

Ejercicio Singleton

El patrón Singleton se utiliza para limitar la creación de instancias de una clase a una única instancia. Este patrón es ideal cuando se requiere un punto de acceso global y controlado a un recurso compartido. En este ejercicio, se aplica el patrón Singleton a la clase `CarritoCompra` en el contexto de una tienda en línea.

Explicación del ejemplo implementado

En una tienda en línea, al agregar productos al carrito de compras, es fundamental que el carrito sea el mismo en todas las pestañas del navegador. Tener carritos de compras separados en cada pestaña podría causar confusión y problemas durante el proceso de compra. El patrón Singleton aborda esta necesidad al garantizar que solo exista una instancia del carrito de compras.

A continuación, se presenta una explicación general de cómo se implementa este patrón en el código:

Este ejercicio consta de 3 clases:

- **Clase Productos:** La clase `Productos` representa un producto en una tienda con tres atributos: `id`, `nombreArticulo` y `precio`. Cada instancia de la clase `Productos` tiene un identificador único generado automáticamente, un nombre y un precio. Esta clase incluye métodos para obtener y modificar estos atributos, así como un método para representar el producto como una cadena de texto.
- **Clase CarritoCompra:** La clase `CarritoCompra` representa un carrito de compras que contiene una lista de productos. Utiliza el patrón Singleton para asegurar que solo haya una instancia de `CarritoCompra` en toda la aplicación. La clase proporciona métodos para agregar y eliminar productos del carrito, ver los productos actuales en el carrito, calcular el precio total de los productos y realizar la compra de todos los productos en el carrito.
- **Clase Principal:** La clase `Principal` contiene el método `main`, que es el punto de entrada de la aplicación. En este método, se crean varios productos, se agrega y se elimina productos del carrito, se visualiza el contenido del carrito y se simula la compra de los productos.