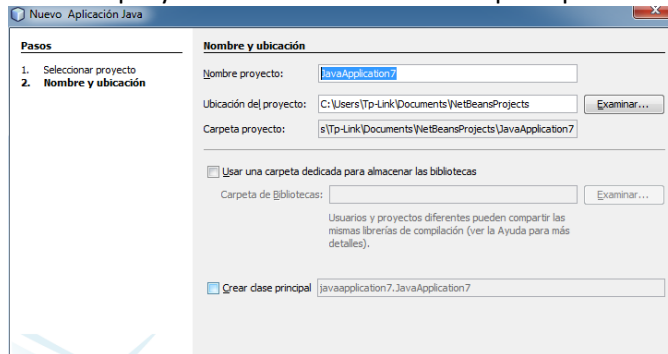


JAVA + POSTGRES

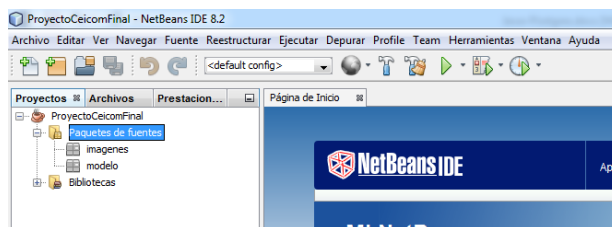
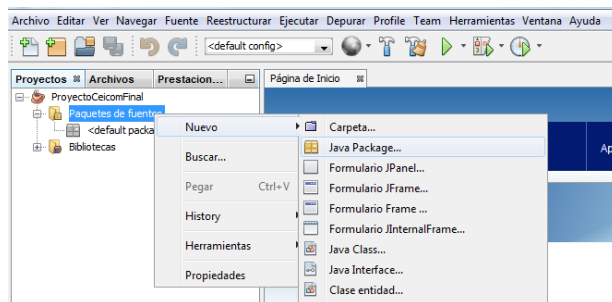
1. Paso 1: Creación del proyecto, paquetes y ventana principal

- Crear un proyecto en Netbeans sin clase principal.

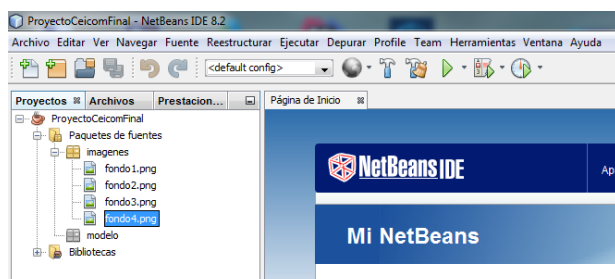
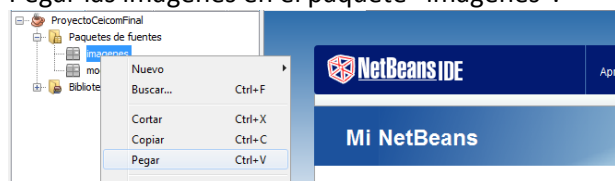


Quitar la opción “Crear clase principal”.

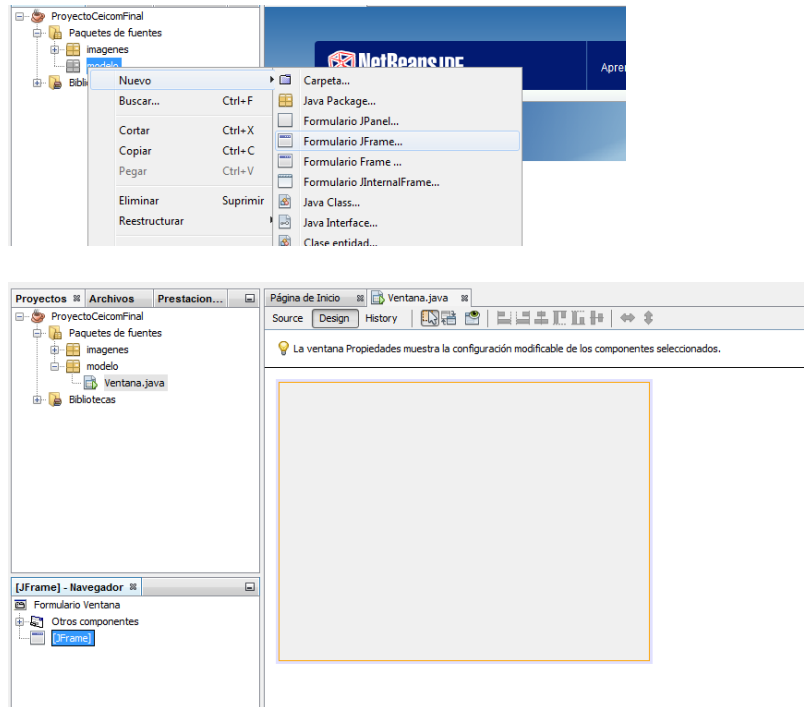
- Crear los paquetes “imagenes y modelo”



