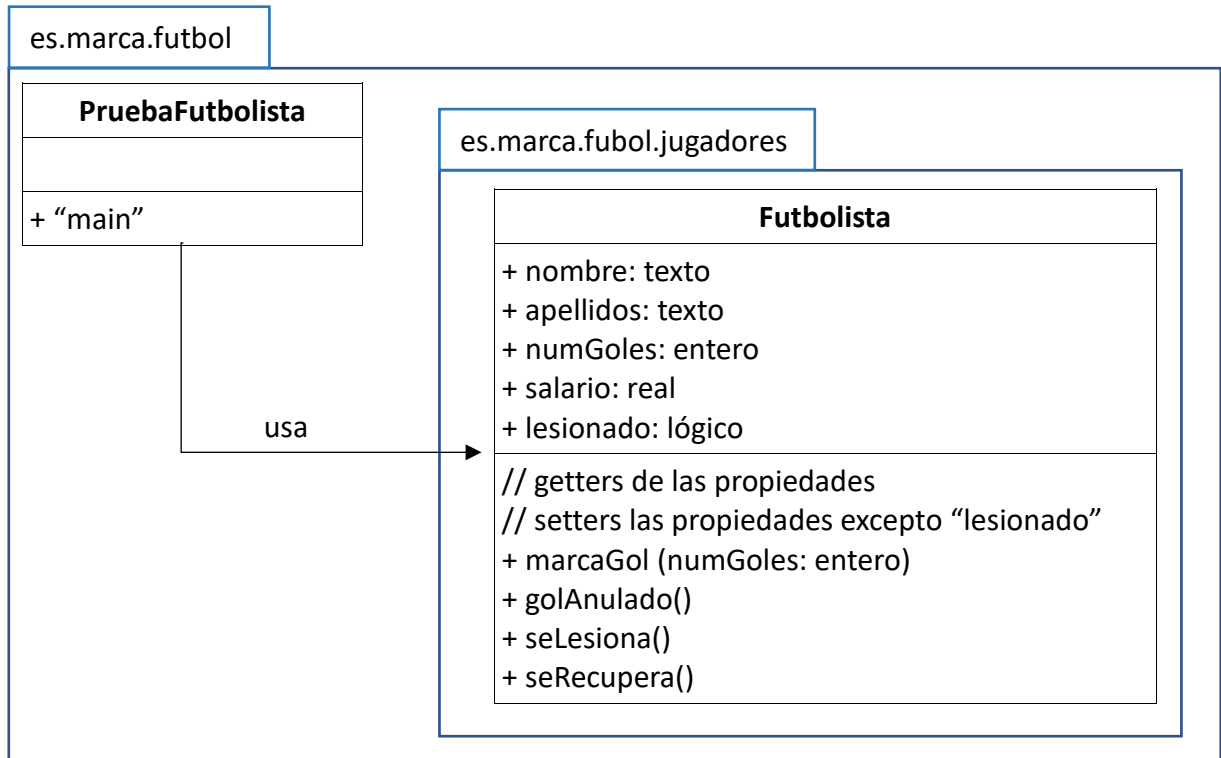


## PRUEBA EVALUABLE UD 1

Nombre y apellidos: \_\_\_\_\_

### EJERCICIO 1 – 5 PUNTOS

Escribir la clase *Futbolista* y probarla con una clase *PruebaFutbolista* siguiendo el siguiente diagrama:



Debes atender, además, a las siguientes restricciones:

- **marcaGol (numGoles)** incrementará la cuenta de goles del futbolista añadiendo los que se reciben como parámetro.
- **golAnulado ()** restará un gol a la cuenta de goles del futbolista.
- **seLesiona()** y **seRecupera()** modificará la propiedad "lesionado" según corresponda.
- La clase **PruebaFutbolista** debe crear dos objetos **Futbolista** y manipular los objetos "creando como una historia" de forma que se utilicen todos los métodos, al menos, una vez cada uno. Una vez se hayan probado todos los métodos se debe imprimir el estado de las propiedades de ambos futbolistas utilizando para ello los métodos `getXXX` de los objetos.

### IMPORTANTE:

*Se debe cumplir con lo que se expresa en el enunciado. No se permite crear métodos nuevos o cambiar los que aparecen en el enunciado.*

**EJERCICIO 2 – 5 PUNTOS**

Escribir la clase *DepositoCircular* que se corresponde con un depósito de riego con una base circular y que lo vamos a modelar como indica el siguiente diagrama:

DepositoCircular
+ radio: real + altura: real + litrosRiegoHora: real + numLitrosActual: real
// getters de las propiedades // setters las propiedades excepto "numLitrosActual" + calculaSuperficieBase() devuelve real + calculaPerimetroBase() devuelve real + calculaCapacidad() devuelve real + cargaDeposito(numLitros: real) + riega (numMinutos: real)

Para ello debes seguir las siguientes restricciones:

- El método *calculaSuperficieBase* devuelve la superficie de la base del depósito ( $Superficie = \pi r^2$ )
- El método *calculaPerimetroBase* devuelve el perímetro de la base del depósito ( $Perimetro = 2\pi r$ )
- El método *calculaCapacidad* devuelve la capacidad en litros del depósito ( $Capacidad = SuperficieBase \times Altura$ )
- Para los cálculos anteriores, se debe utilizar la constante que representa el valor de  $\pi$  de la librería *Math* del lenguaje Java.
- El método *cargaDeposito* añade *numLitros* a los que ya hubiera en el depósito.
- El método *riega* gasta una cantidad de litros proporcional al número de minutos de riego que se recibe como parámetro, teniendo en cuenta el número de litros que se gasta en una hora tal y como expresa la propiedad *litrosRiegoHora*.

A continuación, debes crear una clase *PruebaDeposito* que contenga un método *main*, que cree un objeto de la clase *DepositoCircular* y que manipule el objeto "creando como una historia" de forma que se utilicen todos los métodos. Además, se debe imprimir el estado del objeto ANTES de empezar la "historia", DURANTE el proceso de manipulación del objeto y una vez FINALIZADA la manipulación del objeto.

**IMPORTANTE:**

*Se debe cumplir con lo que se expresa en el enunciado. No se permite crear métodos nuevos o cambiar los que aparecen en el enunciado.*