ENTORNOSDEDESARROLLO

Actividad 3: Diagramas UML

Nombre

Yeimmy Melissa Rodriguez Morales

1. Análisis del problema y requisitos del sistema

Antes de modelar, es fundamental comprender qué hace el sistema y qué se espera de él. Lee atentamente los requisitos y responde estas preguntas clave:

¿Quiénes son los actores que interactúan con el sistema?

Administrador, torneo y partido.

¿Cuáles son las acciones que cada actor puede realizar?

Administrador:

- Registrar Equipo: puede crear un nuevo equipo en el sistema.
- Añadir jugadores: agrega jugadores a su equipo.
- Consultar lista de equipos y jugadores: visualiza los equipos y sus jugadores.
- Inscribir equipo en un torneo: inscribe su equipo en un torneo existente.
- Comprobar si el equipo está inscrito: verifica si su equipo participa en un torneo.

Torneo:

- Crear Torneo: puede crear nuevos torneos.
- Comprobar existencia de torneo: puede verificar si un torneo ya está registrado.
- Generar emparejamientos: puede generar los partidos entre los equipos inscritos.
- Actualizar clasificación torneo: actualiza los puntajes o posiciones de los equipos.
- Registrar resultados: ingresa los resultados de los partidos jugados.
- Comprobar si el equipo está libre: verifica si su equipo aún puede inscribirse.

Partido:

- Actualizar clasificación torneo: Puede ver y actualizar los equipos que se clasifican.
- Registrar resultado: Que haya una constancia del resultado de ese partido.

¿Cómo se relacionan entre sí las entidades del sistema?

Relaciones principales entre clases:

- Un Torneo:
  - ✓ Contiene muchos Equipos inscritos.
  - ✓ Contiene muchos Partidos (partidos generados entre los equipos).
  - ✓ Se actualiza mediante Resultados (que modifican la clasificación).

- Un Equipo:
  - ✓ Está formado por muchos Jugadores.
  - ✓ Puede estar inscrito en uno o varios Torneos.
  - ✓ Participa en (partidos).

- Un Jugador:

  - ✓ Pertenece a un solo Equipo.

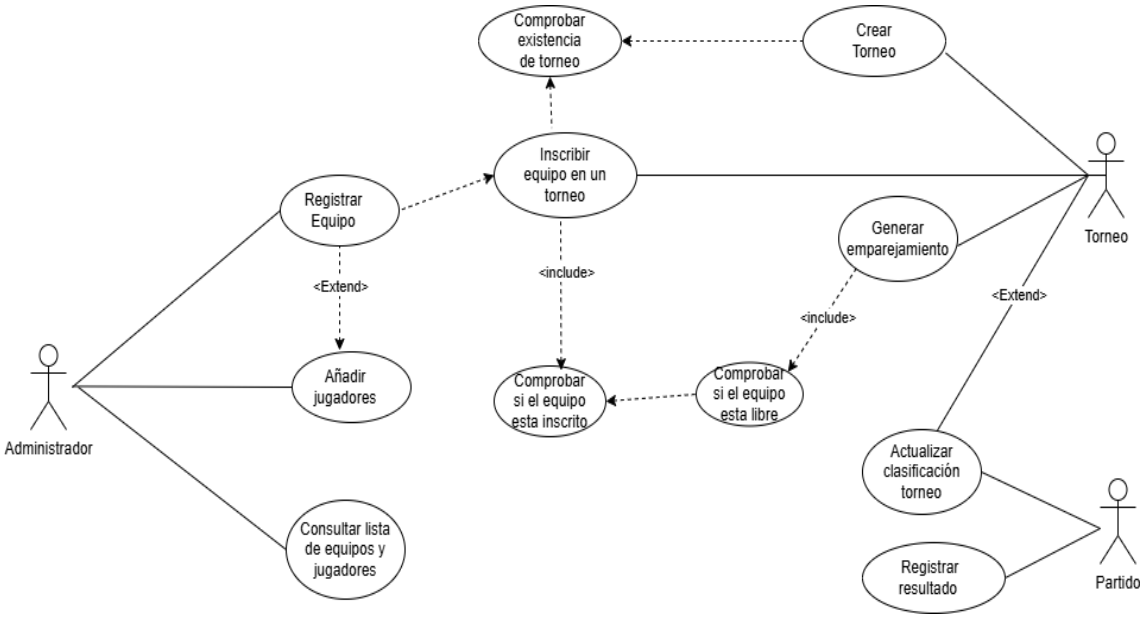
- Un partido:

  - ✓ Involucra a dos Equipos.
  - ✓ Se genera dentro de un Torneo.
  - ✓ Produce un Resultado.

