

Proyecto 1

Equipo: Alfa Buena Maravilla Onda Dinamita Escuadrón Lobo.

Integrantes:

- ☐ Andrade Luviano Ximena - 320249150
- ☐ López Espinosa Ashley Yael - 319129391
- ☐ Villegas Maldonado César Ismael – 320108226

Modo de compilación.

Esta práctica fue realizada mediante el uso de paquetes de directorios, por lo que la manera de compilarla sería usando el comando “javac *.java”.

Modo de ejecución:

La ejecución de este proyecto se realizará mediante dos terminales ubicadas en la carpeta donde hayan decidido guardar el proyecto. Primeramente se compilará, y después en una terminal se ejecutará con el comando “java Server 1234”, y en la otra se ejecutará con el comando “java ClientManager”. De esta manera, la primera terminal ejecutará el servidor de la tienda, y la segunda ejecutará el servidor del cliente.

Versión de java utilizada.

La versión de java utilizada para la realización de esta práctica fue:

*java 16.0.1 2021-04-20
Java(TM) SE Runtime Environment (build 16.0.1+9-24)
Java HotSpot(TM) 64-Bit Server VM (build 16.0.1+9-24, mixed mode, sharing)*

Patrones usados:

- ☐ **Strategy:** Decidimos usar este patrón de diseño en la construcción de las tiendas virtuales (interfaz “VirtualStore”, clases “USAVirtualStore”, “MexicoVirtualStore” y “SpainVirtualStore”), de los manejadores de clientes (interfaz “ClientManagerInterface”, clases “ClientManagerMexico”, “ClientManagerUSA” y “ClientManagerSpain”) y de las conexiones de clientes (interfaz “ClientConnectionInterface”, clases “ClientConnectionMexico”, “ClientConnectionUSA”, “ClientConnectionSpain”).

Decidimos usar ésta implementación en vez de usar el patrón de diseño en vez de “Factory” ya que los métodos definidos en las interfaces se sobrescriben completamente ya que son las que hacen que cambie el idioma del programa, y realmente no tienen cosas en común como las tendrían con una súper clase.

- ☐ **Observer:** Decidimos usar este patrón en la implementación de la clase “Connection” para permitir el uso de lambdas, entonces la clase Connection actúa como el “Observable”, y sus escuchas, que en este caso son los clientes y el servidor serán los observadores. Creemos que permite tener un código más limpio.
- ☐ **Proxy:** Decidimos usar este patrón en la implementación de las clases “Client” y “ProxyClient”, la función de este patrón aquí es preservar los datos de nuestros clientes, y que no puedan ser modificados sus atributos más que el dinero que tienen en su cuenta, y así proteger sus datos personales. Las clases que usan al representante son “ClientConnection”, “ClientConnectionMexico”, “ClientConnectionUSA”, “ClientConnectionSpain”, “ClientManagerMexico”, “ClientManagerUSA”,

“ClientManagerSpain”, “MexicoVirtualStore”, “USAVirtualStore” y “SpainVirtualStore”; que son justamente las clases que interactúan entre sí para darle vida al proyecto.

- **Iterator:** Decidimos usar este patrón de diseño para cumplir con la instrucción de que solamente el servidor podrá tener acceso directo a la base de datos de los productos. Entonces para mostrarle al cliente todos los productos usamos un iterador 😊.
- **Socket:** Decidimos usar este patrón de diseño para darle vida al servidor de la tienda, y que la interacción con él sea algo más acercado a la realidad. Las clases “ClientConnection” y “StoreServer” se comunican a través de la clase “Connection”, que es la que utiliza a los sockets para manejar mensajes de entrada y salida. 😊

Método de compra:

Al momento de la compra, el usuario deberá de ingresar su cuenta de banco, si el usuario no tiene suficientes fondos en su cuenta de banco, el sistema automáticamente cancelará la compra y le volverá a mostrar las opciones para que el usuario pueda escoger si compra otra cosa, quita algo del carrito, etc.

Ejemplo de entradas:

Al momento de iniciar el programa se muestra una pantalla de escoger idioma, escogerán la opción 2 que es la opción para iniciar la tienda usando el español de México, luego escogerán la opción 1 que es la de iniciar sesión; el programa ya tiene precargados a dos usuarios de tienda mexicana:

- **César:**
 - Nombre de usuario: “villegassg”.
 - Contraseña: “proyecto1”.
 - Cuenta de banco: “1234567812345678”
- **Ximena:**
 - Nombre de usuario: “ximeanluv”.
 - Contraseña: “ximena123”.
 - Cuenta de banco: “1234123456785678”.