

## DESARROLLO DE INTERFACES

UD4. DISEÑO INTERFACES. Netbeans

2CFGS

DAM

# Actividad 10. Simulacro 2.0



Nombre: Andreu Sanz Sanz

CFGS DISEÑO APLICACIONES MULTIPLATAFORMA

Módulo: 0488 – Desarrollo de Interfaces

UD4: Diseño interfaces. Netbeans

# MEMORIA ACTIVIDAD 10

## 1. INTRODUCCIÓN/OBJETIVO

*Se va a desarrollar una aplicación que haga uso de AbstractTableModel. He optado por extender la práctica para repasar para el próximo examen. Esta aplicación simulará una tienda online de una empresa llamada Solterra Energía, especializada en la instalación y venta de placas solares y baterías.*

## 2. ELEMENTOS UTILIZADOS

### PantallaPrincipal

- info :: actividad\_10.gui.SobreNosotros
- img :: actividad\_10.gui.imagenes
- prd :: actividad\_10.gui.productos
- ped :: actividad\_10.gui.pedidos
- jPanel5 :: JPanel
- jPanel3 :: JPanel
- jTextFieldProduct :: JTextField
- jLabel3 :: JLabel
- jPanel4 :: JPanel
- jTextFieldCant :: JTextField
- jLabel4 :: JLabel
- jButtonPedir :: JButton
- jMenuBar1 :: JMenuBar
- jMenu1 :: JMenu
- jMenuItemInstalacion\_01 :: JMenuItem
- jMenuItemInstalacion\_02 :: JMenuItem
- jMenuItemInstalacion\_03 :: JMenuItem
- jMenuProducts :: JMenu
- jMenuItem5 :: JMenuItem
- jMenuItemPedidos :: JMenuItem
- jMenuItemInfo :: JMenu
- jMenuItemInfo :: JMenuItem

### Imágenes

- panellImaguen :: actividad\_10.gui.PanellImagen

### Pedidos

- jTableTabla :: JTable
- jScrollPane1 :: JScrollPane

### Productos

- jTableTabla :: JTable
- jScrollPane1 :: JScrollPane

### SobreNosotros

- PanellImagenLogo :: actividad\_10.gui.PanellImagen
- jPanel1 :: JPanel
- jTextArea :: JTextArea
- jScrollPane1 :: JScrollPane

### 3. FUNCIONAMIENTO

Voy a explicar el funcionamiento de esta práctica estructurada por paquetes.

#### Paquete 'dto'

En este paquete, se encuentran dos clases: 'Pedido' y 'Producto'. La clase 'Pedido' tiene dos campos, 'nombre' y 'cantidad', mientras que la clase 'Producto' incluye 'id', 'nombre' y 'descripción'. Ambas clases cuentan con sus respectivos métodos 'getters' y 'setters', además del constructor.

#### Paquete 'lógica'

En este paquete, se encuentra exclusivamente la lógica para la lista de productos. Se ha decidido crear una tabla similar a la del video y otra tabla normal, para disponer de ambos tipos de tablas.

#### Paquete 'img'

Este paquete alberga las fotos utilizadas en el proyecto.

#### Paquete 'gui'

En este paquete, se encuentran todos los archivos necesarios para mostrar la interfaz. Más adelante, explicaré las ventanas. Dentro de este paquete, se encuentra otro llamado 'tablemodels', que contiene la clase 'ProductosTableModel'. Esta clase nos sirve para tener modelos de tablas de productos y es similar a la del video.

#### Ahora, voy a explicar las ventanas:

##### 1. Ventana Principal:

En esta ventana, gracias a un menú, podemos navegar por otras ventanas. Además, incluye dos campos de texto para introducir los datos de un pedido y añadirlos a la tabla de pedidos al presionar un botón.

##### 2. Ventana de Imágenes:

Esta ventana muestra únicamente una imagen, dependiendo de la opción seleccionada en el menú general. Hace referencia a imágenes de instalaciones para que el cliente las pueda visualizar.

##### 3. Ventana de Productos de la Empresa:

Accesible desde el menú de productos > Nuestros productos. En esta ventana, se muestra una tabla creada a partir de la clase 'ProductosTableModel', exhibiendo al cliente los productos de la empresa.

##### 4. Ventana de Productos Añadidos:

Muestra exclusivamente los productos que el cliente ha añadido desde la ventana principal.

##### 5. Ventana de Información de la Empresa

Consta de una foto del logo de la empresa y un área de texto para presentar una breve definición de la empresa.

