

# Examen 2023 2C - Tema 1

Se desea modelar un torneo de fútbol donde compiten equipos jugando partidos entre sí. Cada partido se juega en cierta fecha, con un equipo local y otro visitante, y tiene un resultado asociado que indica los goles que convirtió cada equipo. Si ambos equipos convirtieron la misma cantidad de goles, el partido resulta en empate, de lo contrario gana el que convierte más goles. Cuando un equipo gana un partido recibe 3 puntos en el torneo, si empata recibe 1 punto y si pierde no recibe puntos. Cada equipo se identifica unívocamente con su nombre y tiene también asociado una cantidad de fans.

Se solicita realizar lo siguiente:

- a) Implementar todas las clases necesarias que se describen en el enunciado, incluyendo getters y setters relevantes para resolver las funcionalidades solicitadas. Tip: Para definir la fecha de un partido se puede utilizar la clase nativa `LocalDate` (<https://docs.oracle.com/javase/8/docs/api/java/time/LocalDate.html>).

- b) Implementar la siguiente funcionalidad:

- 1) Crear un **torneo vacío**.
- 2) **Agregar equipos** al torneo.  
Dado un nombre y cantidad de fans, agregar el equipo al torneo.
- 3) **Cargar partidos** jugados en el torneo.  
Dado los nombres de un equipo local, uno visitante, una fecha y un resultado (goles locales, goles visitantes), agregar el partido al torneo.
- 4) Mostrar los **partidos jugados en una fecha**.  
Dada una fecha, mostrar en consola todos los partidos jugados en esa fecha. Cada partido debe mostrar la fecha, nombre de equipos (local y visitante) y resultado.
- 5) Mostrar **tabla de posiciones**.  
Generar una vista tabular en consola (no importa que quede perfectamente alineado) de las posiciones actuales de los equipos en el torneo (dependen de los partidos jugados y cargados). Las filas representan cada equipo y las columnas a mostrar son:
  - nombre de equipo
  - cantidad de partidos jugados
  - puntos en el torneo (ver enunciado cómo se calculan)
  - cantidad de partidos ganados
  - cantidad de partidos empatados
  - cantidad de partidos perdidos
  - cantidad de goles convertidos
  - cantidad de goles recibidos
  - diferencia de gol (goles convertidos - goles recibidos)

**Es deseable** que los equipos aparezcan **ordenados por puntos obtenidos**, de forma que primero aparezcan los que tienen más puntos. Si dos o más equipos coinciden en puntos, mostrar antes el que tenga **mayor diferencia de gol**.

**Forma de Entrega:** Archivo zip conteniendo los archivos de código fuente (\*.java).

**Criterios mínimos de aprobación:**

- No deben existir errores de compilación.
- No deben existir errores en tiempo de ejecución (excepciones no tratadas o no deseadas).
- Implementar la estructura básica de la solución contemplando el POO.
- Completar correctamente la funcionalidad requerida de los puntos b.1 a b.4.

## Anexo 1: Ejemplo de Main

Se puede utilizar el siguiente método main para probar el ejercicio, donde ya hay un escenario de equipos y partidos para verificar la funcionalidad.

```
public static void main(String[] args) {
    Torneo torneo = new Torneo();

    // Carga de equipos
    torneo.cargarEquipo("Team A", 100);
    torneo.cargarEquipo("Team B", 200);
    torneo.cargarEquipo("Team C", 300);
    torneo.cargarEquipo("Team D", 400);
    torneo.cargarEquipo("Team E", 500);
    torneo.cargarEquipo("Team F", 600);
    torneo.cargarEquipo("Team G", 700);
    torneo.cargarEquipo("Team H", 800);

    // Fecha 1 - 4/11
    torneo.cargarPartido("Team A", "Team B", LocalDate.of(2023, 11, 4), 2, 0);
    torneo.cargarPartido("Team C", "Team D", LocalDate.of(2023, 11, 4), 2, 1);
    torneo.cargarPartido("Team E", "Team F", LocalDate.of(2023, 11, 4), 2, 2);
    torneo.cargarPartido("Team G", "Team H", LocalDate.of(2023, 11, 4), 2, 3);

    // Fecha 2 - 12/11
    torneo.cargarPartido("Team B", "Team A", LocalDate.of(2023, 11, 12), 1, 1);
    torneo.cargarPartido("Team C", "Team D", LocalDate.of(2023, 11, 12), 3, 0);
    torneo.cargarPartido("Team F", "Team E", LocalDate.of(2023, 11, 12), 1, 2);
    torneo.cargarPartido("Team H", "Team G", LocalDate.of(2023, 11, 12), 0, 0);

    // Punto b.4
    torneo.mostrarPartidosFecha(LocalDate.of(2023, 11, 12));

    // Punto b.5
    torneo.mostrarTabla();
}
```

```
}
```

## Anexo 2: Ejemplo de salida de consola

Dado el ejemplo del main previo, se podría obtener una salida similar a la siguiente. No es necesario se cumpla con el formato propuesto, simplemente se adjunta para verificar los datos del torneo.

### **mostrarPartidosFecha**

Fecha: 2023-11-12, Team B (1) - Team A (1)

Fecha: 2023-11-12, Team C (3) - Team D (0)

Fecha: 2023-11-12, Team F (1) - Team E (2)

Fecha: 2023-11-12, Team H (0) - Team G (0)

### **mostrarTabla**

Equipo	Ju	Pu	Ga	Em	Pe	GF	GC	DG
--------	----	----	----	----	----	----	----	----

Team C	2	6	2	0	0	5	1	4
--------	---	---	---	---	---	---	---	---

Team A	2	4	1	1	0	3	1	2
--------	---	---	---	---	---	---	---	---

Team H	2	4	1	1	0	3	2	1
--------	---	---	---	---	---	---	---	---

Team E	2	4	1	1	0	4	3	1
--------	---	---	---	---	---	---	---	---

Team F	2	1	0	1	1	3	4	-1
--------	---	---	---	---	---	---	---	----

Team G	2	1	0	1	1	2	3	-1
--------	---	---	---	---	---	---	---	----

Team B	2	1	0	1	1	1	3	-2
--------	---	---	---	---	---	---	---	----

Team D	2	0	0	0	2	1	5	-4
--------	---	---	---	---	---	---	---	----