1. **Análisis del problema y requisitos del sistema**

**• ¿Quiénes son los actores que interactúan con el sistema?**

Los administradores, que diseñan, crean, ejecutan y modifican el juego.

Los propios usuarios o jugadores que interactúan entre ellos y con la aplicación a través de los partidos, torneos, equipos y jugadores.

**• ¿Cuáles son las acciones que cada actor puede realizar?**

Los administradores pueden crear, eliminar o modificar tanto los torneos, como los equipos y jugadores, añadiendo o quitando jugadores de un equipo. Inscribir equipos a un torneo, diseñar un sistema de emparejamiento de equipos en los torneos, así como los premios a los equios ganadores.

Los jugadores podrán registrar equipos a un torneo, modificarlos, añadiendo o quitando jugadores, crear torneos, así como inscribir equipos en los torneos.

**• ¿Cómo se relacionan entre sí las entidades del sistema?**

Mediante asociación, agregación, composición, herencia y dependencia.

Ejemplos:

Un torneo se asocia con varios equipos inscritos en él.

Un equipo agrega jugadores.

Un torneo se compone de partidos.

El administrador puede heredar varios métodos que hace el usuario como ver equipos, crear torneos...

1. **Identificación de clases y relaciones**
2. **Identifica las clases principales en función de los casos de uso seleccionados.**

Las clases principales son Equipo, Jugador, torneo, Partidos, Resultados, Premio, Clasificación.

1. **Distingue las clases de Entidad, Control e Interfaz para mantener una arquitectura modular.**
2. **Define atributos y métodos para cada clase.**

Los atributos y métodos creados para cada una de las clases vienen en el diagrama de clases que se presenta a continuación. Algunos de las propiedades son nombre, id, jugadores, posición número.... y algunos métodos como inscribirEquipo, añadirJugador etc.

1. **Establece relaciones entre clases, asegurando la correcta representación de asociaciones, agregaciones y composiciones.**

En el diagrama de clases vienen dibujadas las líneas que representan las asociaciones, agregaciones y composiciones así como sus cardinalidades.

git@github.com:albertolara83/torneo\_UML.git

**ALBERTO LARA LÓPEZ**

**PRIMERO DAW**