

Programación Frontend y Backend

BLOQUE JAVA

Objetos III

01

Instance of



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro



Operador instanceof

- El operador java **instanceof** se utiliza para probar si el objeto es una instancia del tipo especificado (clase, subclase o interfaz)
- El **instanceof** en java también se conoce como operador de comparación de tipos porque compara la instancia con el tipo. Devuelve verdadero o falso. Si aplicamos el operador **instanceof** con cualquier variable que tenga un valor nulo, devuelve falso.

Operador instanceof

```
SeleccionFutbol[] plantilla = new SeleccionFutbol[5];  
plantilla[0] = futbolista1;  
plantilla[1] = futbolista2;  
plantilla[2] = entrenador1;  
plantilla[3] = entrenador2;  
plantilla[4] = masajista1;
```

¿Es un objeto de tipo SeleccionFutbol?
¿Es un objeto de tipo Futbolista?
¿Es un objeto de tipo Entrenador?
¿Es un objeto de tipo Masajista?



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro



Operador instanceof

```

SeleccionFutbol[] plantilla = new SeleccionFutbol[5];
plantilla[0] = futbolista1;
plantilla[1] = futbolista2;
plantilla[2] = entrenador1;
plantilla[3] = entrenador2;
plantilla[4] = masajista1;

for(SeleccionFutbol seleccion : plantilla) {

    System.out.println(seleccion);

    if(seleccion instanceof SeleccionFutbol) {
        System.out.println("- Soy SeleccionFutbol");
    }

    if(seleccion instanceof Futbolista) {
        System.out.println("- Soy Futbolista");
    }

    if(seleccion instanceof Entrenador) {
        System.out.println("- Soy Entrenador");
    }

    if(seleccion instanceof Masajista) {
        System.out.println("- Soy Masajista");
    }

}

```

Polimorfismo

```

Console
<terminated> Futbol [Java Application] C:\ej_ico_0\Java\jdk1.8.0_144\bin\javaw.exe (14 nov. 2020 15:28:47)
¡Soy un Futbolista y me llamo:Leo Messi!
- Soy SeleccionFutbol
- Soy Futbolista
¡Soy un Futbolista y me llamo:Cristiano Ronaldo!
- Soy SeleccionFutbol
- Soy Futbolista
Soy un Entrenador y me llamo:Ronald Koeman, gracias.
- Soy SeleccionFutbol
- Soy Entrenador
Soy un Entrenador y me llamo:Zinedine Zidane, gracias.
- Soy SeleccionFutbol
- Soy Entrenador
Hola, soy un Masajista y tengo:25 años.
- Soy SeleccionFutbol
- Soy Masajista

```

Operador instanceof

Polimorfismo

```
for(SeleccionFutbol seleccion : plantilla) {  
    System.out.println(seleccion);  
  
    if(seleccion instanceof SeleccionFutbol) {  
        System.out.println("- Soy SeleccionFutbol");  
    }  
  
    if(seleccion instanceof Futbolista) {  
        System.out.print("- Soy Futbolista");  
        int dorsal = ((Futbolista) seleccion).getDorsal();  
        System.out.println(", Mi Dorsal es:" + dorsal);  
    }  
  
    if(seleccion instanceof Entrenador) {  
        System.out.println("- Soy Entrenador");  
    }  
  
    if(seleccion instanceof Masajista) {  
        System.out.println("- Soy Masajista");  
    }  
}
```

```
Console  
<terminated> Futbol [Java Application] C:\ej_ico_0\Java\jdk1.8.0_144\bin\javaw.exe (14 nov. 2020 15:32:55)  
¡Soy un Futbolista y me llamo:Leo Messi!  
- Soy SeleccionFutbol  
- Soy Futbolista, Mi Dorsal es:10  
¡Soy un Futbolista y me llamo:Cristiano Ronaldo!  
- Soy SeleccionFutbol  
- Soy Futbolista, Mi Dorsal es:24  
Soy un Entrenador y me llamo:Ronald Koeman, gracias.  
- Soy SeleccionFutbol  
- Soy Entrenador  
Soy un Entrenador y me llamo:Zinedine Zidane, gracias.  
- Soy SeleccionFutbol  
- Soy Entrenador  
Hola, soy un Masajista y tengo:25 años.  
- Soy SeleccionFutbol  
- Soy Masajista
```



