

# Programación

## Objetos

## Práctica 2.c

Se requiere desarrollar un programa en Java que gestione una liga deportiva. El programa deberá permitir gestionar una liga de cualquier deporte de equipo (fútbol, baloncesto, esports, etc.). Cuando creamos una liga es necesario que indiquemos el nombre de la liga y el deporte.

El programa nos permitirá tener un registro de los equipos y sus jugadores. Además deberá permitir crear partidos donde se registrarán los resultados. Las funcionalidades que se necesitan son las siguientes:

### Equipo

Será necesario almacenar los siguientes datos:

- **ID:** número entero que identifica el equipo.
- **Nombre:** Cadena de texto con el nombre del equipo
- **Presidente:** Cadena de texto con el nombre del presidente
- **Fecha de fundación:** Objeto de tipo fecha que almacena la fecha de fundación del equipo.
- **Lista de jugadores:** Listado de jugadores pertenecientes al equipo.
- **Partidos jugados:** Listado de partidos jugados.

Además el equipo dispondrá de las siguientes funcionalidades:

- **ficharJugador:** permite añadir un jugador a la lista de jugadores.
- **eliminarJugador:** permite eliminar un jugador de la lista de jugadores.
- **getPartidosGanados:** calcula y retorna el número de partidos ganados.
- **getPartidosEmpatados:** calcula y retorna el número de partidos empatados.
- **getPartidosPerdidos:** calcula y retorna el número de partidos perdidos.

### Jugador

Será necesario almacenar los siguientes datos:

- **NIF:** cadena que almacena el NIF del jugador.
- **Nombre:** cadena de texto que almacena el nombre del jugador.
- **Fecha de nacimiento:** Objeto de tipo fecha que guarda la fecha de nacimiento.
- **Posición:** Cadena que guarda la posición de juego del jugador. (Ejemplo fútbol: delantero, defensa, portero, etc.; Ejemplo esports: support, mid, jungla, etc.)
- **Equipo:** Referencia a un objeto de tipo Equipo.

### Partido

Será necesario almacenar los siguientes datos:

- **Fecha:** objeto de tipo fecha que guarda la fecha del partido.
- **Equipo local:** referencia al equipo local.
- **Equipo visitante:** referencia al equipo visitante.
- **Puntuación equipo local:** número que guarda la puntuación del equipo local.
- **Puntuación equipo visitante:** número que guarda la puntuación del equipo visitante.

Además el equipo dispondrá de las siguientes funcionalidades:

- **getResultado:** esta función retorna un 1 en caso de que el equipo local haya ganado, un 2 en caso de que el equipo visitante haya ganado y un 0 en caso de empate. Debe calcularlo a partir de las puntuaciones del partido.

La gestión de la liga dispondrá de las siguientes funcionalidades:

### crearEquipo ( ID, Nombre, Presidente , FechaFundacion )

Este método permitirá crear un nuevo equipo y añadirlo al listado de equipos de la liga. Recibirá como parámetros los indicados anteriormente. Además, inicializará el equipo sin jugadores y sin partidos jugados. Este método debe comprobar si el equipo existe, en caso de que ya exista la función retornará false. Si se ha podido incluir el equipo la función retornará true.

### crearJugador ( NIF, Nombre , Fecha de Nacimiento , Posición)

Este método permitirá crear un nuevo jugador y añadirlo al listado de jugadores de la liga. Recibirá como parámetros los indicados anteriormente. El equipo se inicializará a null, indicando que aún no dispone de equipo. Este método debe comprobar si el jugador existe, en caso de que ya exista la función retornará false. Si se ha podido incluir el jugador la función retornará true.

### ficharJugador( ID , NIF)

Esta funcionalidad permite que un equipo fiche a un jugador. Recibe como parámetros el id del equipo y el NIF del jugador. Se encargara de añadir el jugador al listado de jugadores del equipo y asignar el equipo al jugador. Se deberá comprobar que tanto el equipo como el jugador existen. Además, se debe comprobar que el jugador aún no esta fichado por ningún equipo (valor null). Si el jugador ya estuviera fichado, será necesario eliminar el jugador del equipo antiguo y asignarlo al nuevo. La función retornará los siguientes valores:

Valor devuelto	Significado
0	Sin errores
-1	El ID del equipo no existe
-2	El NIF del jugador no existe

### mostrarJugadores

Esta funcionalidad se encargará de mostrar por pantalla el listado completo de jugadores, con todos sus atributos (NIF, Nombre, Fecha de nacimiento, Posición, Equipo).

### mostrarEquipos

Esta funcionalidad se encargará de mostrar el listado completo equipos. Se mostrarán los campos ID, Nombre, Presidente, Fecha de fundación y el listado de todos sus jugadores. Para cada jugador se mostrara únicamente su nombre seguido de la posición.

### crearPartido( fecha, IDlocal, IDvisitante)

Este método se encargará de crear un nuevo partido. Recibirá como argumento la fecha y el ID de los dos equipos. Añadirá el partido al listado de partidos de la liga. También añadirá el partido a la lista de partidos de los dos equipos. La función debe comprobar que los equipos existan y que tengan jugadores. Además, debe comprobar que no exista ya un partido el mismo día y que son dos equipos diferentes. Retornará los siguientes valores:

Valor devuelto	Significado
0	Sin errores
-1	El ID del equipo local no existe
-2	El ID del equipo visitante no existe
-3	El equipo local no tiene jugadores
-4	El equipo visitante no tiene jugadores
-5	Ya existe un partido en esa fecha
-6	Los equipos deben ser distintos

### resultadoPartido( fecha, IDlocal, IDvisitante, puntosLocal , puntosVisitante)

Este método se encarga de asignar el resultado de un partido. Para ello, se pasarán como argumentos los indicados arriba. Esta función buscará el partido y asignará los valores indicados. Debe comprobar que el partido existe y que la puntuación es un valor mayor que 0. Retornara uno de los siguientes valores

Valor devuelto	Significado
0	Sin errores
-1	El partido no existe
-2	Las puntuaciones son invalidas

### mostrarPartidos

Esta función mostrara un listado de todos los partidos. Se visualizará la fecha, el nombre del equipo local, el nombre del equipo visitante y las puntuaciones. (Opcional: Se deben mostrar los partidos ordenados por fecha)

### mostrarResultadosLiga

Esta función mostrará un listado de los equipos ordenados por el numero de partidos ganados y perdidos. Para ello el alumno podrá utilizar cualquiera de las funciones implementadas anteriormente. Se mostrará el nombre del equipo, el numero de partidos jugados, ganados, perdidos y empatados.