

2º EVALUACIÓN: EXAMEN PRÁCTICO JAVA		
Curso: 1SI		
Entidad: Fundación San Valero	Cod.Curso: 1SI	Fecha: ¿?/¿?/¿?¿?
Nombre:		
		Duración: 3:00
EJERCICIO: WARHAMMER PIELES VERDES VS ENANOS		10.00pto
<p>Antes de empezar...</p> <ul style="list-style-type: none"> • Valora qué clases deberían ser abstractas o finales y crea los métodos getter/setter que son realmente necesarios. • Presta especial atención al modificador de acceso de las propiedades y métodos (public, private, protected, etc.) y al tipo de dato que devuelven los métodos. • Crea las clases dentro de una jerarquía de paquetes adecuada. • Recuerda comentar el código 		
Nota del 0 al 3		
<ul style="list-style-type: none"> • Preparo la arquitectura de clases que se observan en las figuras 1 y 2. <p>Creamos el Ejército de Pielés Verdes:</p> <ul style="list-style-type: none"> • Una vez diseñado, preparo un main() que cumpla con las siguientes condiciones: <ul style="list-style-type: none"> ○ Creo 2 Goblins con un <u>nivel 20</u>. ○ Creo 2 Martilladores con una riqueza 1000. ○ Los 2 Goblins los guardo en un <u>ArrayList<Goblin></u>. ○ Los 2 Martilladores los guardo en un <u>ArrayList<Martillador></u>. ○ Añado a cada objeto Goblin dos armas actualizando el HashMap<>: <ul style="list-style-type: none"> ▪ “Ataque”: una Rebanadora o un Arco (se le asigna de forma aleatoria una única arma) ▪ “Defensa”: un Armadura o Escudo (se le asigna de forma aleatoria una única arma) ○ Añado a cada objeto Martillador dos armas actualizando el HashMap<>: <ul style="list-style-type: none"> ▪ “Ataque”: un Hacha o un Martillo (se le asigna de forma aleatoria una única arma) ▪ “Defensa”: un Casco o Armadura (se le asigna de forma aleatoria una única arma) ○ Recorro en orden el ArrayList, recupero cada personaje e imprimo su nombre, vida y armas utilizando los métodos programadas dentro de cada una de las clases. ○ Imprimo el número de personajes creados en total (Siempre utilizando los métodos programados en cada una de las clases). 		
Nota del 3 al 5		

Creamos el Ejército de Jefazos:

- Aprovechamos el main() anterior y añadimos funcionalidad para que cumpla las siguientes condiciones:
 - Creo 2 **Orcos** con un **nivel 40**.
 - Creo 2 **Rompehierros** con una riqueza 2000.
 - Los 2 **Orcos** los guardo en un ArrayList<PielVerde>.
 - Los 2 **Rompehierros** los guardo en un ArrayList<Enano>.
 - Añado a cada objeto **Orco** dos armas actualizando el HashMap<>:
 - “Ataque”: un Arco o una Rebanadora (se le asigna de forma aleatoria una única arma)
 - “Defensa”: un Escudo o Armadura (se le asigna de forma aleatoria una única arma)
 - Añado a cada objeto **Rompehierro** dos armas actualizando el HashMap<>:
 - “Ataque”: un Hacha o un Martillo (se le asigna de forma aleatoria una única arma)
 - “Defensa”: un Casco o Armadura (se le asigna de forma aleatoria una única arma)
 -
 - Compruebo el funcionamiento correcto del ataque. Para probar esta funcionalidad deberás crearte unos personajes con armas que se ajusten a cada prueba.
 - Con un Rompehierros ataco a un Goblin (no se podrá defender).
 - Realizo un ataque con Arco varias veces (falla tiros).
 - Realizo un ataque con Arco a alguien que tenga un Casco (solo defiende de los arcos)
 - Realizo un ataque con Martillo a alguien que tenga un Escudo (defiende de Martillo y Hacha)
 - Realizo dos ataques (uno con cada arma) contra alguien que tenga un Armadura (de deberá defender de ambas armas)
 - Imprimo el número de personajes creados en total.