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial

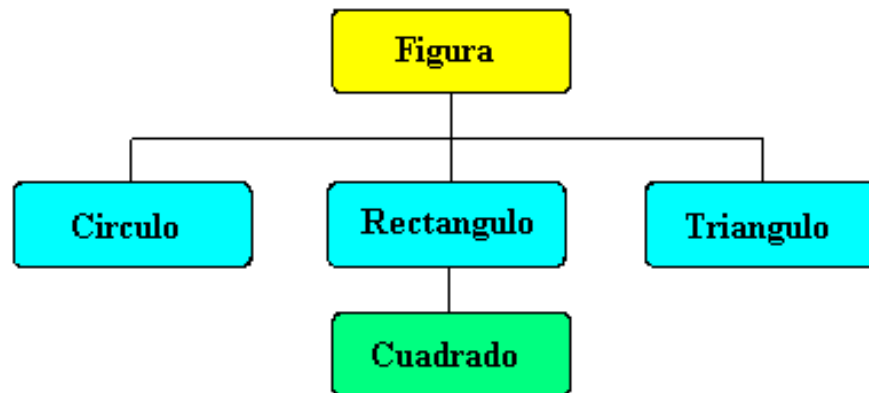


Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro



Polimorfismo

- Es la característica de la Programación Orientada a Objetos que permite que un objeto pueda ser tratado como un objeto de otra clase.



02

Interfaces



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial



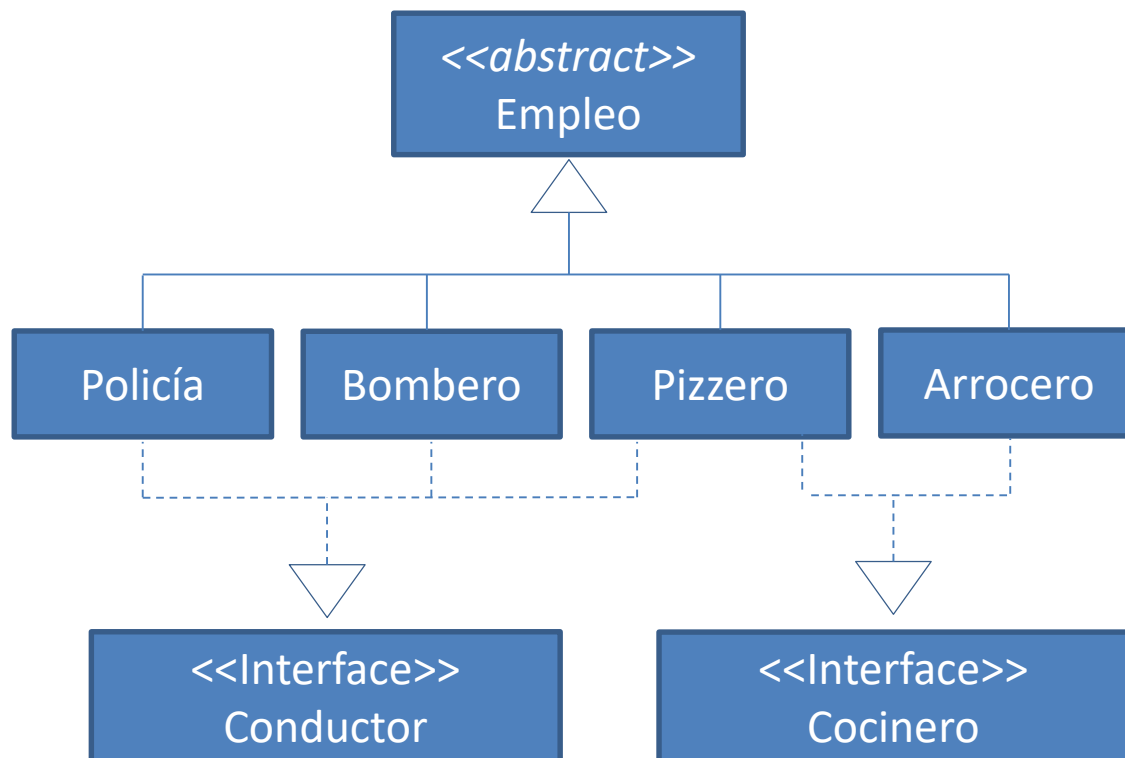
Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro



