	IES Juan de la Cierva	DWES
	Fecha:	Evaluación 1 - E1
	Nombre:	

Ejercicio: Objetos y arrays

Tiempo: 30 análisis + 40 desarrollo

Puntos: 5

Simulación de guerras

En la guerras que vamos a simular solo lucharán dos ejércitos, estos ejércitos están entrenados para luchar hasta la muerte. La guerra acabará cuando un ejercito se quede sin tropas.

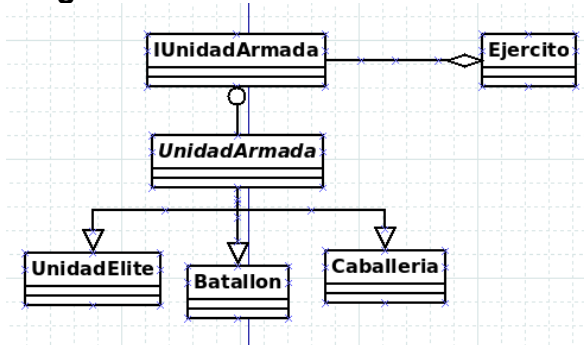
La estructura general del problema está hecha.

Existe un guerra.php que simula una guerra.

El listado de elementos de la aplicación a realizar en el examen es:

- 1 Clase abstracta UnidadArmada que implementa el interface IUnidadArmada
- 0.5 Clase UnidadElite hereda de UnidadArmada
- 0.5 Clase Batallon hereda de UnidadArmada
- 0.5 Clase Caballeria hereda de UnidadArmada
- 1 método estático de UnidadArmada generaUnidad
- 1.5 Clase Ejercito que contiene varias unidades armadas

Diagrama de clases




IUnidadArmada

Todas las tropas deben implementar el interface **IUnidadArmada** que se provee con el examen en src/IUnidadArmada.php

```

interface IUnidadArmada
{
    public function ataqueEfectivo():bool;
    public function ataca(IUnidadArmada $unidad);
    public function dibuja();
    public function estaViva():bool;
}
  
```

	IES Juan de la Cierva	DWES
	Fecha:	Evaluación 1 - E1
	Nombre:	

Clase abstracta UnidadArmada

Todas las unidades armadas tienen dos atributos **protegidos** que son **vida** y **danio**. **vida** indica la cantidad de puntos de vida que le quedan a la unidad, **danio** indica la cantidad de puntos de vida que la unidad quita cuando realiza un ataque sobre otra.

El método **dibuja**, pinta la información de la unidad: tipo de unidad y sus puntos de vida. Este método se debe implementar en las subclases puesto que cada unidad se pintará de una forma.

Ejemplo de código imaginando que las unidades Batallon tienen 100 puntos de vida:

```
$ue1 = new Batallon();
```

```
$ue1->dibuja();
```

Muestra como HTML

```
( Batallon v: 100 )
```

El método **ataqueEfectivo** devuelve un valor bool (true o false) que indica si se ha podido realizar un ataque. Se definirá en las subclases, cada tipo de unidad especificará una forma de determinar si un ataque se ha realizado o no.

El método **atacar** realizará el ataque y pintará el resultado del ataque, en concreto pinta:

- Información de la unidad que ataca, llamando a su método **dibuja**
- Si se ha podido realizar el ataque con las cadenas:
 - ataque efectivo !!!!!
 - ataque fallido xxxxx
- Información de la unidad que es atacada, llamando a su método **dibuja**

Dentro del método **atacar**, se debe determinar si el ataque es efectivo llamando al método **ataqueEfectivo**. En caso de que se pueda realizar el ataque, se le restará a la vida de la unidad que recibe el ataque los puntos de danio de la unidad atacante.

Ejemplo suponiendo que una unidad de élite tiene 20 puntos de danio y un batallón 100 de vida.

```
$ue1 = new UnidadElite();
```

```
$ue2 = new Batallon();
```

```
$ue1->ataca($ue2);
```

Si el ataque es efectivo...

```
Atacante: ( UnidadElite v: 100)
```

```
ataque efectivo !!!!!
```


```
Recibe ataque: ( Batallon v: 80)
```

Si el ataque **no** es efectivo...

```
Atacante: ( UnidadElite v: 100)
```

```
ataque fallido xxxxx
```

```
Recibe ataque: ( Batallon v: 100)
```

	IES Juan de la Cierva	DWES
	Fecha:	Evaluación 1 - E1
	Nombre:	

Por último el método **estaViva**, devuelve true si los puntos de vida son mayores a 0.

Unidades:

Las unidades extienden a la clase abstracta UnidadArmada y especifican: el constructor, ataqueEfectivo y dibujo.

NOTA: Los constructores solo establecen el valor de los atributos protegidos, **no reciben parámetros**.

UnidadÉlite

- En el constructor de la unidad de élite se establecerá su vida a 100 y su danio a 20
- Las unidades de élite siempre devuelven true en la llamada al método ataqueRealizado
- Cuando se pinta una unidad de élite se pinta así (v es la vida actual):
 - (UnidadÉlite v: 100)

Batallón

- En el constructor del batallón se establecerá su vida a 80 y su danio a 20
- Para determinar si un ataque es efectivo escogerán un número entre 1 y 100 y devolverán true si es divisible entre 3
- Cuando se pinta una unidad batallón se pinta así (v es la vida actual):
 - (Batallón v: 65)

Caballería


- En el constructor de la caballería se establecerá su vida a 120 y su danio a 50
- Para determinar si el ataque es efectivo generará dos números aleatorios entre 1 y 10 y devolverá true si el primero es menor al segundo.
- Cuando se pinta una unidad caballería se pinta así (v es la vida actual):
 - (Caballería v: 120)

Método generaUnidad()

Este es un método estático de la clase UnidadArmada devolverá de forma aleatoria un nuevo objeto de tipo: UnidadElite o Batallon o Caballeria.

Ejercito

- Tendrán un constructor que recibe un nombre y un número de tropas, por defecto el número de tropas será 10.
 - Cuando se cree el objeto generará las tropas indicadas en el constructor.
- Tendrá un método formación que colocará el array de unidades de forma aleatoria.
- Tendrá un método ataca, que realizará un ataque de las tropas del ejercito atacante sobre el que recibe el ataque.
 - Debes tener en cuenta que puede que un ejercito tenga más unidades.
- Tendrá un método retirar tropas que retirará del array de unidades las tropas fallecidas.
 - Consejo: Crea un array auxiliar
- Tendrá un método puedeLuchar que indica si aún quedan unidades dentro del ejercito.

	IES Juan de la Cierva	DWES
	Fecha:	Evaluación 1 - E1
	Nombre:	

- Los objetos de este tipo podrán ser concatenados con cadenas. Al concatenarlo mostrará el nombre y el número de unidades:
 - Si el nombre es “Gran ejercito de programadores” y tenemos 3 unidades
 - al escribirlo se mostrará:
 - “Gran ejercito de programadores con 3 unidades”

Bonus

1 punto:

Realiza otra página de guerras en la que el ejercito que primero ataca sea aleatorio pero que luego se vayan alternando los ataques.