

Los Fósiles

17/04/2024

Integrantes:

- ◆ Delgado Díaz Hermes Alberto (319258613)
- ◆ Paredes Gonzalez Emiliano Sebastian (320047956)
- ◆ Quezada Ordoñez Kevin Steve (318331241)

Paquetes:

En este proyecto seccionamos el código en 5 paquetes dentro de la carpeta src, esto para una mejor organizacion de clases y mejor entendimiento. Los paquetes son

- ◆ *catalogo*
Este paquete contiene los siguientes archivos .java
 - Catalogo
 - ComponenteCatalogo
 - Departamento
 - DepartamentoAlimentos
 - DepartamentoElectrodomesticos
 - DepartamentoElectronica
 - Producto
- ◆ *cliente*
Este paquete contiene los siguientes archivos .java
 - Cliente
 - ClienteInterfaz
 - ClienteProxy
 - Observador
- ◆ *colors*
Este paquete contiene el siguiente archivo .java
 - Colors
- ◆ *idiomas*
Este paquete contiene los siguientes archivos .java
 - EspanolEspana
 - EspanolMexico

- Idioma
- Ingles
- ◆ *tienda*
Este paquete contiene los siguientes archivos .java
 - CheemsMart
 - Sujeto
- ◆ Dentro de la carpeta src se encuentra la clase **Main** en la cual se lleva a cabo la ejecución del programa

Patrones usados y justificación

- ◆ **Patrón Observer**
Este patrón se utilizó para poder enviarle las notificaciones sobre las ofertas de la tienda al usuario, en este caso el Cliente funge como observador ya que recibe las ofertas y la tienda funge como sujeto ya que es el que lanza estas notificaciones.
- ◆ **Patrón Proxy**
Este patrón se utilizó para el caso de la compra segura. Es decir, todos los procesos son mediante un intermediario entre el cliente real y el destino(Tienda), en este caso ClienteProxy es la "copia" de Cliente. De esta forma, los datos del usuario no serán vulnerables.
- ◆ **Patrón Composite**
En este caso, este patron no se usa del todo pero si su principio. Sirve para poder implementar a los departamentos dentro de los productos y los productos dentro de los departamentos.
- ◆ **Patrón Iterator**
Este patrón se utiliza para recorrer las listas de componentes en el caso de ComponenteCatalogo movernos entre departamentos y entre productos.
- ◆ **Patrón Singleton**
Este patrón se utiliza para crear una unica instancia del catálogo, esto sirve para no poder modificar el catálogo, ni crear ningun otro.
- ◆ **Patrón Strategy**
Este patrón se utilizó para que cada tienda de su respectivo país debe realizar ciertas acciones en el idioma adecuado dependiendo la nacionalidad del cliente. Realizan las mismas acciones, pero de distinta forma.

La justificación general de porque se usaron estos patrones es que en general queda clara la funcionalidad de cada parte y los requerimientos.

En el caso de la construcción de clases Departamento, no hay un patrón para esta parte, ya que se considero que la elaboración de Departamento con sus derivados es una simple herencia, donde no cambia ningún comportamiento, solo el nombre, por lo cual consideramos que no era necesario usar algún patrón.

Ejemplos para correr el programa

Estos son los usuarios que se crearon para probar el proyecto:

	<i>Usuario 1</i>	<i>Usuario 2</i>	<i>Usuario 3</i>
Nombre de usuario	<i>Hermes319B</i>	<i>Steve318J</i>	<i>Emiliano320T</i>
Contraseña	<i>hermes</i>	<i>steve</i>	<i>emiliano</i>
Nombre	<i>Hermes Alberto</i>	<i>Kevin Steve</i>	<i>Emiliano Sebastian</i>
Telefono	<i>55-2885-2875</i>	<i>38-3692-1857</i>	<i>67-1974-8357</i>
Dirección	<i>Calle 29 CDMX 26</i>	<i>Calle 17 Bilbao España 21</i>	<i>Street 29 California 20</i>
Cuenta de Banco	<i>MX39494JF</i>	<i>ES28946DM</i>	<i>EU85201KW</i>
Nacionalidad	<i>México</i>	<i>España</i>	<i>Estados Unidos</i>
Saldo	<i>29793.0</i>	<i>32500.0</i>	<i>18900.0</i>

Comandos

- ◆ Para compilar el programa

```
1 user:~/src$ javac Main.java
```

- ◆ Para ejecutar el programa

```
1 user:~/src$ java Main
```