- Un Resultado:

  - ✓ Está asociado a un partido.
  - ✓ Impacta la Clasificación del Torneo.

2. Identificación de los casos de uso y elaboración del diagrama, Ahora debes definir qué interacciones existen entre los actores y el sistema.



3. Identificación de clases y relaciones

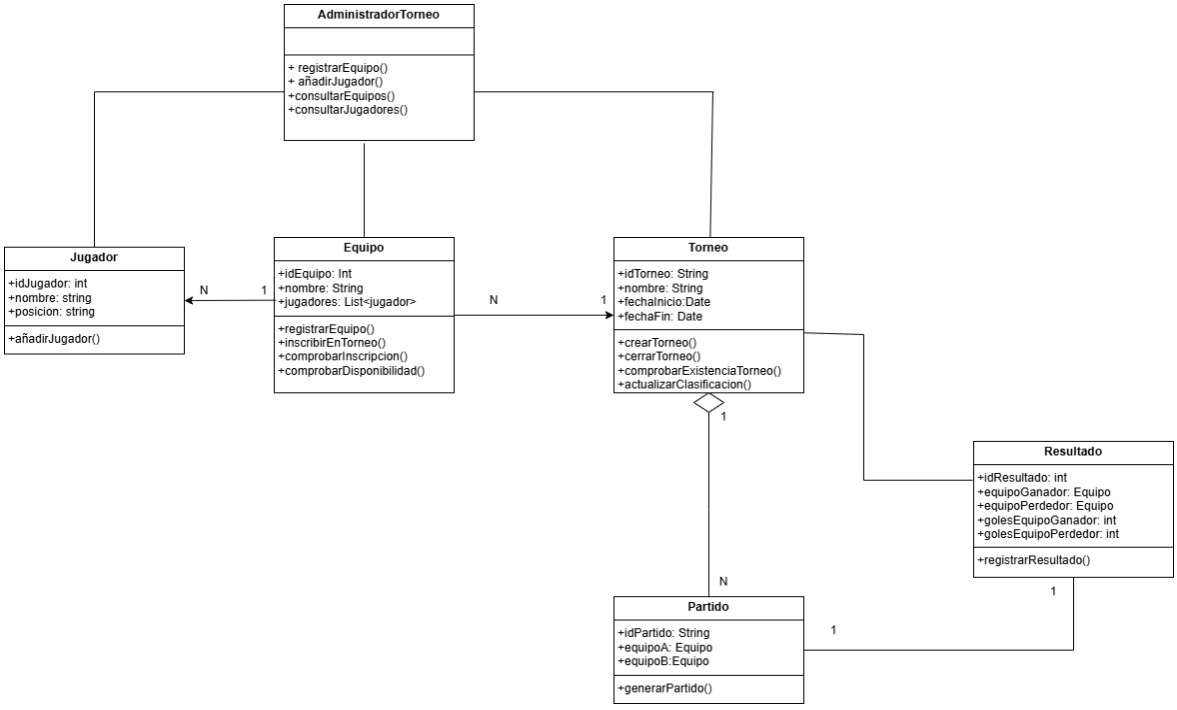
Una vez definidos los casos de uso, pasamos al modelo estructural del sistema.

- ✓ Identifica las clases principales en función de los casos de uso seleccionados.
- ✓ Distingue las clases de Entidad, Control e Interfaz para mantener una arquitectura modular.
- ✓ Define atributos y métodos para cada clase.
- ✓ Establece relaciones entre clases, asegurando la correcta representación de asociaciones, agregaciones y composiciones.

Clase	Tipo	Atributos	Métodos	Relaciones
Administrador	Control		registrarEquipo() añadirJugador() consultarEquipos() consultarJugadores()	Torneo equipo jugador
Torneo	Entidad	idTorneo: String nombre: String fechaInicio:Date fechaFin: Date	crearTorneo() cerrarTorneo() comprobarExistenciaTorneo() actualizarClasificacion()	Equipo Partido Resultado
Equipo	Entidad	idEquipo: Int nombre: String jugadores: List<jugador>	registrarEquipo() inscribirEnTorneo() comprobarInscripcion() comprobarDisponibilidad()	Torneo Jugador Administrador
Jugador	Entidad	idJugador: int nombre: string posicion: string	añadirJugador ()	Administrador, equipo
Partido	Entidad	idPartido: String equipoA: Equipo equipoB:Equipo	generarPartido()	Resultado, torneo

Resultado	Entidad	idResultado: int equipoGanador: Equipo equipoPerdedor: Equipo golesEquipoGanador: int golesEquipoPerdedor: int	registrarResultado()	Torneo, Partido

4. **Creación del diagrama de clases UML**



Enlace al repositorio compartido en GitHub

https://github.com/meli2304/torneo-esports-uml.git