- Pegar las imágenes en el paquete “imagenes”.

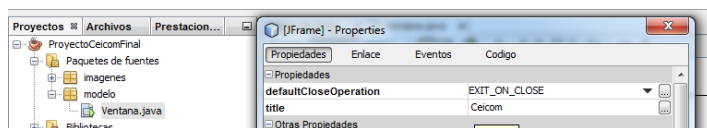
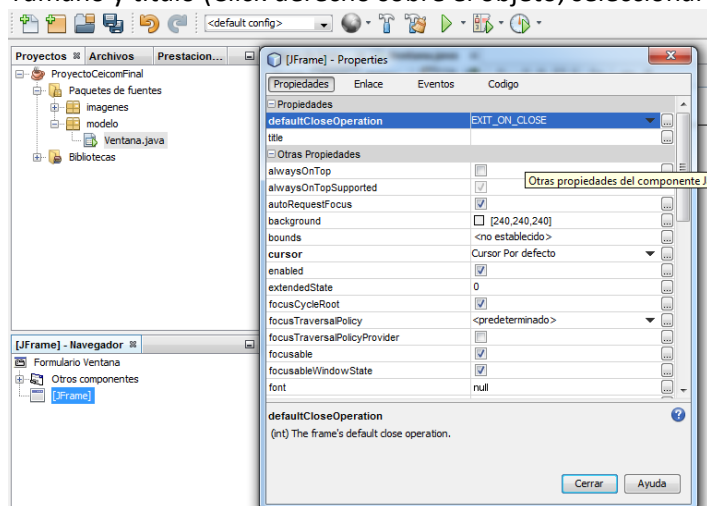


- Crear un formulario JFrame con el nombre "Ventana"

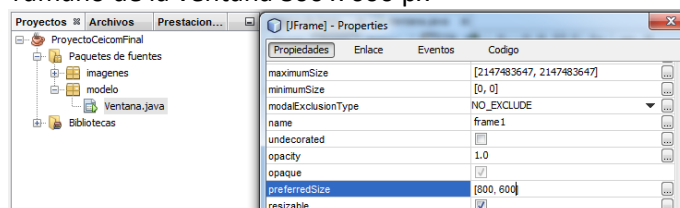


2. Paso 2: Configuración de la ventana

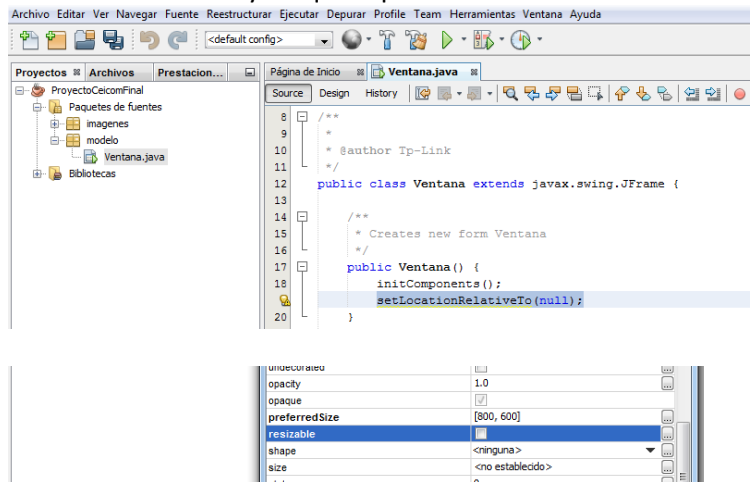
- Tamaño y título (Click derecho sobre el objeto, seleccionar propiedades).



