



## **Título**

McBurger.

## **Tiempo estimado de ejecución**

120 minutos

## **Objetivo de la actividad**

- Identificar los conceptos de herencia, superclase y subclase.
- Utilizar modificadores para bloquear y forzar la herencia de clases y métodos.
- Reconocer la incidencia de los constructores en la herencia.
- Crear clases heredadas que sobrescriban la implementación de métodos de la superclase.
- Diseñar y aplicar jerarquías de clases.

## **Criterios de evaluación**

- Se ha analizado el problema de forma correcta.
- Se han utilizado correctamente la herencia, la superclase y la subclase en Java.
- Se ha utilizado un entorno de trabajo (IDE) correcto para programar en Java.
- Se ha realizado correctamente el programa que resuelve el problema.

## **Recursos necesarios**

Un entorno de desarrollo integrado (IDE), preferiblemente IntelliJ Idea.



## Enunciado

- 1.- Crea un proyecto nuevo llamado McBurger.
- 2.- Crea tres paquetes distintos: *dominio*, *interfaces* y *principal*.
- 3.- Descarga la clase *Principal.java* y añádela al paquete principal (esta clase NO puede ser modificada).
- 4.- En el paquete interfaces, crea la siguiente interfaz:
  - Cocinable
- 4.1.- En la interfaz Cocinable, define las constantes: **CRUDA** = 0, **FRITA** = 1, **COCIDA** = 2 y **ASADA** = 3.
- 4.2.- En la interfaz Cocinable, define los métodos: freír, cocer y asar. (sin parámetros, pues estos no devolverán nada).
- 5.- En el paquete dominio, crea las siguientes clases:
  - Menú
  - Ingrediente (clase abstracta)
  - Comida (clase abstracta), que hereda de Ingrediente.
  - Bebida (clase abstracta), que hereda de Ingrediente.
  - Hamburguesa, que hereda de Comida.
  - Patata, que hereda de Comida e implementa la interfaz Cocinable.
  - Agua, que hereda de Bebida
  - Cola, que hereda de Bebida

### 5.1.- Clase Menú

Atributos de clase:

numeroMenus (público)

Atributos de instancia (privados):

numeroIngredientes (entero)

listaIngredientes (ArrayList de Ingrediente)

Crear un Constructor sin parámetros con la funcionalidad correspondiente.

Métodos:

Añadir los métodos necesarios para que el programa compile y para que funcione la clase Principal.java.

Añadir también el siguiente método (que NO puede ser modificado):

```
public void imprimirMenu() {  
  
    for (int i= 0; i < this.listaIngredientes.size(); i++) {  
  
        System.out.println(this.listaIngredientes.get(i) );  
    }  
}
```



```
}
```

```
}
```

Consideraciones:

El método *addComida()*, añade una comida o una bebida al menú.

El método *obtenerPrecioMenu()*, obtiene el precio total del menú, es decir, el precio de las comidas más el precio de las bebidas.

## 5.2.- Clase Ingrediente

Atributos de instancia (privados):

nombre (String), el nombre del ingrediente concreto (hamburguesa, agua...).

tipoIngrediente (String), si es comida o bebida.

Getters y Setters para los dos atributos de instancia

Añadir el siguiente método abstracto:

```
//Métodos abstractos  
  
public abstract double obtenerPrecio();
```

Añadir los métodos necesarios para que el programa compile y para que funcione la clase Principal.java.



### 5.3.- Clase Bebida

Atributos de instancia (privados):

refrigerada (boolean)

Crear un constructor sin parámetros con la funcionalidad correspondiente.

Añadir los métodos necesarios para que el programa compile y para que funcione la clase Principal.java.

Consideraciones:

Cuando se crea una bebida, por defecto, esta NO está refrigerada.

El precio de TODAS las bebidas es:

1 € cuando NO está refrigerada.

1,5 0€ cuando SÍ está refrigerada.

### 5.4.- Clase Agua

Crear un constructor sin parámetros con la funcionalidad correspondiente.

### 5.5.- Clase Cola

Crear un constructor sin parámetros con la funcionalidad correspondiente.

### 5.6.- Clase Comida

Atributos de instancia (protegidos):

Cocinado (boolean)

Crear un constructor sin parámetros con la funcionalidad correspondiente.

Consideraciones:

Cuando se crea una comida, por defecto, esta NO está cocinada.



### 5.7.- Clase Hamburguesa

Atributos de instancia (privados):

fechaCaducidad (Date)

Crear un constructor que recibe la fechaCaducidad como un string y crea la funcionalidad correspondiente.

Añadir los métodos necesarios para que el programa compile y para que funcione la clase Principal.java.

Consideraciones:

El precio de una hamburguesa es de 3,50 €.

Si falta 1 día para que caduque, se le hará un descuento del 50 %.

Si faltan 2 días para que caduque, se le hará un descuento del 40 %.

Si faltan 3 días para que caduque, se le hará un descuento del 30 %.

Si faltan 4 días para que caduque, se le hará un descuento del 20 %.

### 5.8.- Clase Patata

Atributos de instancia (privados):

estado (entero), podrá ser cruda, frita, cocida o asada.

Crear un constructor sin parámetros con la funcionalidad correspondiente.

Añadir los métodos necesarios para que el programa compile y para que funcione la clase Principal.java.

Consideraciones:

El precio de las patatas fritas es de 1,10 €.

El precio de la patata cocida es de 0,80 €.

El precio de la patata asada es de 0,90 €.



