



BianCommerce

eCommerce Platform

INGENIERIA DE SOFTWARE I - ITBA

GRUPO 6

Benenzon, Nicolás

Moynac, Hugo

Ritorto, Bianca

Santoflaminio, Alejandro

Torreguitar, José Martín

Introducción

El programa consiste en una plataforma de comercio electrónico. El mismo consiste en un prototipo. Se trata de una plataforma de comercio diseñada como si el cliente fuera un retailer de productos tecnológicos cuyo nombre es “Electro Store”.

Los productos y usuarios cargados en el programa se van guardando en archivos .txt cuyos datos son recuperados cuando el programa vuelve a iniciarse.

Se permite la creación de usuarios mediante el botón “Crear cuenta” con los datos que completa el usuario que luego van a ser guardados en el archivo.

Los usuarios registrados pueden entonces ingresar al sistema y cargar artículos en su carrito de compras. Finalmente pueden finalizar la compra eligiendo el método de pago.

Por otro lado para acceder como administrador basta con clickear el botón “Acceso admin” en la pantalla principal. Allí se puede eliminar usuarios, ver una lista con todos los usuarios registrados, cargar productos nuevos y modificar los productos ya cargados.

Por default el programa tiene seteada como contraseña de administrador “123456”, la cual se supone que se crea a pedido del cliente.

Uso del programa

Al iniciarse el programa, si el usuario ya estaba registrado el mismo puede ingresar sus datos e ingresar al sistema. Para ello debe escribir su correo y su contraseña. Por otro lado si no estaba registrado puede hacer click en “Crear cuenta” y completar con sus datos.



Figura 1: Pantalla inicial

Si en cambio, se hace click en “Acceso admin” se le pedirá una contraseña, la cual en este prototipo se configuró como “123456”. Una vez ingresado como administrador, el usuario puede eliminar o ver usuarios así como cargar o editar productos (eliminandolos o modificando su precio y stock). Para cargar productos es necesario completar todos los datos incluyendo la ubicación de la imagen. Si la misma luego no es encontrada se mostrará una imagen por default que dice “Image not found”.



Figura 2: Pantalla de acceso como administrador

Si se ingresa como usuario se puede proceder a comprar. Se puede acceder a las distintas categorías o bien buscar algo específico en la barra de búsqueda. También el usuario puede editar sus datos. Al hacer esto si había algo cargado en su carrito se borrara.

Salir

Ingrese el nombre del artículo a buscar o seleccione una categoría

Usuario:
Mariano Lopez

Editar usuario

Buscar



Carro de compras

Categorías



Celulares



Notebooks



Televisores



Auriculares



Tablets



Cámaras

Figura 3: Ingreso como usuario

Seleccione los productos que desea agregar a su carrito

Para salir de esta categoría cierre la ventana

Tablets - iPads y tablets android



Nombre	Descripcion	Imagen	Precio
Apple iPad Pro 10.5	Space Gray 64gb		23000.0
Apple iPad	Silver 32gb		9500.0
Huawei Mediapad T3	7 pulgadas. Negro.		2000.0

Figura 4: Contenido a modo de ejemplo de la categoría Tablets

3

Finalmente cuando el usuario terminó de comprar puede hacer click en “Carro de compras” y proceder a abonar del modo que desee.



Figura 5: Carrito de compras

Testing mediante CodeClimate

Para testear el programa se decidió utilizar el framework de testeo presentado por el grupo durante la materia: CodeClimate.

A continuación se muestran los resultados obtenidos para la versión final del programa. Previamente ya se intentó minimizar la cantidad de problemas al mínimo que se consideró adecuado para la versión final del producto.

Resultados generales: Vemos que el programa obtiene una calificación relativamente alta (B) con 40 casos de código duplicado y 16 posibles problemas en el código.



Figura 6: Resultados generales de CodeClimate

Código repetido: Vemos que hay 40 casos de código repetido. Esto se debe a que hay clases que son similares. Por ejemplo en la figura 2 se observa que las clases SearchTable.java y ProductTable.java tienen código repetido. Esto sucede debido a que las clases son prácticamente similares, una muestra los productos que corresponden a cierta categoría y otra los productos que corresponden a la búsqueda realizada por el usuario.

Identical blocks of code found in 2 locations. Consider refactoring.

OPEN

```

61         for(Product p: productsHere) {
62             data[i][0] = p.getName();
63             data[i][1] = p.getDescription()    ;
64             if(this.getClass().getResource(p.getPhotoPath()) == null) {
65                 Image img = new ImageIcon(this.getClass().getResource("/noimage.jpeg")).getIma

```

Found in src/Model/SearchTable.java and 1 other location - About 2 hrs to fix

src/Model/ProductTable.java on lines 63..76

Figura 7: Code Smells

Problemas de código: Los 16 problemas de código mencionados anteriormente tienen que ver con el largo de los métodos y la cantidad de métodos que tienen. Esto ocurre por la complejidad del programa y porque algunas clases son más complejas que otras.

Method `initialize` has 139 lines of code (exceeds 25 allowed). Consider refactoring.

[OPEN](#)

```
79     private void initialize() throws IOException {  
80         frame = new JFrame();  
81         frame.setResizable(false);  
82         frame.setBackground(UIManager.getColor("Desktop.background"));  
83         frame.getContentPane().setBackground(UIManager.getColor("Desktop.background"));
```

Found in `src/Model/Window.java` - About 5 hrs to fix

`User` has 25 methods (exceeds 20 allowed). Consider refactoring.

[OPEN](#)

```
6     public class User implements Cloneable{  
7         private String name;  
8         private String mail;  
9         private int dni;  
10        private String cellphone;
```

Found in `src/Model/User.java` - About 2 hrs to fix

Figura 8: Duplicación de código