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 = validationPanel1.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);

    validationPanel1.addChangeListener(((ChangeEvent e) -> {
        if (validationPanel1.getProblem() == null) {
            jButtonRegister.setEnabled(b: true);
        }else{
            jButtonRegister.setEnabled(b: false);
        }
    }));

}
```

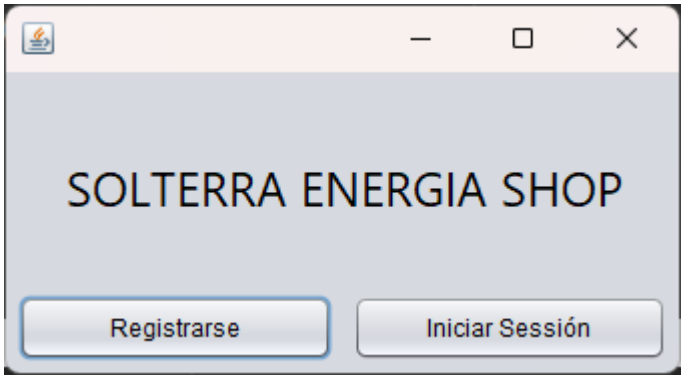
```
private void jButtonRegisterActionPerformed(java.awt.event.ActionEvent evt) {
    p.anadirUsuario(
            nombre:jTextFieldNombre_Usuario.getText(),
            new String(value: jPasswordFieldPassword.getPassword())
    );
    dispose();
}
```

Pantalla de Inicio Sessión Source

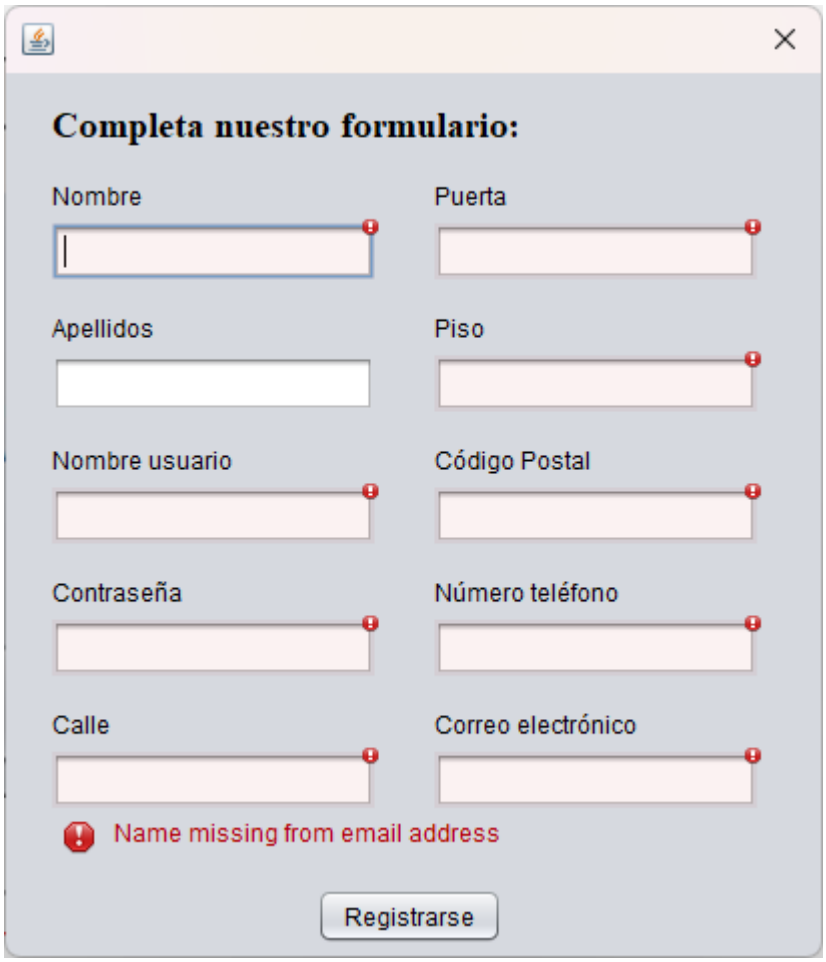
```
PantallaPrincipal p;
public InicioSession(java.awt.Frame parent, boolean modal) {
    super(owner:parent, modal);
    initComponents();
    p = (PantallaPrincipal) parent;
    ValidationGroup group = validationPanel1.getValidationGroup();

    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);//?


    validationPanel1.addChangeListener(((ChangeEvent e) -> {
        if (validationPanel1.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 está registrado", title:"Login fallido", messageType:JOptionPane.ERROR_MESSAGE);
    }
}
```

Pantalla Principal Design



Pantalla de Registro Design



Pantalla de Inicio Sesión Design