



Pasos a seguir para crear una aplicación visual en el lenguaje de programación Java con Netbeans IDE.



Autores: Ing. Maikel Zuñiga Suarez.

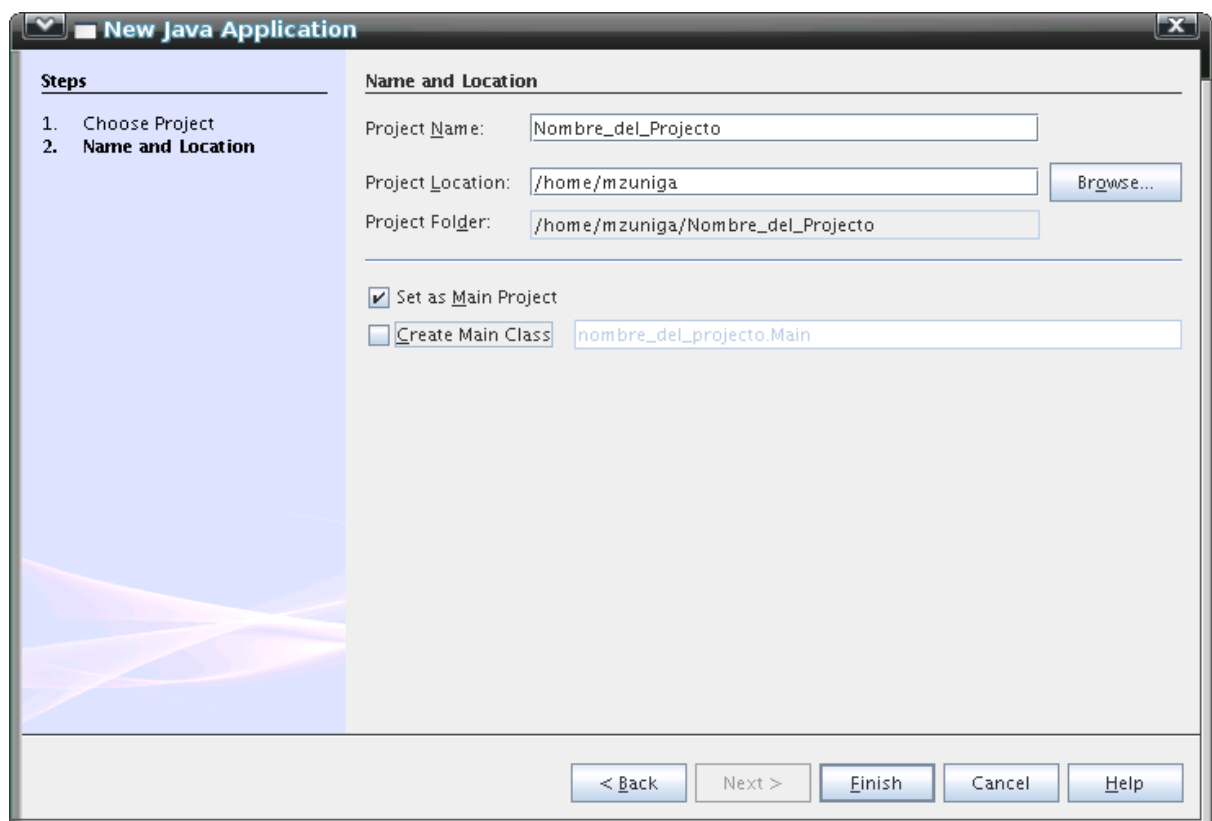
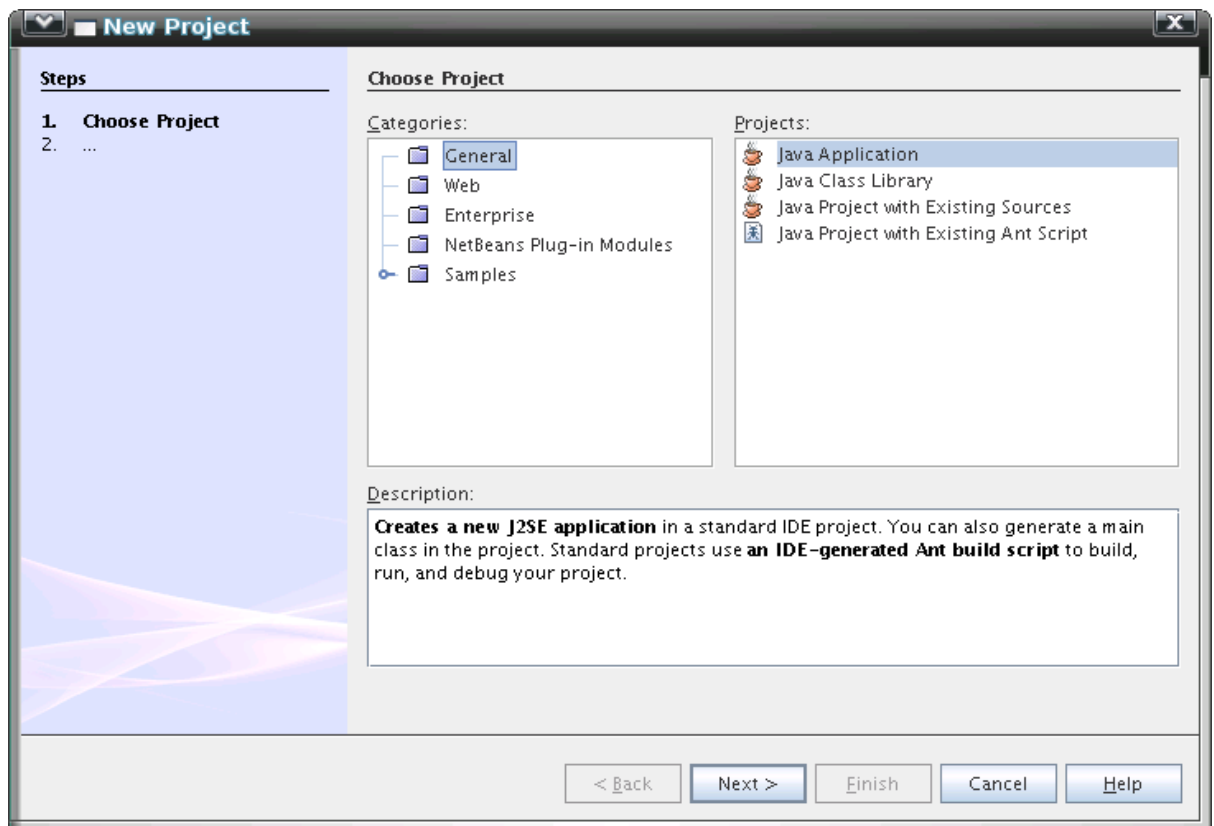
Reisel González Pérez.

