
Proyecto 01

Equipo Michitas:

Díaz Reyes Lilith Jaquelin - 320277571

Monroy Romero Sahara Mariel - 320206391

Del Monte Ortega Maryam Michelle - 320083527

Descripción del proyecto.

Michi3B es una tienda con aspiraciones de crecer, por lo que hemos creado este programa que permite a los clientes de México, España y Estados Unidos comprar de forma segura y eficiente, pensando en que el programa sea flexible y escalable para soportar el tráfico de clientes a futuro.

Estructura del proyecto.

- src/: Esta carpeta contiene los archivos fuente de Java.
- UML.png contiene el diagrama UML que representa al proyecto.
- Secuencias.png contiene el diagrama de secuencias que representa al proyecto.

Estructura del proyecto.

Para compilar y ejecutar este proyecto, asegúrate de tener instalado java en tu sistema. Luego, sigue estos pasos:

1. Abre una terminal y navega hasta la carpeta raíz del proyecto: `\Proyecto01_Michitas\src\`
2. Compila el proyecto ejecutando el siguiente comando: `javac *.java`
3. Ejecuta el programa con el siguiente comando: `java Michi3B`

Justificación de patrones utilizados.

En este proyecto optamos por usar 4 patrones: Facade, Strategy, Proxy e Iterator.

Las razones de estas elecciones son las siguientes:

- Facade.

Para mantener nuestro código organizado elegimos Facade, ya que permite:

1. Juntar los subsistemas del programa.
2. Reutilizar código, evitando tener que repetir código para interactuar con los subsistemas en distintas partes del programa.
3. Crear un software flexible, ya que los cambios en la implementación interna de los subsistemas no afectan a la fachada, permitiendo hacer modificaciones en los mismos de ser necesario.
4. Mantener una organización que propicie la claridad y fácil lectura del código, haciendo de este comprensible y mantenible.

- Strategy.

Con razón de poder hacer los mismos procesos de forma distinta según la región desde la cual el usuario se conectaba usamos Strategy. Sus ventajas son:

1. Es sencillo adaptar cada estrategia según las necesidades de cada región, y en caso de ser necesario, incluso modificarlas sin afectar los procesos de las otras regiones, volviendo al programa flexible.
2. Permite la escalabilidad, ya que si a futuro se considera añadir una opción para permitir que los clientes cambien de regiones, Strategy hará que fácilmente sea posible.
3. El código es legible, contribuyendo a que su mantenimiento no se complique.

- Proxy.

Para proteger la seguridad de los clientes al momento de comprar en nuestra tienda, proxy fue el obvio candidato, ya que es muy útil por las siguientes razones:

1. Actúa como un mediador para controlar el acceso a las cuentas bancarias de los usuarios por medio de la autenticación, garantizando la seguridad de los datos bancarios.
2. Si la tienda comienza a tener más clientes registrados, en caso de problemas técnicos, proxy puede manejarlo sin afectar directamente al usuario ni a sus datos, fomentando la escalabilidad.
3. Con la misma razón que el punto anterior, en caso de comenzar a tener más tráfico de usuarios, proxy hace fácil reforzar la seguridad sin volver al código confuso.

- Iterator.

Para manejar al catálogo de productos que tenemos a la venta, nuestra elección fue Iterator porque:

1. Permite que el código sea más flexible ya que puede resistir a cambios en la estructura de nuestro catálogo.
2. Fomenta la legibilidad del código, ya que oculta los detalles de la implementación y lógica del recorrido del catálogo de productos, pero manteniendo la claridad de que es lo que hace, haciendo al código mantenible.

Usuarios.

Para poder comprar en nuestra tienda, los siguientes usuarios están disponibles:

- Usuario 1.

Usuario: OscarEvil

Contraseña: Oscar123

- Usuario 2.

Usuario: LeGringoLoco

Contraseña: ILoveGuns

- Usuario 3.

Usuario: Samuel777

Contraseña: Eyymuibuenasatoosguaapisimoss

- Usuario 4.

Usuario: TacoDeMole

Contraseña: MoleRojo

- Usuario 5.

Usuario: Iluminado

Contraseña: contraseña

- Usuario 6.

Usuario: Carrie

Contraseña: MrBig