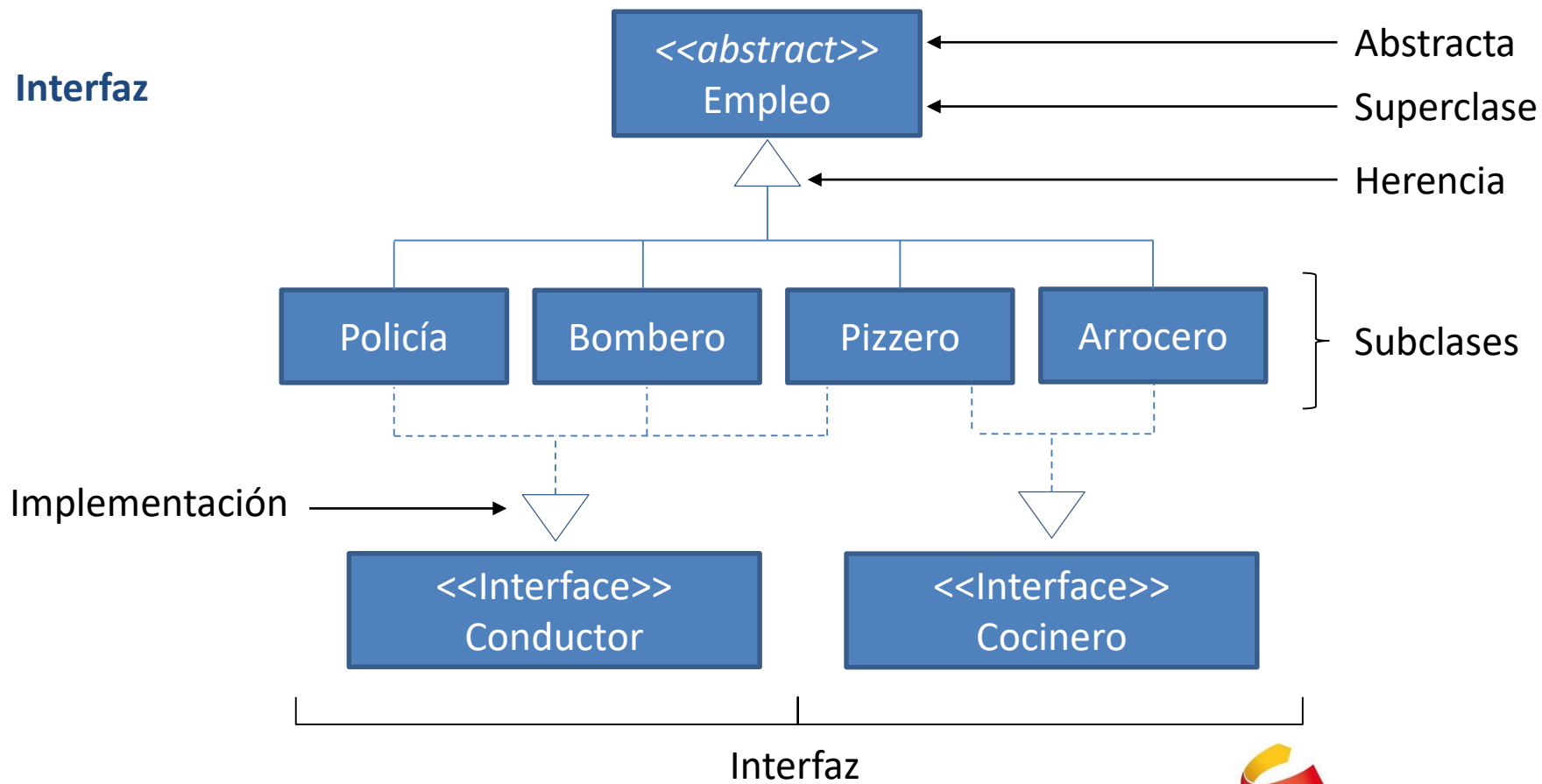
Interfaz

- Una interfaz es una lista de acciones que puede llevar a cabo un determinado objeto.
- Una interfaz sólo existe la definición de un método, no su código.
- Una interfaz en Java es una colección de métodos abstractos y propiedades constantes. En las interfaces se especifica qué se debe hacer pero no su implementación. Serán las clases que implementen estas interfaces las que describen la lógica del comportamiento de los métodos.

Interfaz



Interfaz



<<abstract>>
Empleo

Interfaz

```
public abstract class Empleo {  
    protected double sueldo;  
  
    public abstract void cobrar();  
  
    public abstract void fichar();  
  
    public double getSueldo() {  
        return sueldo;  
    }  
  
    public void setSueldo(double sueldo) {  
        this.sueldo = sueldo;  
    }  
}
```

Interfaz

- ▼ es.eoi.objetos
 - > Cocinero.java
 - > Conductor.java

```
public interface Conductor {  
    public void conducir();  
    public void repostar();  
    public void aparcar();  
}
```

<<Interface>>
Conductor

```
public interface Cocinero {  
