

Explicacion Detallada de Metodos en Java

1. Connect()

Propósito: Establece la conexión con la base de datos MySQL.

Descripción:

- Carga el controlador JDBC usando Class.forName().
- Establece la conexión mediante DriverManager.getConnection().
- Maneja posibles excepciones para asegurar que errores se registren en el log.

2. CargarProductoPorNumero()

Propósito: Llena el combo box txtId con los IDs de los productos disponibles.

Descripción:

- Ejecuta una consulta SELECT para obtener todos los ID de productos.
- Limpia el combo box antes de llenarlo.
- Agrega cada ID obtenido al combo box.

3. Fetch()

Propósito: Recupera todos los registros de productos y los muestra en jTable1.

Descripción:

- Ejecuta una consulta SELECT * para recuperar todos los productos.
- Usa ResultSetMetaData para obtener el número de columnas.

Explicacion Detallada de Metodos en Java

- Limpia la tabla y llena cada fila de la tabla con los datos obtenidos.

4. btnSearchActionPerformed()

Propósito: Busca un producto específico por su ID seleccionado.

Descripción:

- Prepara y ejecuta una consulta SELECT para obtener un producto por ID.
- Si encuentra un resultado, llena los campos de nombre, precio y cantidad.
- Muestra un mensaje de error si no se encuentra el producto.

5. btnDeleteActionPerformed()

Propósito: Elimina un producto por su ID seleccionado.

Descripción:

- Ejecuta una sentencia DELETE para eliminar el producto con el ID seleccionado.
- Muestra un mensaje de éxito y actualiza la tabla tras la eliminación exitosa.

6. txtNameActionPerformed() y txtQtyActionPerformed()

Propósito: Métodos generados automáticamente para manejar eventos de acción en los campos de texto.

Descripción:

Explicacion Detallada de Metodos en Java

- No contienen lógica, pero pueden ser utilizados para personalizar eventos como cambios en los campos de texto.

7. btnAddActionPerformed()

Propósito: Añade un nuevo producto a la base de datos.

Descripción:

- Recupera los datos de nombre, precio y cantidad de los campos de texto.
- Inserta los datos en la base de datos con una consulta INSERT.
- Limpia los campos de texto y actualiza la tabla tras una inserción exitosa.

8. btnUpdateActionPerformed()

Propósito: Actualiza un producto existente en la base de datos.

Descripción:

- Recupera los datos actualizados y el ID del producto.
- Ejecuta una consulta UPDATE para modificar los datos del producto.
- Muestra un mensaje de éxito y actualiza la tabla tras la actualización exitosa.

9. btnCSVActionPerformed()

Explicacion Detallada de Metodos en Java

Propósito: Exporta los datos de los productos a un archivo CSV.

Descripción:

- Crea un archivo CSV y escribe los datos de los productos en él.
- Muestra un mensaje de éxito al finalizar la exportación.