Universidad de las Ciencias Informáticas

Ciudad de la Habana



- 1- Crear un nuevo proyecto y desmarcar la opción Create Main Class en la segunda ventana que aparece y finalizar.





2- Implementar todas las clases necesarias para el proyecto con todas sus funcionalidades.

```
package poficina;

public class OficinaMultas {
    private Multa[] multas;
    private int cantM;

    public OficinaMultas() {
        multas=new Multa[100];
        int cantM=0;
    }

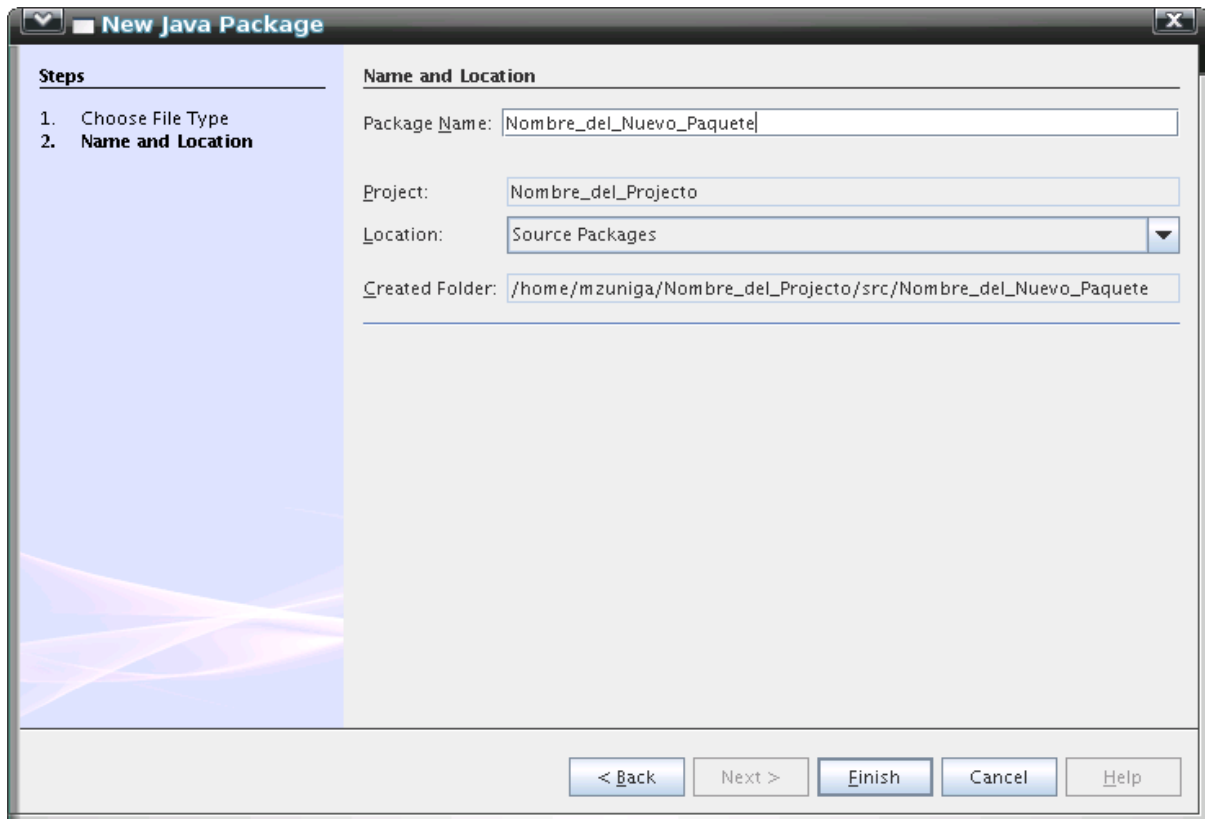
    public void AddInfraccion(String nombreInf, String cID, int puntosMulta, int idMultas) {
        if(cantM<100) {
            for(int i = 0; i < cantM; i++)
                if(multas[i].GetIdMultas()==idMultas)
                    throw new Exception("Ya existe una multa con ese id");
            multas[cantM++]=new Multa(nombreInf, cID, puntosMulta, idMultas);
        } else throw new Exception("No se puede agregar mas infracciones");
    }

    public boolean RetirarLicencia(String cID){
        int sumador=0;
        for(int i=0; i<cantM; i++)
            if(cID.equals(multas[i].GetCI()))
                sumador+=multas[i].GetPuntosMultas();
        if(sumador>=30)
            return true;
        return false;
    }

    public void CancelarMulta(int idMultas){
        Multa[] multas = new Multa[100];
    }
}
```

3- Por una cuestión de buenas prácticas de programación crearemos un nuevo paquete para mantener las formas visuales separadas del código de nuestras clases, por lo tanto el tercer paso es crear un nuevo paquete.