    public void cocinar();  
    public void preparar();  
    public void comprarAlimentos();  
}
```

<<Interface>>
Cocinero



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro



Interfaz

```
public class Bombero extends Empleado implements Conductor {  
  
    @Override  
    public void cobrar() {  
        // TODO Auto-generated method stub  
    }  
  
    @Override  
    public void fichar() {  
        // TODO Auto-generated method stub  
    }  
  
    @Override  
    public void conducir() {  
        // TODO Auto-generated method stub  
    }  
  
    @Override  
    public void repostar() {  
        // TODO Auto-generated method stub  
    }  
  
    @Override  
    public void aparcar() {  
        // TODO Auto-generated method stub  
    }  
}
```

<<Interface>>
Conductor<<abstract>>
Empleado

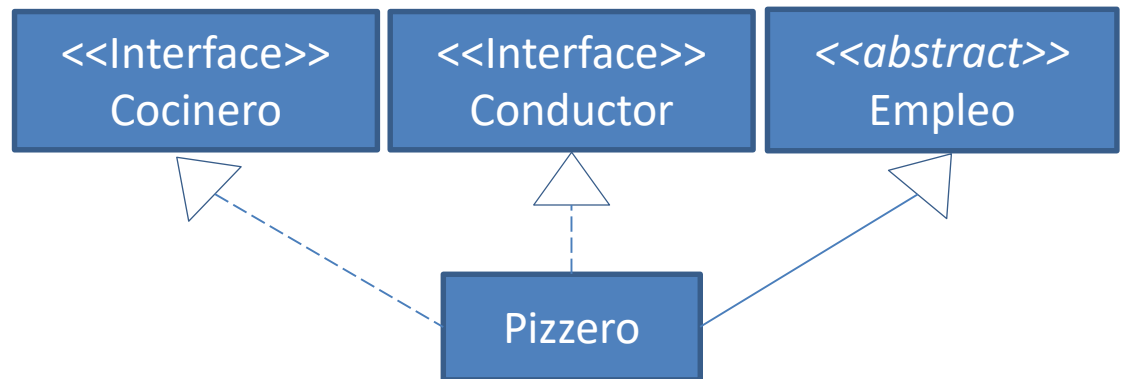
Bombero

Empleado

Conductor

GOBIERNO
DE ESPAÑAMINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMOEscuela de
organización
industrialUnión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuroSISTEMA NACIONAL DE
GARANTÍA
JUVENIL

