**1. Análisis del Problema y Requisitos del Sistema**

**¿Qué son los actores que interactúan con el sistema?**

Son todas aquellas personas o no personas que van a interactuar con la aplicación. Se identifican en este ejemplo a dos actores. Administrador y jugador.

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

El Administrador puede🡪

- Registrar equipo

- Añadir jugadores a un equipo

- Consultar lista de equipos y jugadores

El jugador puede🡪

- Consultar lista de equipos

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

Un equipo tiene muchos jugadores

Cada jugador pertenece a un equipo

El administrador gestiona equipos y jugadores

**-Diagrama de Clases**

Diagrama

El contenido generado por IA puede ser incorrecto.

**Análisis del Problema y Requisitos del Sistema**

**Identifica las clases principales en función de los casos de uso seleccionados:**

- Registrar equipo

- Añadir jugadores a un equipo

- Consultar lista de equipos y jugadores

**Distingue las clases de Entidad, Control e Interfaz para mantener una arquitectura modular:**

- **Admin**(Entidad)

- **Jugador** (Entidad)

- **Equipo** (Entidad)

**Define atributos y métodos para cada clase:**

**Admin**

**Atributos:**

- `id: int`

- `nombre: String`

- `email: String`

**Métodos**

- `+ registrarEquipo(EquipoDTO equipoDTO)`

- `+ añadirJugadorAEquipo(JugadorDTO jugadorDTO)`

- `+ consultarEquiposYJugadores(): List<EquipoJugadorDTO>`

**Equipo**

**Atributos:**

- `id: int`

- `nombre: String`

- `jugadores: List<Jugador>`

**Métodos**

- `+ listarJugadores(): List<JugadorDTO>`

**Jugador**

**Atributos:**

- `id: int`

- `nombre: String`

- `posicion: String`

- `equipo: Equipo`

**Métodos**

- `+ verEquipo(): EquipoDTO`

---

**Establece relaciones entre clases**

- **Equipo → Jugador**: **Composición**

  Un equipo contiene jugadores. Si el equipo se elimina, los jugadores son también eliminados junto con el equipo.

Conclusiones🡪

Realizando este trabajo he podido practicar sobre la implementación de UML.