6.- El resultado de ejecutar la clase Principal.java debe ser lo más parecido posible a esta impresión por pantalla:

```

Problems Javadoc Declaration Console Debug Tasks
<terminated> Principal (7) [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\javaw.exe

*****
** Menú ( 4 ingredientes) **
*****
Agua..... 1,00€
Cola..... 1,50€
Hamburguesa..... 3,50€
Patatas.fritas..... 1,10€

Precio Total..... 7,10€

*****
** Menú ( 4 ingredientes) **
*****
Agua..... 1,50€
Cola..... 1,50€
Hamburguesa..... 3,50€
Patata.asada..... 0,90€

Precio Total..... 7,40€

*****
** Menú ( 5 ingredientes) **
*****
Agua..... 1,00€
Cola..... 1,50€
Hamburguesa..... 3,50€
Patata.cocida..... 0,80€
Patata.cruda..... 0,00€

Precio Total..... 6,80€

*****
** Menú ( 6 ingredientes) **
*****
Agua..... 1,50€
Agua..... 1,00€
Cola..... 1,50€
Hamburguesa..... 3,50€
Patata.cocida..... 0,80€
Patatas.fritas..... 1,10€

Precio Total..... 9,40€

*****
** Menú ( 3 ingredientes) **
*****
Hamburguesa..... 3,50€
Patata.cocida..... 0,80€
Patatas.fritas..... 1,10€

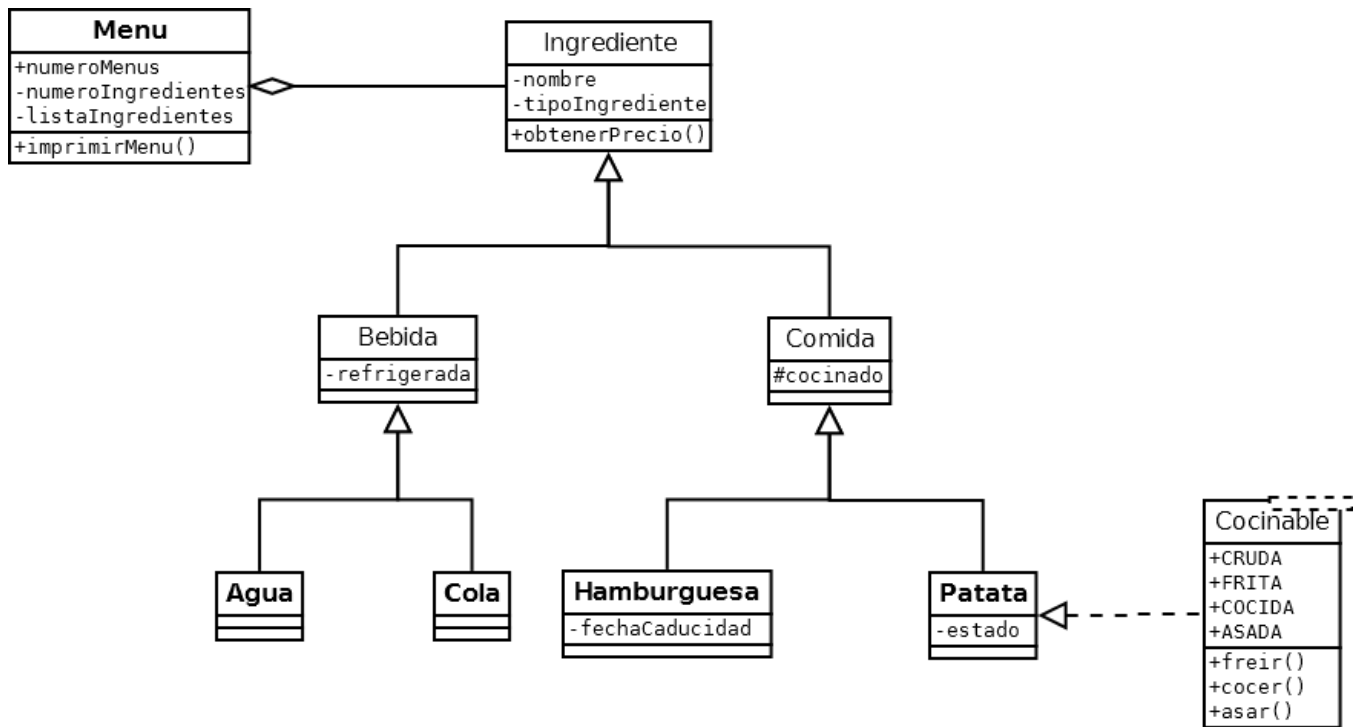
Precio Total..... 5,40€

*****
Total menús: 5
*****

```



El siguiente diagrama de clases nos permite entender mejor la estructura del proyecto:



## Formato e instrucciones de la entrega

Un único archivo comprimido con la carpeta *src* del proyecto, que contendrá los tres paquetes (carpetas), más todas las clases con extensión *.java* necesarias para realizar la tarea.



## Corrección

---

### Solución

Se adjunta la clase Principal.java, que se debe pasar al estudiante para comenzar la tarea.

### Orientaciones para su corrección

El docente observará si se han cumplido los objetivos, para lo que atenderá a los criterios de evaluación fijados.

Se deberá comprobar que la solución es correcta.

Asimismo, se valorará positivamente a aquellas personas que ajusten su solución a la forma de imprimir los tickets, tal y como aparece esta en la imagen del enunciado.