## 4. CAPTURAS

### Design PantallaPrincipal

Solterra Enegia

Instalaciones Productos Solterra

Nombre de el producto:

Cantidad:

Pedir

### Design imagenes



### Design productos

ID	Nombre	Descripción
P001	Panel Solar 100W	Panel solar monocristalino de 100W
P002	Batería de Litio 12V	Batería recargable de litio de 12V, 200Ah
P003	Inversor Solar 3000W	Inversor solar de onda sinusoidal pura, 3000W
P004	Regulador de Carga 20A	Regulador de carga solar PWM de 20A
P005	Cable Solar 5 metros	Cable solar fotovoltaico de 5 metros

### Design pedidos

Nombre	Cantidad
Batería	3
Panel Solar	7

## Design SobreNosotros



## Source PantallaPrincipal

```
SobreNosotros info;
imagenes img;
productos prd;
pedidos ped;

public pantallaPrincipal() {
    initComponents();
    this.setIconImage(image: new ImageIcon(location: getClass().getResource(name: "/actividad_10/img/Favicon.png")).getImage());
    info = new SobreNosotros(parent: this, modal: true);
    img = new imagenes(parent: this, modal: true);
    prd = new productos(parent: this, modal: true);
    ped = new pedidos(parent: this, modal: true);
}
```

```
private void jMenuItemInstalacion_02ActionPerformed(java.awt.event.ActionEvent evt) {
    img.setTitle(title: "Instalación de Raquel");
    img.setimg(new File(pathname: "./src/actividad_10/img/casa.png"));
    img.setVisible(b: true);
}

private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
    prd.setVisible(b: true);
}

private void jMenuItemPedidosActionPerformed(java.awt.event.ActionEvent evt) {
    ped.setVisible(b: true);
}

private void jMenuItemInfoActionPerformed(java.awt.event.ActionEvent evt) {
    info.setVisible(b: true);
}

private void jMenuItemInstalacion_01ActionPerformed(java.awt.event.ActionEvent evt) {
    img.setTitle(title: "Instalación de Cristhian");
    img.setimg(new File(pathname: "./src/actividad_10/img/casaconplacas.jpg"));
    img.setVisible(b: true);
}
```

```
private void jButtonPedirActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String name = jTextFieldProduct.getText();
        int cant = Integer.parseInt(jTextFieldCant.getText());
        if (name.isBlank()) {
            JOptionPane.showMessageDialog(parentComponent: this, message: "La producto no puede estar vacio", title: "Error", messageType: JOptionPane.ERROR_MESSAGE);
        } else {
            if (cant <= 0) {
                JOptionPane.showMessageDialog(parentComponent: this, message: "La cantidad debe ser mayor que cero", title: "Error", messageType: JOptionPane.ERROR_MESSAGE);
            } else {
                ped.anadirPedido(new Pedido(nombre: name, cantidad: cant));
            }
            JOptionPane.showMessageDialog(parentComponent: this, message: "El producto se a añadido", title: "OK", messageType: JOptionPane.INFORMATION_MESSAGE);
            jTextFieldProduct.setText(s: null);
            jTextFieldCant.setText(s: null);
        }
    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(parentComponent: this, message: "Ingrese una cantidad válida", title: "Error", messageType: JOptionPane.ERROR_MESSAGE);
    } catch (Exception e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(parentComponent: this, message: "Ocurrió un error inesperado", title: "Error", messageType: JOptionPane.ERROR_MESSAGE);
    }
}

private void jMenuItemInstalacion_03ActionPerformed(java.awt.event.ActionEvent evt) {
    img.setTitle(title: "Instalación de Raul");
    img.setimg(new File(pathname: "./src/actividad_10/img/instalacion.png"));
    img.setVisible(b: true);
}
```

## Source *imagenes*

```
    */
    public imagenes(java.awt.Frame parent, boolean modal) {
        super(owner:parent, modal);
        initComponents();
        seting(new File(pathname: "../src/actividad_10/img/casa.png"));
    }

    public void seting(File path){
        panelImaguen.setPath(path);
    }
}
```

