

## DESARROLLO DE INTERFACES

UD4. DISEÑO INTERFACES. Netbeans

2CFGS

DAM

# Actividad VII. Alta Tienda Online.



Nombre: Andreu Sanz Sanz

CFGS DISEÑO APLICACIONES MULTIPLATAFORMA

Módulo: 0488 – Desarrollo de Interfaces

UD4: Diseño interfaces. Netbeans

# MEMORIA ACTIVIDAD VII

## 1. INTRODUCCIÓN/OBJETIVO

*Se ve a crear una interfaz simple para registrarse e iniciar sesión*

## 2. ELEMENTOS UTILIZADOS

### **PantallaPrincipal :: JFrame**

jLabel1 :: JLabel

jButtonReg :: JButton

jButtonLog :: JButton

reg :: Registro

log :: InicioSession

usuarios :: HashMap<String, String>

### **InicioSession :: JDialog**

jPasswordFieldPass :: JPasswordField

jLabelUser :: JLabel

jTextFieldUser :: JTextField

jLabelPass :: JLabel

jButtonLog :: JButton

validationPanel1 :: ValidationPanel

### **Registro :: JDialog**

jLabel2 :: JLabel

jTextFieldNombre :: JTextField

jLabel1 :: JLabel

jTextFieldApellidos :: JTextField

jLabel3 :: JLabel

jTextFieldNombre\_Usuario :: JTextField

jLabel4 :: JLabel

jPasswordFieldPassword :: JPasswordField

jLabel5 :: JLabel

jTextFieldCalle :: JTextField

jLabel6 :: JLabel

jTextFieldPuerta :: JTextField

jLabel7 :: JLabel

jTextFieldPiso :: JTextField

jLabel8 :: JLabel

jTextFieldCodigo\_Postal :: JTextField

jLabel9 :: JLabel

jTextFieldTelefono :: JTextField

jLabel10 :: JLabel

jTextFieldEmail :: JTextField

jLabel11 :: JLabel

jButtonRegister :: JButton

validationPanel1 :: ValidationPanel

### 3. FUNCIONAMIENTO

Esta aplicación consta de tres ventanas. En la primera, observamos un título y dos botones; estos últimos redirigen al usuario hacia la ventana de registro o a la ventana de inicio de sesión.

En la ventana de registro, se presentan varios campos que deben completarse. Una vez que todos estos campos han sido debidamente llenados y se ha pulsado el botón de registro, el usuario registrado se añade a un hashMap, tal como se define en la pantalla principal. La gestión de este hashMap sigue un enfoque similar al de la práctica anterior.

En la ventana de inicio de sesión, se encuentran dos campos a completar y un botón para llevar a cabo dicha acción. Este botón verifica la existencia de los campos en el hashMap y muestra un mensaje apropiado si los datos son correctos. En caso de datos erróneos, se mostrará un mensaje de error.

Para asegurar que todos los campos se han completado correctamente, hacemos uso de nuestra librería personalizada y del objeto validationPanel.

### 4. CAPTURAS

#### *Pantalla Principal Source*

```
Registro reg = new Registro(parent: this, modal: true);
InicioSession log = new InicioSession(parent: this, modal: true);
public PantallaPrincipal() {
    initComponents();
}
```

```
private void jButtonRegActionPerformed(java.awt.event.ActionEvent evt) {
    reg.setVisible(b: true);
}

private void jButtonLogActionPerformed(java.awt.event.ActionEvent evt) {
    log.setVisible(b: true);
}
```

```
private HashMap<String, String> usuarios = new HashMap<>();

public void anadirUsuario(String nombre, String contrasena) {
    usuarios.put(key: nombre, value: contrasena);
}

public boolean buscarUsuario(String nombre, String contrasena) {
    if (usuarios.containsKey(key: nombre)) {
        String password = usuarios.get(key: nombre);
        return password != null && password.equals(anObject: contrasena);
    }
    return false;
}
```