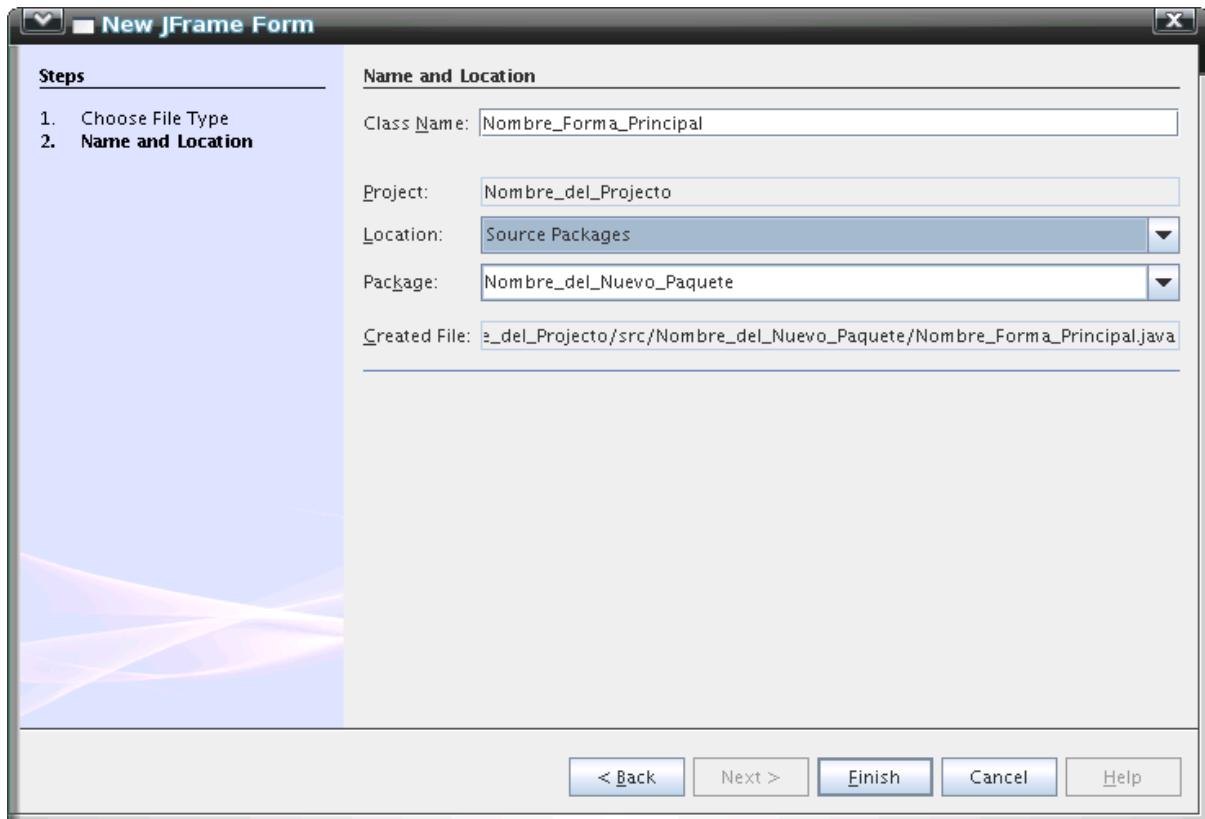
3.1- En las carpetas del proyecto damos click derecho en Source Packages, en new seleccionamos la opción Java Package, le damos un nombre y finalizamos.



