

3. Identificación de clases y relaciones.

Clases :

- Equipo :

Tipo

-> Entidad

Atributos

-> codigoEquipo : String

-> nombreEquipo : String

-> jugadores : List<Jugador>

Metodos

-> agregarJugador

-> buscarJugador

-> Getters, Setters y ToString de los atributos

- Jugad@r :

Tipo

->Entidad

Atributos

-> codigoJugador: String

-> nombreJugador: String

-> edadJugador: Number

-> codigoEquipo: String

Metodos

-> Getters, Setters y ToString de los atributos

- Gestor equipo :

Tipo

-> Control

Atributos

-> equipos: List<Equipo>

Metodos

-> crearEquipo

-> buscarEquipo

-> modificarEquipo

-> eliminarEquipo

- Gestor jugador :

Tipo

-> Control

Atributos

-> equipos: List<Equipo>

-> jugadores: List<Jugador>

Metodos

-> añadirJugador

-> eliminarJugador

-> buscarJugador

- TestingsClass :

Tipo

--> Interfaz

Atributos

--> No necesita estrictamente atributos.

Metodos

--> mostrarEquipos

--> mostrarJugadores

Relaciones UML entre clases:

Composición

La clase Equipo se relaciona con Jugador mediante composición, ya que un equipo contiene una lista de jugadores y estos no tienen sentido por sí mismos dentro del sistema si el equipo desaparece. Es una relación fuerte: si se elimina el equipo, también podrían eliminarse sus jugadores.

Asociación

La clase GestorEquipo está asociada a Equipo, y GestorJugador está asociada tanto a Jugador como a Equipo. Estas relaciones representan conexiones directas y permanentes, donde los gestores mantienen listas de objetos y realizan operaciones sobre ellos, pero no son dueños de esas entidades.

Dependencia

La clase TestingClass depende de GestorEquipo y GestorJugador, ya que utiliza sus métodos para mostrar datos, pero no mantiene referencias permanentes ni los contiene.

4. Creación del diagrama de clases UML