Interfaz (Pizzero)



```
public class Pizzero extends Empleado implements Conductor, Cocinero {  
    @Override  
    public void cocinar() {  
        // TODO Auto-generated method stub  
    }  
    @Override  
    public void preparar() {  
        // TODO Auto-generated method stub  
    }  
    @Override  
    public void comprarAlimentos() {  
        // TODO Auto-generated method stub  
    }  
}
```

Cocinero

Interfaz (Pizzero)

```
@Override
public void conducir() {
    // TODO Auto-generated method stub
}

@Override
public void repostar() {
    // TODO Auto-generated method stub
}

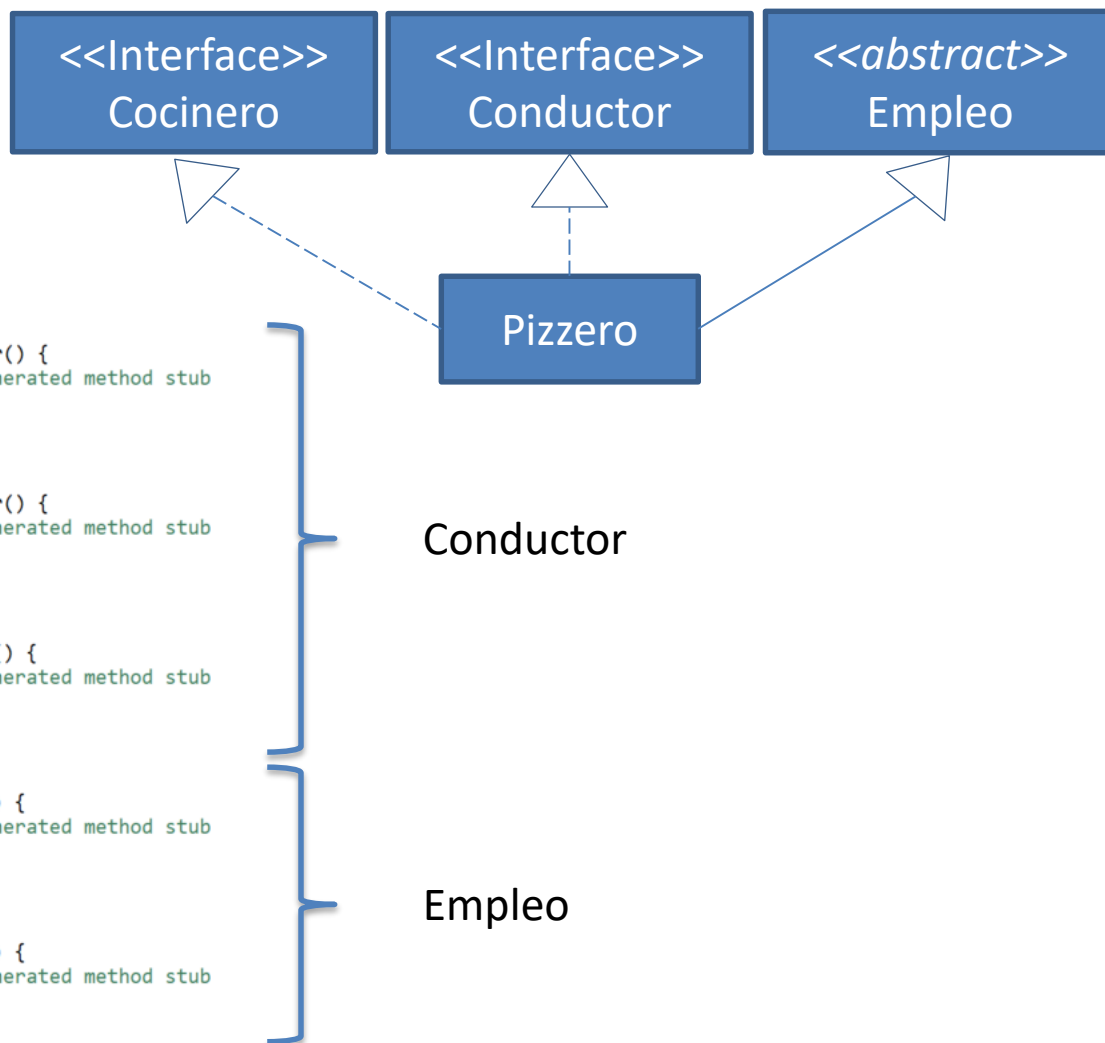
@Override
public void aparcar() {
    // TODO Auto-generated method stub
}

@Override
public void cobrar() {
    // TODO Auto-generated method stub
}

@Override
public void fichar() {
    // TODO Auto-generated method stub
}
```