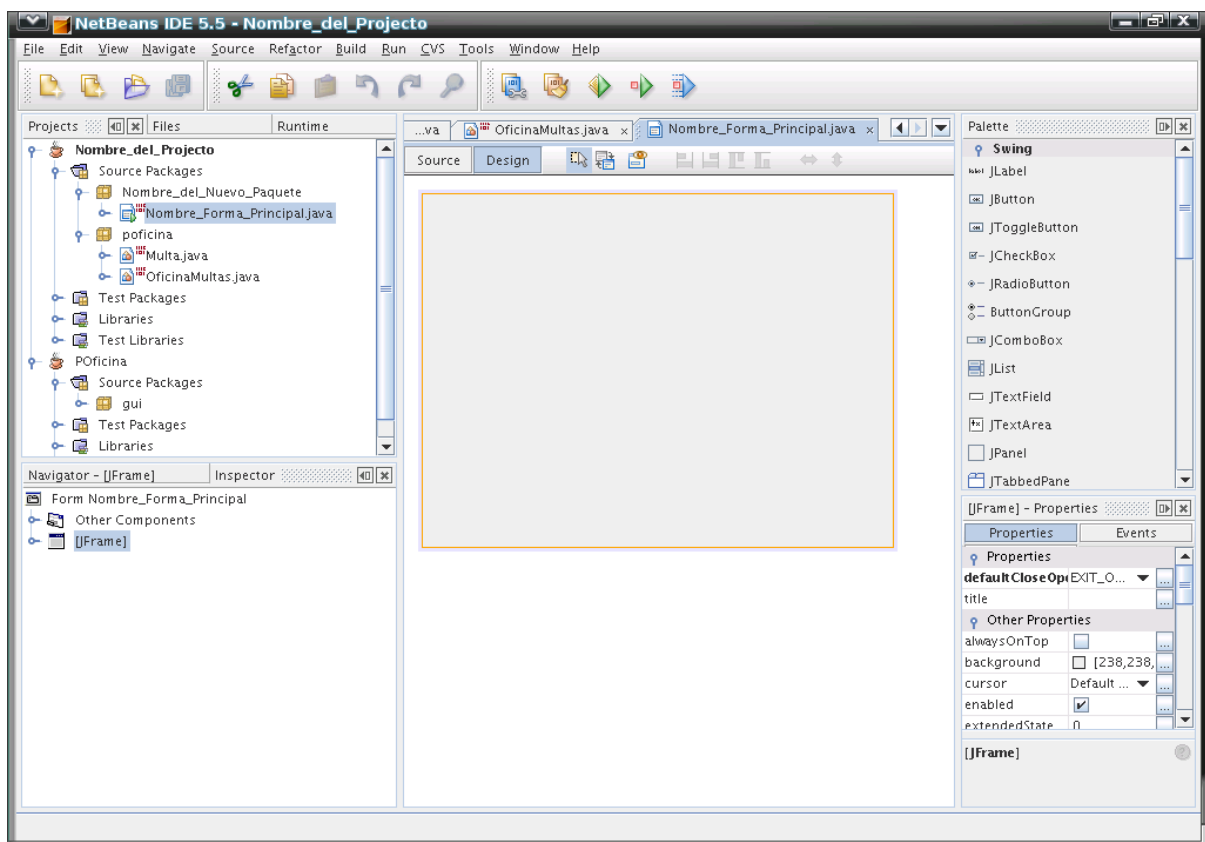
Nuestra aplicación visual debe tener una interfaz principal (JFrame) y puede tener o no varias interfaces auxiliares (JDialog) según convenga.

4- Crear la forma o interfaz principal

4.1- Para esto nos paramos sobre el nuevo paquete que creamos y damos click derecho, en new seleccionamos la opción JFrame Form, seleccionamos un nombre y damos finalizar.



