+======+=======+
| KikeFC     |    2 |   1 |   1 |   0 |    5 |    2 |    3 |     4 |
+------------+------+-----+-----+-----+------+------+------+-------+
| EstrellaFC |    2 |   1 |   1 |   0 |    2 |    1 |    1 |     4 |
+------------+------+-----+-----+-----+------+------+------+-------+
| CarolFC    |    2 |   1 |   0 |   1 |    7 |    4 |    3 |     3 |
+------------+------+-----+-----+-----+------+------+------+-------+
| RobertoFC  |    2 |   0 |   0 |   2 |    0 |    7 |   -7 |     0 |
+------------+------+-----+-----+-----+------+------+------+-------+

--- MENÚ DE RESULTADOS Y CLASIFICACIÓN ---
1. Registrar resultado
2. Mostrar clasificación general
3. Ver estadísticas por equipo
4. Volver al menú principal
Selecciona una opción: 3

--- Estadísticas por equipo ---
Introduce el ID del equipo: 1
+----------+------+-----+-----+-----+------+------+------+-------+
| Equipo   |   PJ |   G |   E |   P |   GF |   GC |   DG |   PTS |
+==========+======+=====+=====+=====+======+======+======+=======+
| KikeFC   |    2 |   1 |   1 |   0 |    5 |    2 |    3 |     4 |
+----------+------+-----+-----+-----+------+------+------+-------+

--- MENÚ DE RESULTADOS Y CLASIFICACIÓN ---
1. Registrar resultado
2. Mostrar clasificación general
3. Ver estadísticas por equipo
4. Volver al menú principal
Selecciona una opción: 3

--- Estadísticas por equipo ---
Introduce el ID del equipo: 2
+----------+------+-----+-----+-----+------+------+------+-------+
| Equipo   |   PJ |   G |   E |   P |   GF |   GC |   DG |   PTS |
+==========+======+=====+=====+=====+======+======+======+=======+
| CarolFC  |    2 |   1 |   0 |   1 |    7 |    4 |    3 |     3 |
+----------+------+-----+-----+-----+------+------+------+-------+
```

```
--- MENÚ DE RESULTADOS Y CLASIFICACIÓN ---
1. Registrar resultado
2. Mostrar clasificación general
3. Ver estadísticas por equipo
4. Volver al menú principal
Selecciona una opción: 3

--- Estadísticas por equipo ---
Introduce el ID del equipo: 2
+----------+------+-----+-----+-----+------+------+------+-------+
| Equipo   |   PJ |   G |   E |   P |   GF |   GC |   DG |   PTS |
+==========+======+=====+=====+=====+======+======+======+=======+
| CarolFC  |    2 |   1 |   0 |   1 |    7 |    4 |    3 |     3 |
+----------+------+-----+-----+-----+------+------+------+-------+

--- MENÚ DE RESULTADOS Y CLASIFICACIÓN ---
1. Registrar resultado
2. Mostrar clasificación general
3. Ver estadísticas por equipo
4. Volver al menú principal
Selecciona una opción: 4
Volviendo al menú principal...
---- MENÚ PRINCIPAL ----
1. Gestión de equipos
2. Gestión de jugadores
3. Calendarios de partidos
4. Resultados y clasificación
5. Salir
-------------------------
Selecciona una opción:
```