

Swing 12 - Introducción a las Múltiples Ventanas

En esta práctica se agregará una nueva ventana con la funcionalidad de login, en caso de ser incorrecta, se mostrará en un panel emergente un mensaje indicándolo y en caso de ser adecuada, se presentará la ventana de la aplicación principal.

Se comenzará cargando el proyecto de Navegación con Menú. Se agregará un nuevo **JFrame** llamado **LoginFrm** en cuál contendrá los siguientes elementos:

- Etiqueta "Validación de Usuario" (**lblTitulo**)
- Panel con borde y título "Login" (**pnlContenedor**)
- Etiqueta "Usuario" (**lblUsuario**)
- Etiqueta "Contraseña" (**lblPassword**)
- Boton "Validar" (**btnValidar**)

Se indicará que también se mostrará maximizado.

LoginFrm.java

```
public LoginFrm() {  
    initComponents();  
    this.setExtendedState(MAXIMIZED_BOTH);  
}
```

Se agregarán dos nuevos paquetes: **uam.pvoe.sw.menu.operaciones** y **uam.pvoe.sw.menu.clases** para simular el proceso de validación.

Usuario.java

```
public class Usuario {  
  
    private String login;  
    private String password;  
  
    public String getLogin() {  
        return login;  
    }  
    public void setLogin(String login) {  
        this.login = login;  
    }  
    public String getPassword() {  
        return password;  
    }  
    public void setPassword(String password) {  
        this.password = password;  
    }  
}
```

ValidarUsuario.java

```
public class ValidarUsuario {

    public boolean validarUsuario(Usuario usr) {
        if(usr.getLogin().compareTo("31749")==0) {
            if(usr.getPassword().compareTo("31749")==0) {
                return true;
            }else{
                return false;
            }
        }else{
            return false;
        }
    }
}
```

LoginFrm.java

```
private void btnValidarActionPerformed(java.awt.event.ActionEvent evt) {

    String usuario = txtUsuario.getText();
    char [] password = pwdPassword.getPassword();
    String pwd = new String(password);

    Usuario usuarioValidar = new Usuario();
    usuarioValidar.setLogin(usuario);
    usuarioValidar.setPassword(pwd);

    if(validarUsuario(usuarioValidar)){
        System.out.println("Usuario Válido");
    }else{
        System.out.println("Login o Password incorrecto");
    }
}
```

LoginFrm.java

```
private boolean validarUsuario(Usuario usr) {
    ValidarUsuario validar = new ValidarUsuario();
    boolean resultado;

    resultado = validar.validarUsuario(usr);
    return resultado;
}
```

Desplegando Mensaje Emergente

En caso de que la validación no sea correcta, se indicará con un mensaje emergente.

LoginFrm.java

```
private void btnValidarActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String usuario = txtUsuario.getText();  
    char [] password = pwdPassword.getPassword();  
    String pwd = new String(password);  
    Usuario usuarioValidar = new Usuario();  
    usuarioValidar.setLogin(usuario);  
    usuarioValidar.setPassword(pwd);  
  
    if(validarUsuario(usuarioValidar)){  
        System.out.println("Usuario Válido");  
    }else{  
        System.out.println("Login o Password incorrecto");  
        JOptionPane.showMessageDialog(this, "El Login o Password son  
incorrectos", "Validación Incorrecta", JOptionPane.ERROR_MESSAGE);  
    }  
}
```

Finalmente se cambiará al contenedor principal en caso de que la validación sea correcta.

LoginFrm.java

```
private void btnValidarActionPerformed(java.awt.event.ActionEvent  
evt) {  
  
    String usuario = txtUsuario.getText();  
    char [] password = pwdPassword.getPassword();  
  
    String pwd = new String(password);  
  
    Usuario usuarioValidar = new Usuario();  
    usuarioValidar.setLogin(usuario);  
    usuarioValidar.setPassword(pwd);  
  
    if(validarUsuario(usuarioValidar)){  
        this.dispose();  
        FrmContenedorPrincipal principal = new FrmContenedorPrincipal();  
        principal.setVisible(true);  
    }else{  
        JOptionPane.showMessageDialog(this, "El Login o Password son  
incorrectos", "Validación Incorrecta", JOptionPane.ERROR_MESSAGE);  
    }  
}
```

Con la instrucción dispose() se liberan los recursos de la ventana actual, posteriormente se instancia un objeto del Frame que se quiere mostrar y se hace visible.