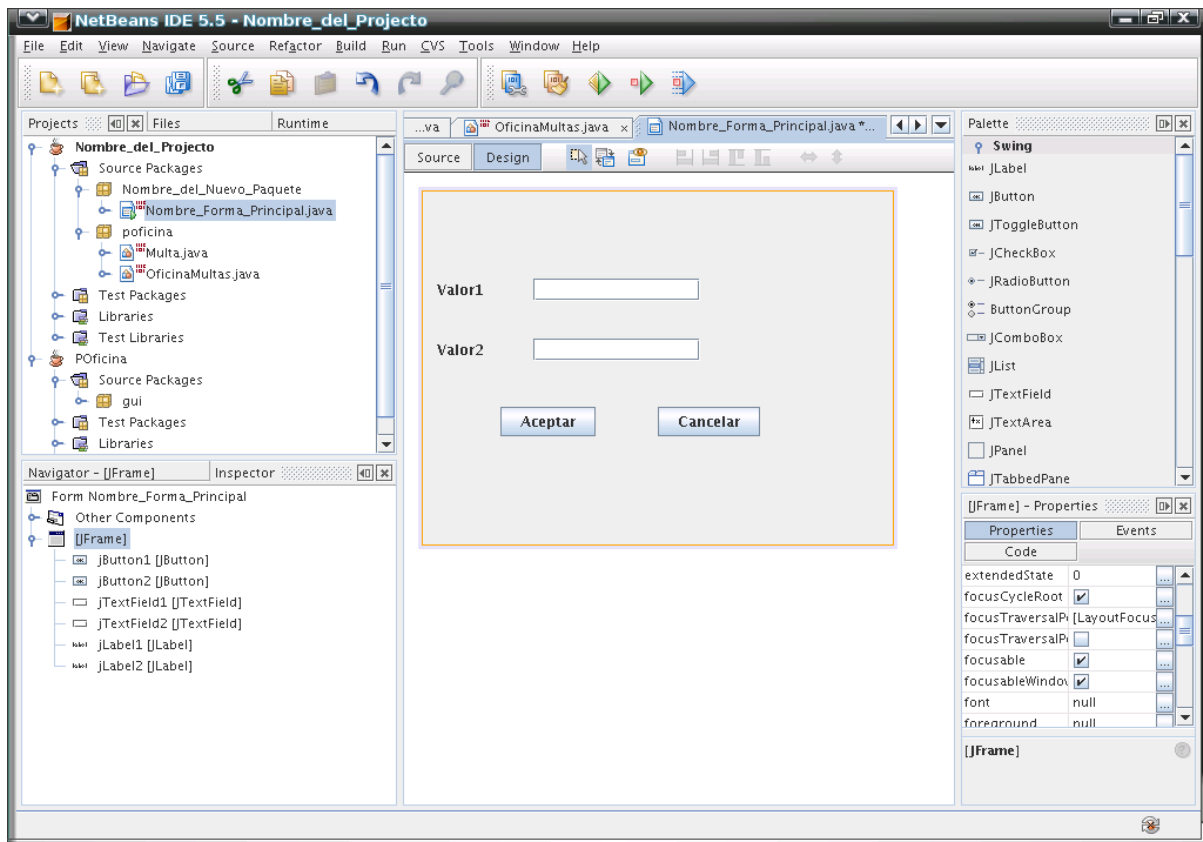
Una vez hecho esto aparecerá la forma visual en el Netbeans.



5- Añadir los componentes visuales deseados para nuestra aplicación.



