

## Ejercicio 4: Modelación con herencia

### Análisis:

El programa debe ser capaz de diferenciar los tipos de jugador (Portero y extremo) y tomar los datos de dicha clase para hacer los cálculos solicitados. El programa debe ser fácil de entender, amigable con el usuario además de proveer los datos solicitados correctamente; debe ser capaz de identificar qué tipo de jugador esta ingresando el usuario y crear una instancia de la clase respectiva.

Clase/herencia	Atributos	Métodos
Driver/padre		<p>TopGK(): Función para obtener los 3 mejores porteros en función de su efectividad.</p> <p>PlayerList(): Función para mostrar todos los jugadores registrados en el campeonato.</p> <p>Exteff(): Función para contar la cantidad de extremos con más del 85% de efectividad.</p>
Jugador/padre	<p>Nombre: Variable string protegida que contiene el nombre del jugador</p> <p>País: Variable string protegida que contiene el país de procedencia del jugador</p> <p>Faltas: Variable int protegida con el numero de faltas de un jugador</p> <p>DGol: Variable int protegida con el numero de goles directos de un jugador</p>	Getters y setters.

	Tlantzamientos: Variable int con el numero de lanzamientos de un jugador	
Portero/ Hereda de Jugador	<p>Eparadas: Variable int privada con el numero de paradas efectivas de un portero.</p> <p>Grecibidos: Variable int privada con el numero de goles recibidos de un portero.</p>	<p>Getters y setters.</p> <p>EfectividadGK(): Método que utiliza la formula dada en el documento del ejercicio para obtener el valor numérico de la efectividad del portero.</p>
Extremo/ Hereda de jugador	<p>Pases: Variable int con el numero de pases de un extremo</p> <p>Asistencias: Variable int con el numero de asistencias de un extremo.</p>	<p>Getters y setters.</p> <p>EfectividadEx(): Método que utiliza la formula dada en el documento del ejercicio para obtener el valor numérico de la efectividad del extremo</p>