

Pasos para crear un formulario en Java Swing y no morir en el intento

Prerrequisitos

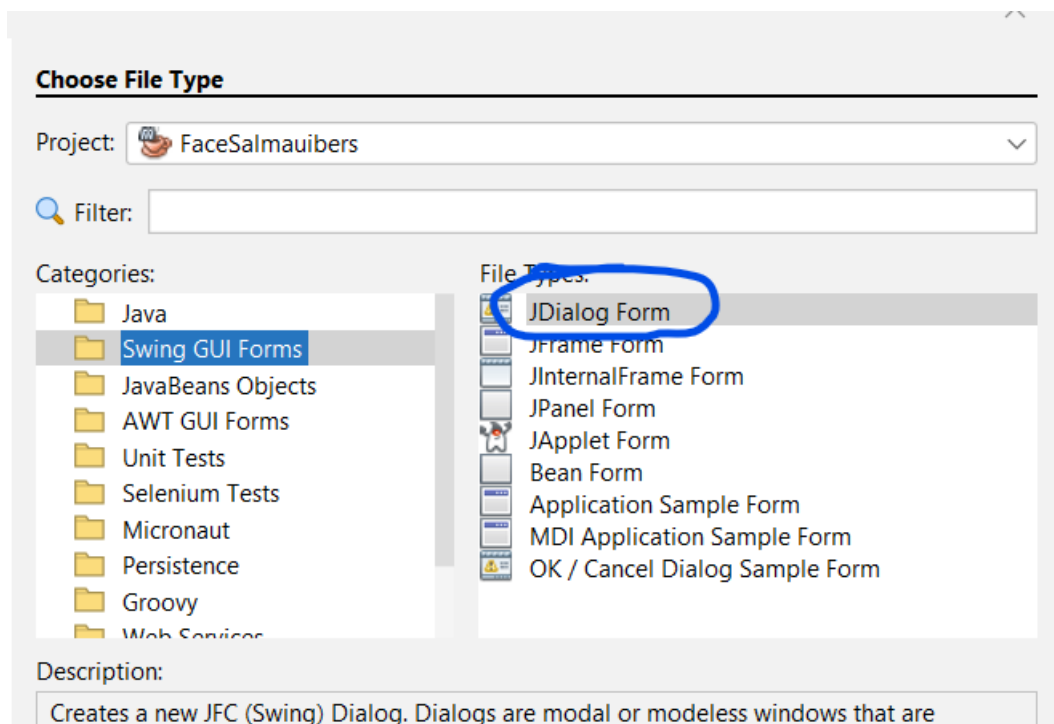
Paso 1) Tener un mínimo de IQ

Paso 2) Tener netbeans

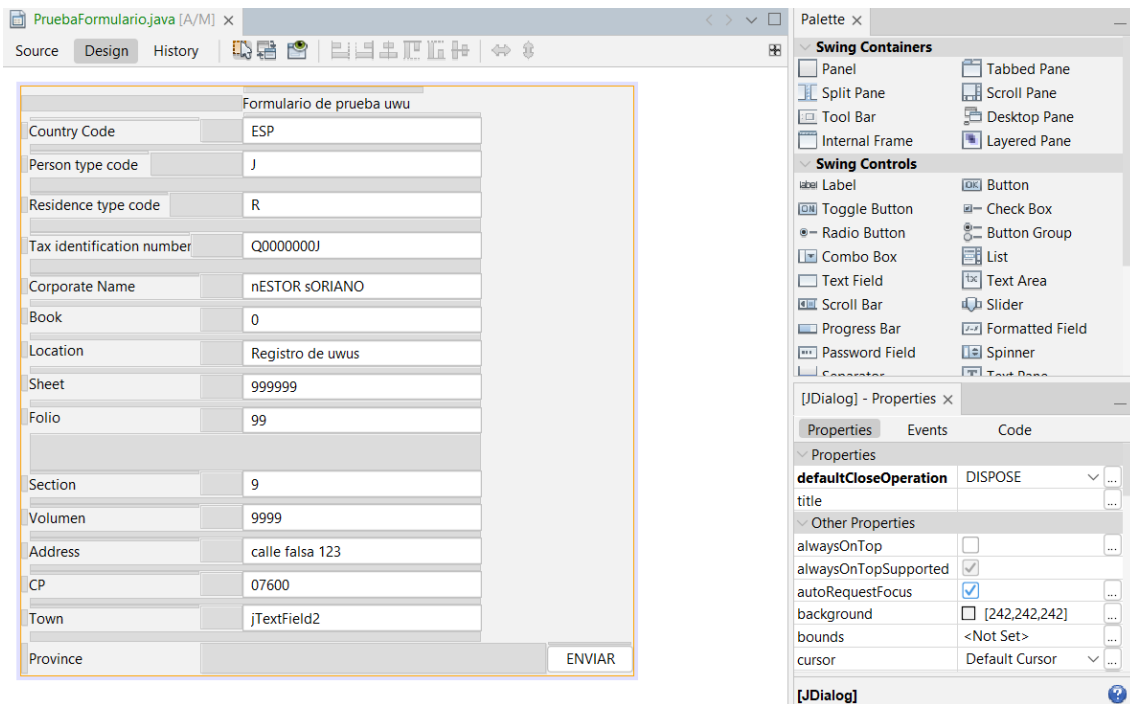
Paso 3) CONOCER LA ESTRUCTURA JERÁRQUICA DE LAS CLASES JAXB

1 Creando el formulario en netbeans

Para crear el formulario en netbeans, hay que elegir **JDialog** en lugar de JFrame porque es modal.