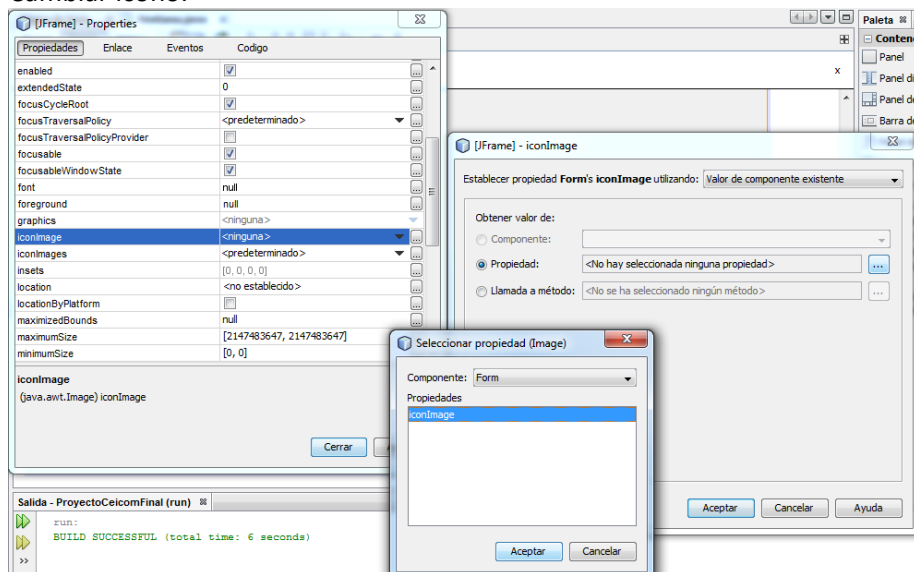
Tamaño de la ventana 800 x 600 px



- Ubicación centrado y bloquear para extender la ventana.

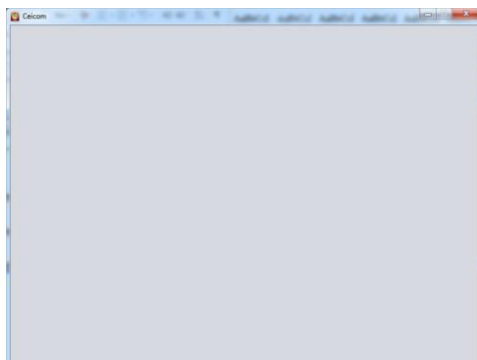


- Cambiar icono.