Nota del 5 al 8

- Preparo la arquitectura de clases que se observa en la figura 3.
- Una vez diseñado, preparo un main() que cumpla con las siguientes condiciones:
 - Creo una escaramuza con el nombre que quiera.
 - Añado a Orco con su batallón con la etiqueta “CombatientesMalos”.
 - Añado a Rompehierro con su batallón con la etiqueta “CombatientesBuenos”.
 - Imprimo combate.
 - Ejecuto el combate: Ejecuta un ataque de cada Personaje sobre un Personaje aleatorio del otro ejército
 - Vuelvo a imprimir el estado del combate y el estado de sus personajes.
 - Alimento a todos los personajes.
 - Vuelvo a imprimir el estado del combate y el estado de sus personajes.

Nota del 8 al 10

- Preparo la arquitectura de clases que se observa en la figura 4.

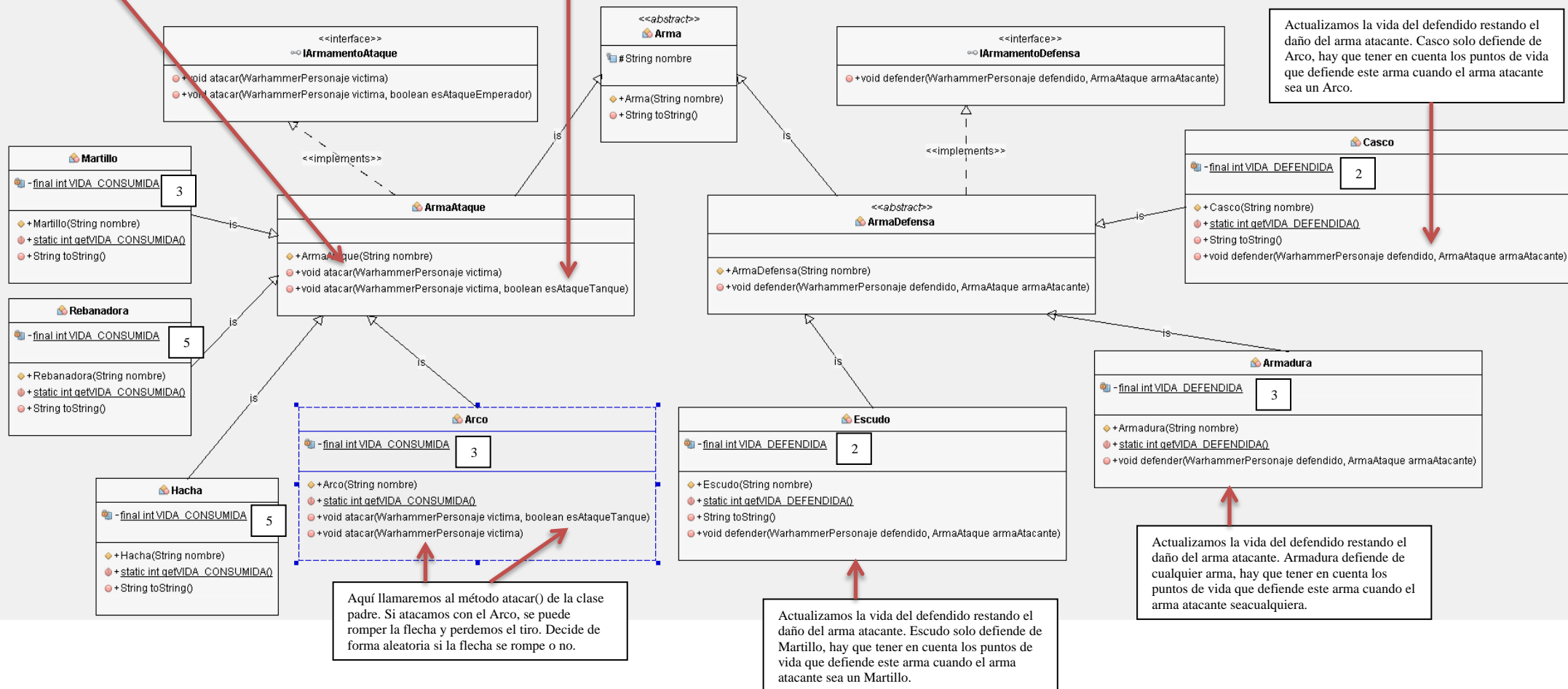
- Una vez diseñado, preparo un main() que cumpla con las siguientes condiciones:
 - Creo una Guerra.
 - Preparo 2 batallas con líderes y escuadrones buenos y malos.
 - Creo 2 personajes nuevos, un Goblin y un Martillador.
 - Añado uno de ellos a una batalla con el nombre que le hayas puesto.
 - Añado el otro a la otra batalla con el nombre que le corresponda y al escuadrón que le corresponda.
- Imprimo guerra.

