

DOCUMENTO EXPLICATIVO DE QUE PARTES DEL PROYECTO SE HAN USADO HERENCIA, COMPOSICIÓN...

Alejandro Caballero, Pedro José, Juan Antonio Noci

CAMPUSFP

1. Herencia

```
package pujas;  
  
public class Pujador extends Persona {  
    private float dinero;  
    private int pujasganadas;  
  
    public Pujador(String nombre, String dni, String direccion, float dinero, int pujasganadas) {  
        super(nombre, dni, direccion);  
        this.dinero = dinero;  
        this.pujasganadas = pujasganadas;  
        // TODO Auto-generated constructor stub  
    }  
  
    public Pujador() {  
        super("Caba", "53906685K", "Calle falsa 123");  
        this.dinero = 15600000;  
        this.pujasganadas = 100000000;  
        // TODO Auto-generated constructor stub  
    }  
  
    public float getDinero() {  
        return dinero;  
    }  
    public void setDinero(int dinero) {  
        this.dinero = dinero;  
    }  
    public int getPujasganadas() {  
        return pujasganadas;  
    }  
    public void setPujasganadas(int pujasganadas) {  
        this.pujasganadas = pujasganadas;  
    }  
}
```

Hemos usado la herencia en la clase pujador que hereda datos básicos de identificación de la clase Persona.

2. Composición

```
public Puja(String nombre, float precio, String estado, String descripcion, int id) {  
    super();  
    this.precioInicial = precio;  
    this.idPuja = id;  
    Producto producto = new Producto(nombre, precio, estado, descripcion, id);  
}
```

Hemos usado la composición para añadir un nuevo producto cada vez que se añada una puja ya que como es obvio sin un producto es imposible realizar esta misma.

```
int idPuja;  
float precioInicial;  
String ganador;  
Pujador arrayPujador[] = new Pujador[1024];  
int contadorPujadores = 0;
```

```
public void agregarPujador(String nombre, String dni, String direccion, float dinero) {  
    contadorPujadores = contadorPujadores + 1;  
    int pujasganadas = 0;  
    arrayPujador[contadorPujadores] = new Pujador(nombre, dni, direccion, dinero, pujasganadas);  
}
```

También hemos metido a través de composición débil en puja los pujadores en un array

3. Sobrecarga

```
public Pujador() {  
    super("Caba", "53906685K", "Calle falsa 123");  
    this.dinero = 15600000;  
    this.pujasganadas = 100000000;  
    // TODO Auto-generated constructor stub  
}
```

Usamos la sobrecarga en un constructor para meter un pujador predeterminado con los datos de uno de nosotros

4. Sobreescritura

```
@Override  
public String toString() {  
    return "Puja [idPuja=" + idPuja + ", precioInicial=" + precioInicial + ", contadorPujadores=" + contadorPujadores + "];"  
}
```

Para la sobreescritura no nos complicamos mucho las cosas ya que al verlo poco optimo para nuestro proyecto decidimos solo usarlo para sobrescribir el toString que te devuelve el espacio de memoria predeterminado por un toString que te da los datos de la puja.