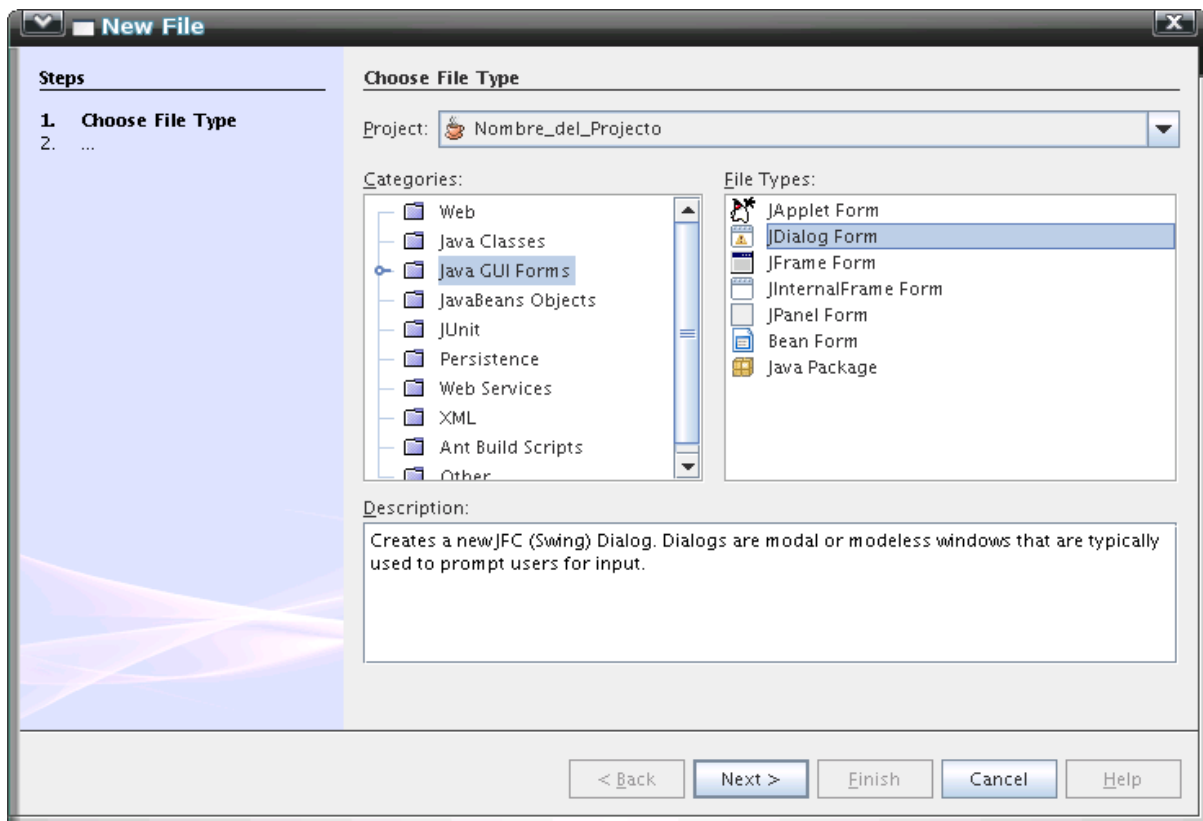
5.1- Para esto vamos a la paleta de componentes visuales y añadimos todos los que deseemos para nuestra aplicación.