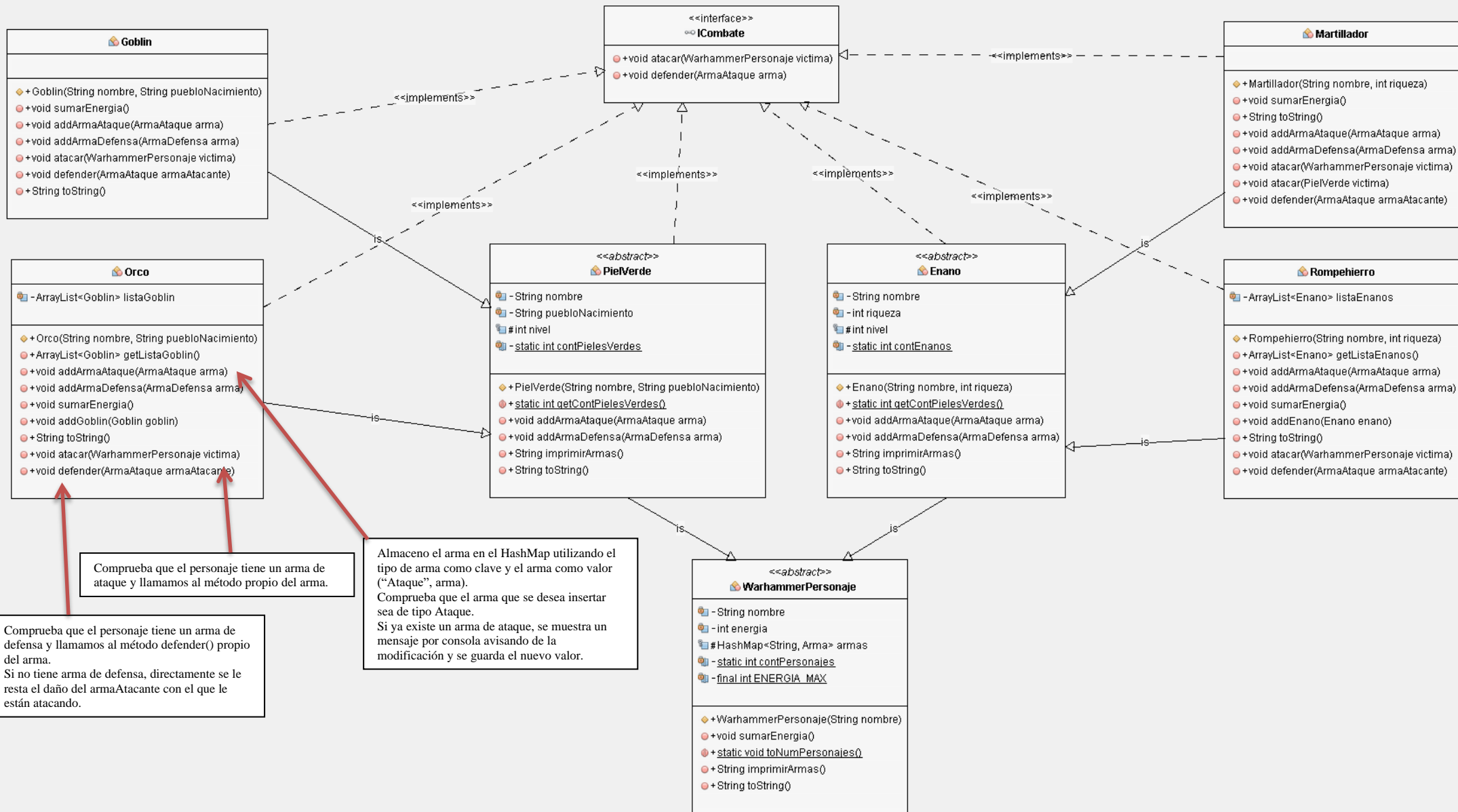
Criterios de corrección	
Optimización y reutilización del código	3,00 ptos
Comentarios	1,00 ptos
Atributos/métodos descriptivos	2,00 ptos
Ejecución	4,00 ptos

