Breve manual para guiar la corrección

Elabora un breve documento que sirva como guía para la revisión de la funcionalidad de la aplicación. Proporciona credenciales (username/password) de ejemplo y explicita en qué partes del proyecto se han utilizado hilos y recursividad.

Este documento estará en formato PDF en la carpeta "**doc**" de tu proyecto.

Guía de Revisión de la Funcionalidad de la Aplicación.

Pasos para Revisar la Aplicación

Antes de nada, las credenciales.

Administrador:

- Usuario: admin
- Contraseña: 1234

Cliente:

- Usuario: cliente
- Contraseña: 5678

Otros usuarios:

- Usuario: jon
  Contraseña: 1234
- Usuario: prueba1
  Contraseña: prueba

Si eso intenta introducir credenciales incorrectas y comprueba si aparece un mensaje de error.

- Navegación de la Aplicación

Explora las diferentes secciones: categorías, carrito, historial, etc. Comprobar que todos los botones y enlaces funcionan y la interfaz sea clara y sencilla

- Carrito de la Compra

Añade muchos artículos a la cesta y asegúrate de que aparecen con la cantidad correcta.

Mira si el total del carrito se calcula correctamente, teniendo en cuenta los precios y los descuentos.

Intenta eliminar artículos del stock o vaciar la cesta y verifica si se refleja bien.

- Categorías y Productos

Entra en la sección de categorías y productos y…

Si eres administrador, intenta añadir, editar y eliminar categorías y productos.

Comprueba que las modificaciones se manifiestan tanto en el frontend como en la tabla de la base de datos.

- Historial de Compras

Accede al historial de compras de un usuario para verificar que las compras realizadas se registran correctamente.

Comprueba que la información coincide con las transacciones reales.

- Panel de Administración (solo para admins)

Si inicias sesión con una cuenta de administrador, abre el panel de administración y realiza pruebas de gestión de usuarios y comprueba que los cambios son permanentes.

HILOS

Se utiliza un hilo para inicializar la base de datos en el método main:

```
Thread initThread = new Thread(() -> {
    servicioPersistenciaBD = ServicioPersistenciaBD.getInstance();
    if (servicioPersistenciaBD.init("supermarket.db")) {
                System.out.println("Base de datos inicializada correctamente.");
    } else {
            System.err.println("Error al inicializar la base de datos.");
        return;
    }
});
initThread.start();
```

El hilo se usa para que la base de datos se inicie en segundo plano, mientras el resto del programa puede seguir funcionando sin quedarse "congelado". Una vez que termina de prepararse la base de datos, el programa principal espera a que ese proceso termine (usando initThread.join()) antes de mostrar la ventana de la aplicación.

RECURSIVIDAD

En la clase CarritoCompras se utiliza recursividad en **dos** métodos:

1-Cálculo del total de precios de productos:

(calcularTotalRecursivo y calcularTotal)

Método recursivo para calcular el total de los precios de los productos en el carrito:

```
public double calcularTotalRecursivo() {
    return calcularTotal(new ArrayList<>(productos.entrySet()), 0);
}

private double calcularTotal(List<Map.Entry<Producto, Integer>>
items, int index) {
    if (index >= items.size()) {
        return 0;
    }
    Map.Entry<Producto, Integer> item = items.get(index);
    double subtotal = item.getKey().getPrecio() * item.getValue();
    return subtotal + calcularTotal(items, index + 1);
}
```

2-Contar la cantidad total de productos:

(contarProductos y contarProductosTotales)

Implementa un método recursivo para sumar la cantidad total de productos:

```
public int contarProductos(List<Map.Entry<Producto, Integer>> items,
int index) {
    if (index >= items.size()) {
        return 0;
    }
    int cantidad = items.get(index).getValue();
    return cantidad + contarProductos(items, index + 1);
}


public int contarProductosTotales() {
      return  contarProductos(new  ArrayList<>(productos.entrySet()),
0);
}
```