

Ejercicio 4

Analisis y Diseño

1. ¿Qué debe hacer el programa? Haga una lista de las opciones que brindará al usuario en la interacción con su sistema:

- Registrar un jugador en el campeonato con todos sus datos relevantes, incluyendo nombre, país, faltas, goles directos y total de lanzamientos, y datos específicos para porteros (paradas efectivas y goles recibidos) o extremos (pases y asistencias efectivas).
- Mostrar todos los jugadores registrados en el campeonato.
- Obtener y mostrar los 3 mejores porteros en función de su efectividad.
- Contar la cantidad de extremos con más de un 85% de efectividad.

2. ¿Qué clases necesitará para dar solución a la situación planteada?

- Jugador (clase base)
- Portero (subclase de Jugador)
- Extremo (subclase de Jugador)
- Campeonato (para gestionar la lista de jugadores y realizar las operaciones requeridas)

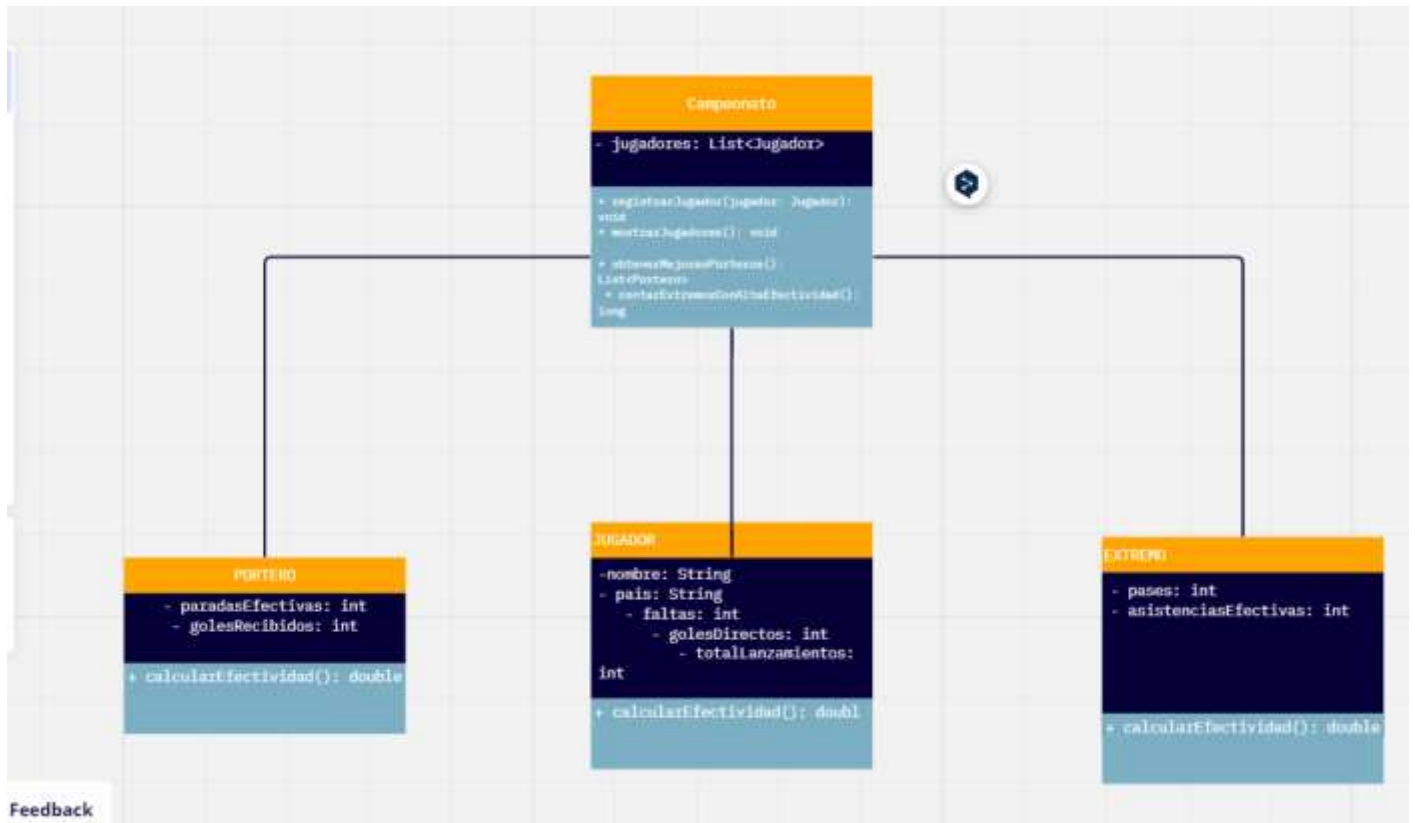
3. ¿Qué propiedades y métodos tendrá cada clase?

4. Jugador:

- Propiedades:
- nombre (String)
- país (String)
- faltas (int)

- golesDirectos (int)
- totalLanzamientos (int)
- **Métodos:**
- calcularEfectividad(): double (a implementar en subclases)
- getters y setters
- Portero (subclase de Jugador):
- Propiedades adicionales:
- paradasEfectivas (int)
- golesRecibidos (int)
- Métodos adicionales:
- calcularEfectividad() (sobrescrito)
- Extremo (subclase de Jugador):
- Propiedades adicionales:
- pases (int)
- asistenciasEfectivas (int)
- **Métodos adicionales:**
- calcularEfectividad() (sobrescrito)
- Campeonato:
- Propiedades:
- jugadores (Lista de Jugador)
- **Métodos:**
- registrarJugador(Jugador jugador)
- mostrarJugadores()
- obtenerMejoresPorteros(): List<Portero>
- contarExtremosConAltaEfectividad(): long
- main

UML



Feedback