6- Crear las formas auxiliares necesarias.

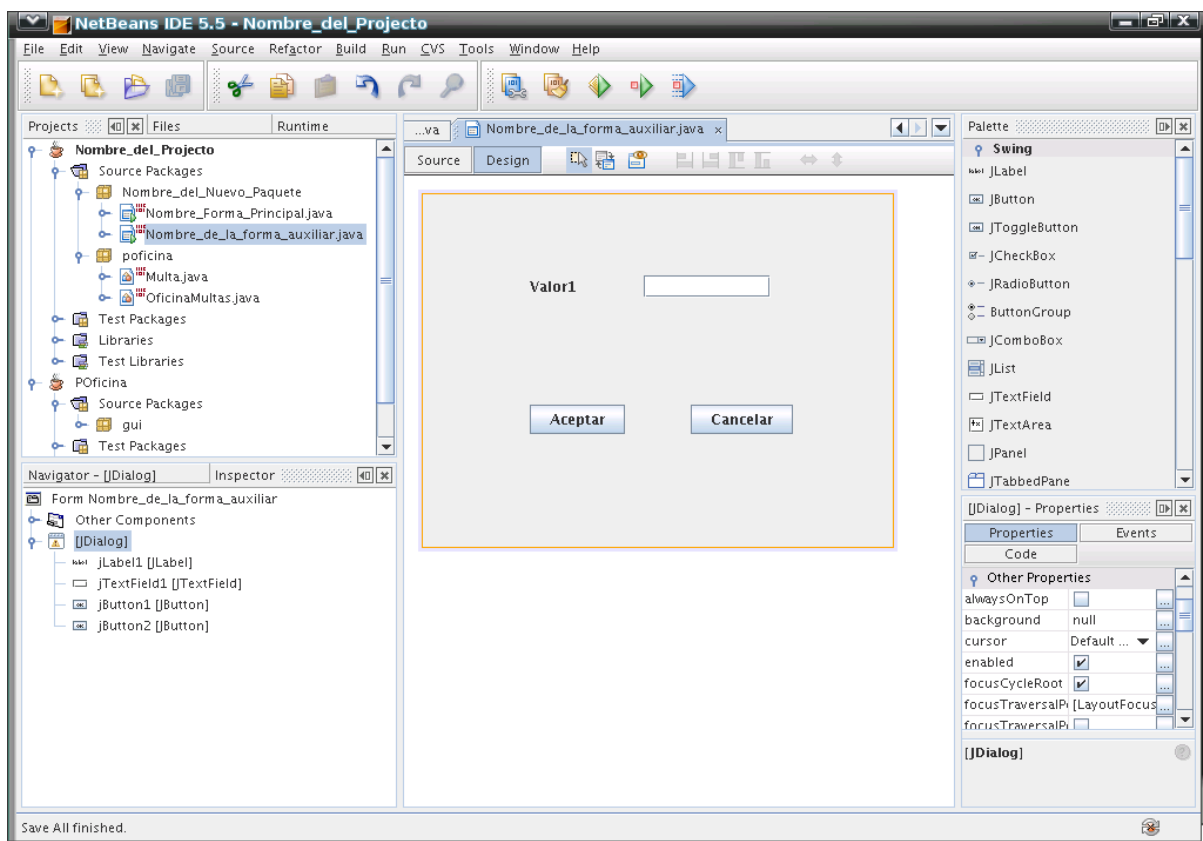
* Se necesita una forma auxiliar para cada método que requiera entrada de datos en forma de parámetros.