Aquí llamaremos al método defender() propio de la víctima.

Este método lo utilizarán los Jefes (tanques) para atacar. El comportamiento es igual al método atacar() anterior, lo único que hay que tener en cuenta es que si viene el segundo parámetro a true y la víctima es un esbirro (Goblin o Martillador), no se podrá defender.

Actualizamos la vida del defendido restando el daño del arma atacante. Casco solo defiende de Arco, hay que tener en cuenta los puntos de vida que defiende este arma cuando el arma atacante sea un Arco.

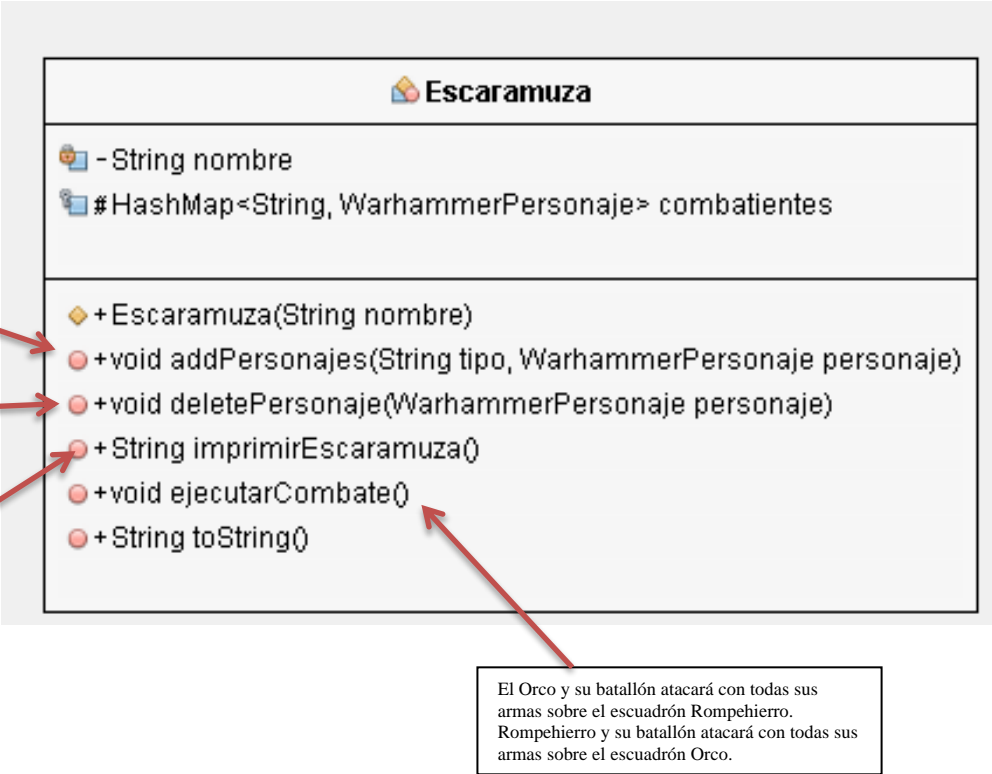


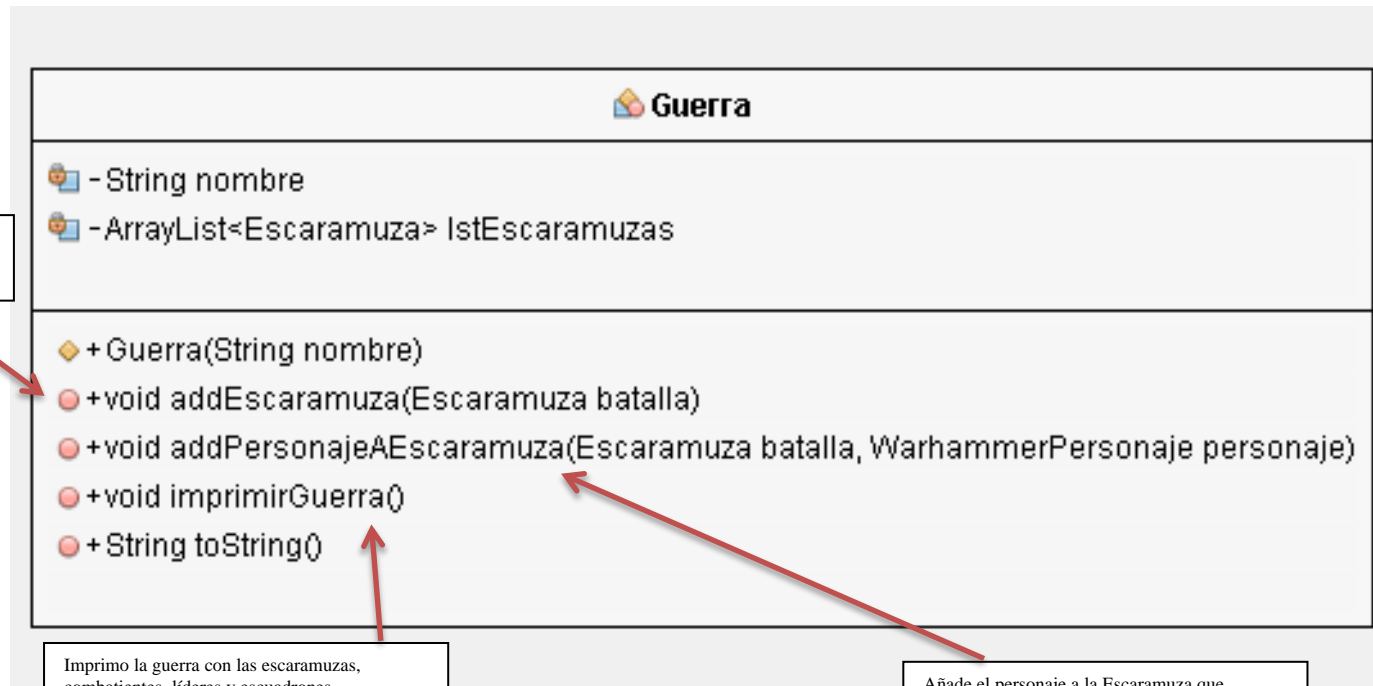


Si es de tipo "CombatientesBuenos", se los
añado al Rompehierro. Si es de tipo
"CombatientesMalos, se los añado al Orco.

Elimino el personaje por nombre

Recorro los CombatientesBuenos y muestro su
líder y su escuadrón.
Recorro los CombatientesMalos y muestro su
líder y su escuadrón.





Añade una Escaramuza a la Guerra con los CombatientesBuenos y Malos y con sus líderes y escuadrones.

Imprimo la guerra con las escaramuzas, combatientes, líderes y escuadrones.

Añade el personaje a la Escaramuza que corresponda teniendo en cuenta si le corresponde ir al equipo bueno o malo