

## Práctica

### Descripción de la Práctica

El objetivo de esta práctica es que el alumno sea capaz de construir una aplicación simplificada para la gestión de ventas de una tienda on-line de componentes informáticos. A lo largo de esta memoria se llamará indistintamente “componente” o “producto” a cada uno de los componentes informáticos que se venden en la tienda.

### Etapa 1

Dentro de esta primera parte se realizará una aproximación básica al modelado inicial de algunos de los elementos del sistema.

**Clientes y Fabricantes.** Para el desarrollo básico se necesitará tener información básica de los clientes y los fabricantes. En ambos casos se deben modelar como objetos. Para ello debemos tener en cuenta que un cliente debe tener las siguientes características:

- código de cliente
- nombre del cliente
- apellidos del cliente
- número de tarjeta bancaria (en un String)

Para el caso del fabricante se deberán tener las siguientes características:

- código del fabricante
- nombre del fabricante
- teléfono

Los elementos básicos de nuestro sistema en este punto del desarrollo serán los **componentes** informáticos que se venden como productos en nuestra tienda. Cada componente concreto se modela como un objeto y, por tanto, para definir un componente necesitaremos crear una clase. Para ello, tenga en cuenta que un componente tendrá las siguientes características:

- Fabricante.
- Código interno de producto.
- Descripción de producto.
- Precio de venta al público (PVP).

Además, los objetos fabricante, cliente y componente debe permitir consultar y modificar estos campos a través de una serie de métodos creados a tal efecto: por cada campo debe haber dos métodos, uno para consultar su valor y otro para modificarlo. Por ejemplo, para el campo PVP tendríamos los métodos getPVP() y setPVP().

Asimismo, se deberá permitir visualizar por pantalla información del producto (el nombre y código del fabricante, el código interno, la descripción de producto y el PVP) con un formato legible como, por ejemplo, el mostrado a continuación:

```
Producto: <códigoProducto> <descripciónProducto>  
Precio: <precioProducto>  
Fabricante: <nombreFabricante> (<códigoFabricante>)
```

Para ello se implementará el método `toString` que se llamará en las visualizaciones por pantalla que se realicen en el `main`.

Crear también un **CarritoDeLaCompra** que deberá cumplir las siguientes condiciones:

1. Deberá poder almacenar una lista de productos que el usuario esté pensando en comprar.
2. Deberá contener una fecha y el cliente que realiza la compra. Por simplicidad asumimos que para que un cliente pueda usar un carrito será necesario estar registrado en el sistema
3. Se ha de permitir añadir y eliminar productos del carrito. Se puede tomar el código interno de un producto como parámetro para identificarlo. Nótese que una venta puede contener varias veces un mismo componente, es decir, que un usuario puede comprar dos productos iguales y, por tanto, el carrito podrá contener varias instancias de un mismo producto.
4. El `CarritoDeLaCompra` debe ser capaz de presentar por pantalla los productos que contiene y sus precios, así como el precio total de la posible venta. Se mostrará la información básica de cada uno de los productos contenidos en el carrito (para ello se debe utilizar el método creado en la clase `Componente`). Se debe implementar el método `toString` para realizar la visualización por pantalla del carrito.
5. Deberá tener un método `comprar` que permita efectuar la compra del contenido del carrito y que devuelva un objeto de la clase `Venta` (más adelante se describen las características que deberá de tener esta clase). Para ello habrá que introducir la fecha actual y una forma de pago, que podrá ser únicamente por “transferencia bancaria” o mediante “pago con tarjeta”.
6. En relación al tratamiento de errores, la aplicación ha de contemplar casos como que un código interno no exista (por ejemplo, cuando se quiera eliminar un producto de un carrito) o que no haya componentes para listar porque el carrito esté vacío. Deberá gestionar este tipo de situaciones informando adecuadamente por pantalla de lo que sucede.

Se debe crear una nueva clase `Venta` que:

1. Deberá contener los mismos atributos que tiene la clase `CarritoDeLaCompra`, es decir, los productos seleccionados, un cliente y una fecha que será la correspondiente a la venta y, por tanto, que no tendrá por qué coincidir con la fecha del carrito de la compra.
2. Además contendrá el método de pago con el que se realizó la compra, que como ya se comentó anteriormente podrá ser únicamente por medio de “transferencia bancaria” o mediante “pago con tarjeta”.

Implemente a continuación un método `main()` que permita lanzar la aplicación de forma independiente y que contenga el código necesario para crear un `CarritoDeLaCompra`, añadirle varios componentes, eliminar alguno de ellos y presentar por pantalla el listado de productos, sus precios de venta al público, el precio total y la información básica de cada producto contenido en el carrito. También se realizará la venta del carrito y se visualizará en contenido de la venta efectuada.