6.1- nos paramos sobre el nuevo paquete que creamos y damos click derecho, en new seleccionamos la opción File/Folder, en la nueva ventana que sale seleccionamos Java GUI Forms y dentro de esta JDialog Form.



6.2- Seleccionamos el nombre de la nueva forma auxiliar y damos click en Finish

7- Añadir los componentes visuales deseados para nuestra nueva forma. Este paso es similar al 5.1





Con los pasos anteriores tenemos nuestra parte visual lista para vincularla con nuestras clases. Para esto generalmente es necesario tener una instancia (objeto) de la clase principal en la forma principal, por ejemplo:

```
public class Nombre_Forma_Principal extends javax.swing.JFrame {

    /** Creates new form Nombre_Forma_Principal */

    private OficinaMultas oficina; // declaración de la variable de tipo de la clase principal en la forma principal

    public Nombre_Forma_Principal() {

        initComponents();

        oficina = new OficinaMultas(); // inicialización de la variable

    }

    /* Resto del Código de la forma*/

}
```

El objetivo de esto es poder trabajar sobre nuestra clase principal desde la interfaz visual, lo que facilita mucho el trabajo (en comparación con la consola) a nuestros usuarios finales. Además es necesario tener referencias a este objeto desde las formas auxiliares para trabajar con los diferentes métodos del mismo. Esto se logra en varios pasos:

8.1- Crear un objeto del tipo de la clase principal en la forma auxiliar deseada.

8.2- Sobrecargar el constructor de la forma auxiliar de forma tal que cuando creamos una instancia de la misma desde la forma principal podamos especificarle que el objeto creado anteriormente va a ser una referencia al objeto de la forma principal.

```
public class VAdicionarInfra extends javax.swing.JDialog {

    private OficinaMultas oficina; // creación del objeto
```




```
/** constructor por defecto de la forma auxiliar */  
public VAdicionarInfra(java.awt.Frame parent, boolean modal) {  
    super(parent, modal);  
    initComponents();  
}  
  
/** sobrecarga del constructor, este es el que hay que crear */  
public VAdicionarInfra(OficinaMultas of) {  
    this(null, true); // llamada al constructor por defecto  
    oficina = of;  
}  
/*Cuerpo de la forma*/  
}
```

Ahora solo nos falta darle funcionalidad a nuestra aplicación, o sea vincular la forma principal con las formas auxiliares para que el usuario pueda usar todas ventajas de nuestra aplicación. Esto se logra de manera muy sencilla realizando los pasos siguientes:

9- Seleccionar con el mouse el componente visual que queremos usar para acceder a la forma auxiliar. Este componente generalmente suele ser un JMenuItem o un JButton.

10- Damos click derecho sobre el componente seleccionado en el menú que se despliega seleccionamos:

Events->Mouse->MouseClicked

Esto nos va a llevar a la sección del código de la forma donde tenemos que implementar dicho evento (posteriormente se utilizaran otros eventos), o sea, el código que escribamos aquí es el que se va a ejecutar cuando demos click con el mouse sobre ese componente. Como lo que queremos es acceder a la forma auxiliar lo hacemos a través del siguiente código:

```
Nombre_de_la_forma_auxiliar forma_auxiliar = new Nombre_de_la_forma_auxiliar(oficina);  
//crea instancia de la forma auxiliar  
  
forma_auxiliar.setVisible(true); //muestra la instancia de la forma auxiliar
```