## Pantalla de Registro Source

```
PantallaPrincipal p;  
public Registro(java.awt.Frame parent, boolean modal) {  
    super(owner: parent, modal);  
    initComponents();  
    p = (PantallaPrincipal) parent;  
    ValidationGroup group = validationPanell.getValidationGroup();  
    group.add(comp: jTextFieldNombre, validators: StringValidators.REQUIRE_NON_EMPTY_STRING);  
    group.add(comp: jTextFieldNombre_Usuario, validators: StringValidators.REQUIRE_NON_EMPTY_STRING, validators: StringValidators.NO_WHITESPACE);  
    group.add(comp: jPasswordFieldPassword, validators: StringValidators.REQUIRE_NON_EMPTY_STRING, validators: StringValidators.NO_WHITESPACE);  
    group.add(comp: jTextFieldCalle, validators: StringValidators.REQUIRE_NON_EMPTY_STRING);  
    group.add(comp: jTextFieldPuerta, validators: StringValidators.REQUIRE_VALID_INTEGER);  
    group.add(comp: jTextFieldPiso, validators: StringValidators.REQUIRE_VALID_INTEGER);  
    group.add(comp: jTextFieldCodigo_Postal, validators: StringValidators.REQUIRE_VALID_INTEGER, validators: StringValidators.minLength(length:5), validators: StringValidators.maxLength(length:5));  
    group.add(comp: jTextFieldTelefono, validators: StringValidators.REQUIRE_VALID_INTEGER, validators: StringValidators.minLength(length:9), validators: StringValidators.maxLength(length:9));  
    group.add(comp: jTextFieldEmail, validators: StringValidators.EMAIL_ADDRESS, validators: StringValidators.NO_WHITESPACE);  
  
    validationPanell.addChangeListener(((ChangeEvent e) -> {  
        if (validationPanell.getProblem() == null) {  
            jButtonRegistrar.setEnabled(b: true);  
        }else{  
            jButtonRegistrar.setEnabled(b: false);  
        }  
    }));  
}
```

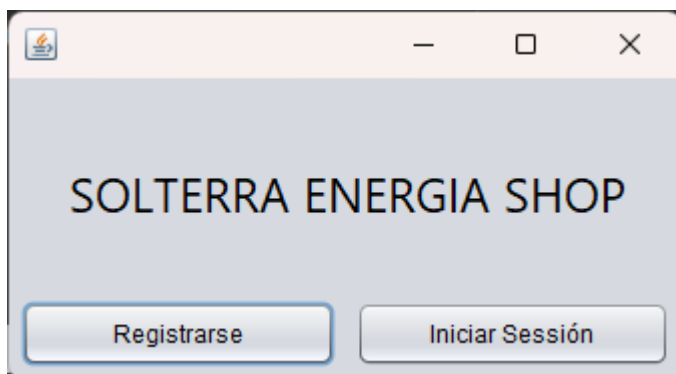
```
private void jButtonRegistrarActionPerformed(java.awt.event.ActionEvent evt) {  
    p.anadirUsuario(  
        nombre: jTextFieldNombre_Usuario.getText(),  
        new String(value: jPasswordFieldPassword.getPassword())  
    );  
    dispose();  
}
```

## Pantalla de Inicio Sesión Source

```
PantallaPrincipal p;  
public InicioSession(java.awt.Frame parent, boolean modal) {  
    super(owner: parent, modal);  
    initComponents();  
    p = (PantallaPrincipal) parent;  
    ValidationGroup group = validationPanell.getValidationGroup();  
  
    group.add(comp: jTextFieldUser, validators: StringValidators.REQUIRE_NON_EMPTY_STRING, validators: StringValidators.NO_WHITESPACE);  
    group.add(comp: jPasswordFieldPass, validators: StringValidators.REQUIRE_NON_EMPTY_STRING, validators: StringValidators.NO_WHITESPACE);  
  
    validationPanell.addChangeListener(((ChangeEvent e) -> {  
        if (validationPanell.getProblem() == null) {  
            jButtonLog.setEnabled(b: true);  
        }else{  
            jButtonLog.setEnabled(b: false);  
        }  
    }));  
}
```

```
private void jButtonLogActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    boolean user = p.buscarUsuario(nombre: jTextFieldUser.getText(), new String(value: jPasswordFieldPass.getPassword()));  
    if (user) {  
        JOptionPane.showMessageDialog(parentComponent: this, message: "El usuario se ha registrado correctamente", title: "Login correcto", messageType: JOptionPane.INFORMATION_MESSAGE);  
    }else {  
        JOptionPane.showMessageDialog(parentComponent: this, message: "El usuario no esta registrado", title: "Login fallido", messageType: JOptionPane.ERROR_MESSAGE);  
    }  
}
```

## Pantalla Principal Design



### ***Pantalla de Registro Design***

Completa nuestro formulario:

Nombre <input type="text"/>	Puerta <input type="text"/>
Apellidos <input type="text"/>	Piso <input type="text"/>
Nombre usuario <input type="text"/>	Código Postal <input type="text"/>
Contraseña <input type="password"/>	Número teléfono <input type="text"/>
Calle <input type="text"/>	Correo electrónico <input type="text"/>

❗ Name missing from email address

Registrarse

### ***Pantalla de Inicio Sesión Design***

Nombre de usuario:

Contraseña:

❗ null may not be empty

Entrar