## Source *productos*

```
private logicaSolterra logicaSolterra = new logicaSolterra();
public productos(java.awt.Frame parent, boolean modal) {
    super(owner:parent, modal);
    initComponents();
    inicializarTabla();
}

private void inicializarTabla() {
    ProductosTableModel tmp = new ProductosTableModel(listaProductos:logicaSolterra.getListProductos());
    jTableTabla.setModel(dataModel:tmp);

    TableRowSorter<ProductosTableModel> sorter = new TableRowSorter<>(model:tmp);
    jTableTabla.setRowSorter(sorter);

    List<SortKey> sortKey = new ArrayList<>();
    sortKey.add(new SortKey(column: 0,sortOrder:SortOrder.ASCENDING));
    sorter.setSortKeys(sortKeys: sortKey);
}

public void anadirProducto(Producto p) {
    DefaultTableModel dtm = (DefaultTableModel) jTableTabla.getModel();

    // Agregar una nueva fila
    String[] rowData = new String[dtm.getColumnCount()];
    rowData[0] = p.getId(); // Supongamos que quieres agregar el ID del producto en la primera columna
    rowData[1] = p.getNombre();
    rowData[2] = p.getDescripcion();

    dtm.addRow(rowData);
}
```

## Source *Producto*

```
public class Producto {
    private String id;
    private String nombre;
    private String descripcion;

    public Producto(String id, String nombre, String descripcion) {
        this.id = id;
        this.nombre = nombre;
        this.descripcion = descripcion;
    }

    public String getId() {
        return id;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
}
```

## Source logicaSolterra

```
public class logicaSolterra {
    List<Producto> listProductos = new ArrayList<>();

    public logicaSolterra() {
        listProductos = new ArrayList<>();
        listProductos.add(new Producto(id: "P001", nombre: "Panel Solar 100W", descripcion: "Panel solar monocristalino de 100W"));
        listProductos.add(new Producto(id: "P002", nombre: "Bateria de Litio 12V", descripcion: "Bateria recargable de litio de 12V, 200Ah"));
        listProductos.add(new Producto(id: "P003", nombre: "Inversor Solar 3000W", descripcion: "Inversor solar de onda sinusoidal pura, 3000W"));
        listProductos.add(new Producto(id: "P004", nombre: "Regulador de Carga 20A", descripcion: "Regulador de carga solar PWM de 20A"));
        listProductos.add(new Producto(id: "P005", nombre: "Cable Solar 5 metros", descripcion: "Cable solar fotovoltaico de 5 metros"));
    }

    public List<Producto> getListProductos() {
        return listProductos;
    }
}
```

## Source ProductosTableModel

```
List<Producto> listaProductos; //
String[] columnas = {"ID", "Nombre", "Descripción"};

public ProductosTableModel(List<Producto> listaProductos) {
    this.listaProductos = listaProductos;
}

@Override
public int getRowCount() {
    return this.listaProductos.size();
}

@Override
public int getColumnCount() {
    return this.columnas.length;
}

@Override
public Object getValueAt(int rowIndex, int columnIndex) {
    switch (columnIndex) {
        case 0:
            return listaProductos.get(index: rowIndex).getId();
        case 1:
            return listaProductos.get(index: rowIndex).getNombre();
        case 2:
            return listaProductos.get(index: rowIndex).getDescripcion();
        default:
            return null;
    }
}

@Override
public String getColumnName(int column) {
    return columnas[column];
}
```

## Source pedidos

```
public pedidos(java.awt.Frame parent, boolean modal) {
    super(owner: parent, modal);
    initComponents();
    inicializarTabla();
}

private void inicializarTabla() {
    DefaultTableModel dtm = new DefaultTableModel();
    dtm.setColumnIdentifiers(new String[]{"Nombre", "Cantidad"});
    jTableTabla.setModel(dataModel: dtm);
}

public void anadirPedido(Pedido p) {
    DefaultTableModel dtm = (DefaultTableModel) jTableTabla.getModel();

    // Agregar una nueva fila
    String[] rowData = new String[dtm.getColumnCount()];
    rowData[0] = p.getNombre();
    rowData[1] = String.valueOf(p.getCatidad());
    dtm.addRow(rowData);
}
```

### Source Pedido

```
private String nombre;
private int catidad;

public Pedido(String nombre, int catidad) {
    this.nombre = nombre;
    this.catidad = catidad;
}

public int getCatidad() {
    return catidad;
}

public void setCatidad(int catidad) {
    this.catidad = catidad;
}

public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}
```

### Source SobreNosotros

```
*/
public SobreNosotros(java.awt.Frame parent, boolean modal) {
    super(owner:parent, modal);
    initComponents();
    PanelImagenLogo.setPath(new File(pathname: "../src/actividad_10/img/LogoLateral.png"));
}
```