Conductor

Empleo



04

Ejercicios



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial

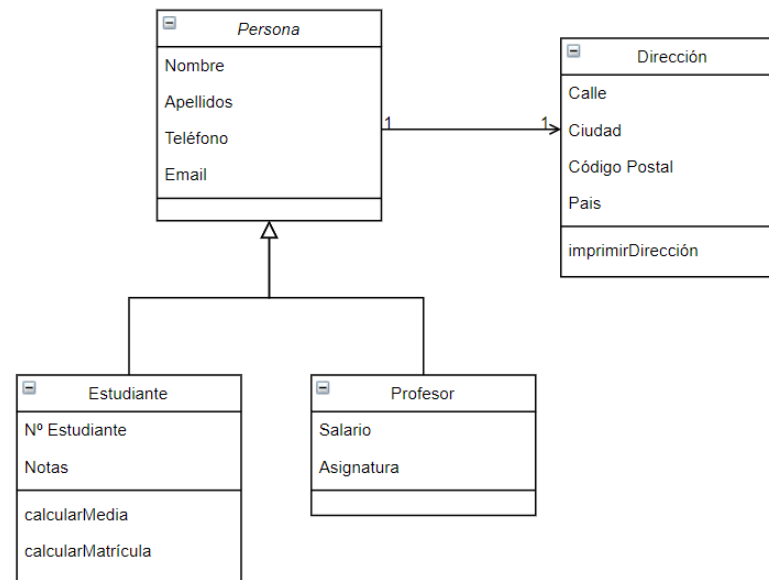


Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro



Ejercicio I

Realizaremos la implementación del siguiente diagrama de clases:



Ejercicio I

- Notas será double []
- La matrícula se calculará contando el número de notas aprobadas x 100 más el número de suspensos x 200.
- Crea una clase main Campus.java, crea 5 alumnos y 5 profesores y almacénalos en un mismo array. Además crearemos un menú con las siguientes opciones:
 - Listar Alumnos
 - Listar Profesores

Ejercicio II

Vamos a representar una biblioteca que tiene libros y revistas, las características comunes que se almacenan tanto para las revistas como para los libros son el código, el título, y el año de publicación. Estas tres características se pasan por parámetro en el momento de crear los objetos.

Los libros tienen además un atributo prestado. Los libros, cuando se crean, no están prestados. Las revistas tienen un número. En el momento de crear. En el momento de crear las revistas se pasa el número por parámetro.

Para prevenir posibles cambios en el programa se tiene que implementar una interfaz Prestable con los métodos prestar(), devolver() y prestado. La clase Libro implementa esta interfaz