Una vez creado, hay que añadir los campos que queramos, funciona igual que Android Studio. Crea mucha paja en el archivo, pero NO HAY QUE ASUSTARSE, no hace falta entender todo lo que hay, lo que está en gris NO es modificable y lo añade netbeans.



2 Lógica del formulario

Para crear la lógica del formulario, se necesitan 4 (cuatro) elementos imprescindibles.

- El listener del botón que llamará a los métodos necesarios
- El constructor del formulario tiene que aceptar como parámetro un objeto e instanciarlo. `This.objeto = objeto`.
- El formulario tiene que tener un método para rellenar el formulario que recogerá los campos del formulario y hará `campo.set(campoFormulario)`
- La clase padre de lo que hayamos elegido rellenar tendrá un método `rellenar()` que llamará al formulario y lo hará visible.

Listener del botón

El listener del botón funciona igual que los de Android Studio.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    rellenarDatosFormulario();
    dispose();
}
```

Esto simplemente hace que cuando hagas click al botón, se abre el formulario, y luego lo cierra con `dispose()`;. Estaría bien aquí añadir que abra el siguiente formulario antes del `dispose`. El listener lo añade netbeans y nosotros lo rellenamos.

Constructor del formulario

Es IMPORTANTE que el formulario tenga un campo del padre de MAS ALTA JERARQUIA y que en el constructor lo inicializemos.

Ejemplo: SellerParty

El formulario tendrá como atributo un objeto SellerParty que se inicializará en el constructor. Esto funciona por que los objetos en Java se pasan por referencia, no lo se explicar, pero tiene que estar para funcionar.

```
public class PruebaFormulario extends javax.swing.JDialog {  
    private SellerParty seller;  
    /**  
     * Creates new form PruebaFormulario  
     */  
    public PruebaFormulario(java.awt.Frame parent, boolean modal, SellerParty sellerParty) {  
        this.seller = sellerParty;  
        initComponents();  
        this.setModal(true);  
    }  
}
```

Método que rellena el formulario

No tiene mucho, coge los campos del formulario, y los mete.

```
] private void rellenarDatosFormulario() {  
  
    TaxIdentification sellerTaxId = new TaxIdentification();  
    sellerTaxId.setPersonTypeCode(personType.getText());  
    sellerTaxId.setResidenceTypeCode("R");  
    sellerTaxId.setTaxIdentificationNumber(taxIdNumber.getText());  
    seller.setTaxIdentification(sellerTaxId);  
  
    LegalEntity sellerLegalEntity = new LegalEntity();  
    sellerLegalEntity.setCorporateName(corporateName.getText());  
  
    RegistrationData sellerRegistrationData = new RegistrationData();  
    sellerRegistrationData.setBook(Integer.valueOf(book.getText()));  
    sellerRegistrationData.setRegisterOfCompaniesLocation(location.getText());  
    sellerRegistrationData.setSheet(Integer.valueOf(sheet.getText()));  
    sellerRegistrationData.setFolio(Integer.valueOf(folio.getText()));  
    sellerRegistrationData.setSection(Integer.valueOf(seccion.getText()));  
    sellerRegistrationData.setVolume(Integer.valueOf(volumen.getText()));  
    sellerLegalEntity.setRegistrationData(sellerRegistrationData);  
  
    AddressInSpain sellerAddress = new AddressInSpain();  
    sellerAddress.setAddress("Calle falsa 123");  
    sellerAddress.setPostCode("07600");  
    sellerAddress.setTown("Palma");  
    sellerAddress.setProvince("Baleares");  
    sellerAddress.setCountryCode("ESP");  
    sellerLegalEntity.setAddressInSpain(sellerAddress);  
    seller.setLegalEntity(sellerLegalEntity);  
}
```

Método de la clase padre que llama al formulario

Es como el handler del formulario, lo tiene la clase SellerParty porque es la de mas alta jerarquía, y abre el formulario.

```
public void rellenar() {  
    java.awt.Frame parentFrame = null;  
    PruebaFormulario miFormulario = new PruebaFormulario(parentFrame, true, this); // Modal  
    miFormulario.setVisible(true);  
}
```

En la clase del MARSHAL se utiliza así:

```
// SellerParty  
  
SellerParty sellerDeFormulario = new SellerParty();  
sellerDeFormulario.rellenar();  
System.out.println(sellerDeFormulario.getLegalEntity().getCorporateName());  
parties.setSellerParty(sellerDeFormulario);
```