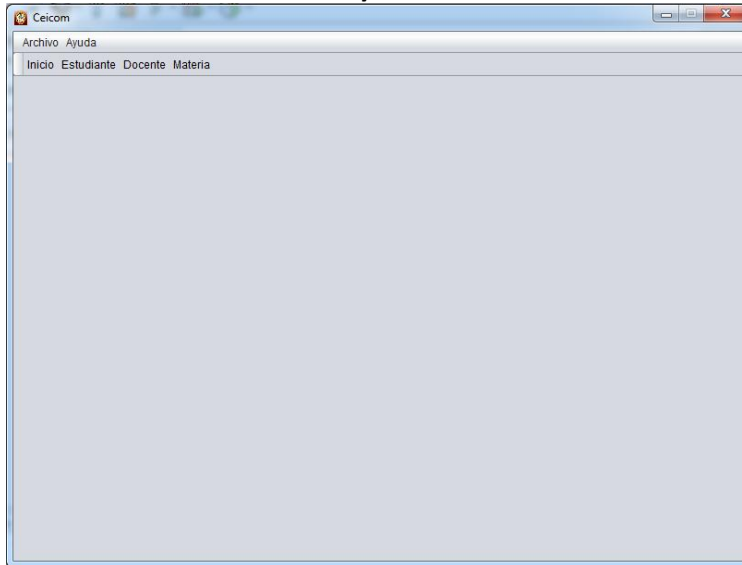


Crear un método con el nombre `getIconImage` que te retorne una Imagen

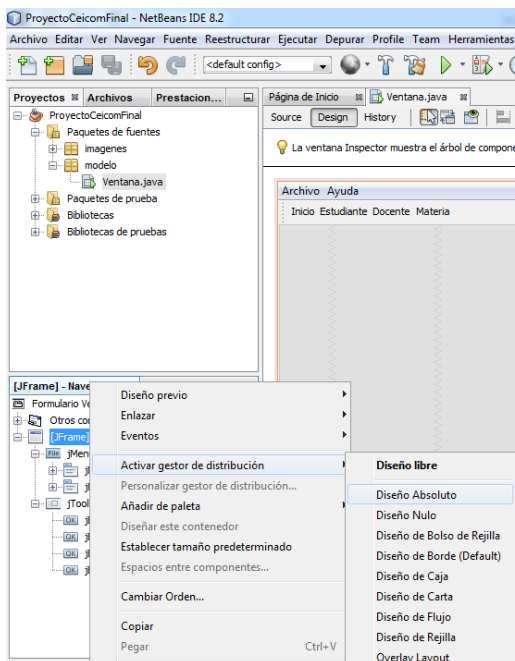
```
public Image getIconImage()
{
    return Toolkit.getDefaultToolkit().getImage(ClassLoader.getResource("imagenes/ceicom.jpg"));
}
```



3. Paso 3: Crear barra de menú y barra de herramientas.

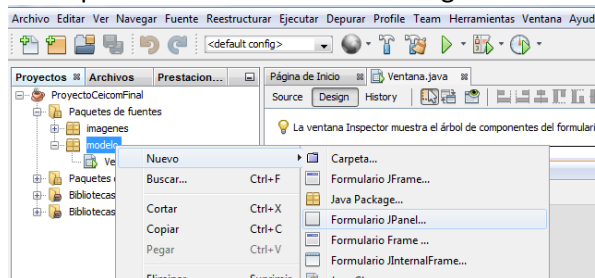


Cambiar la opción del layout a diseño absoluto para que nos permita colocar componentes uno sobre otro.



4. Paso 4: Panel para registrar estudiantes.

- Crear panel con el nombre "PanelRegEstudiante"



- Tamaño del panel 800 x 549
- Cambiar la opción del layout a diseño absoluto para que nos permita colocar componentes uno sobre otro.
- Para un fondo usamos un JLabel.

5. Conexión a la base de datos.

```

Connection connection=null;
public void conectarDB()
{
    if(connection!=null)
    {
        return;
    }
    String url="jdbc:postgresql://localhost:5432/problema1";
    try
    {
        Class.forName("org.postgresql.Driver");
        connection=DriverManager.getConnection(url,"postgres","baltazar");
        if(connection!=null)
        {
            System.out.println("Conectado con exito");
        }
    }
    catch(Exception e)
    {
        System.out.println("Problemas de conexión");
    }
}

```

6. Insertar un valor a la base de datos.

```

Connection connection=null;
Statement s=null;

connection=null;
conectarDB();
String ci = jTextFieldCi.getText();
int ciNum = Integer.parseInt(ci);
String nombre = jTextFieldNombre.getText();
String apellido = jTextFieldApellido.getText();
String fecha = jTextFieldFechaNac.getText();

try
{
    s=connection.createStatement();
    String sql="insert into estudiante(codigo_est,ci_est,nombre_est,apellido_est,fechanac_est)
values(nextval('seq_codigo_est'),'"+ciNum+"','"+nombre+"','"+apellido+"','"+fecha+"')";
    int z=s.executeUpdate(sql);
    if(z==1)
    {
        System.out.println("se ingresó con éxito");
    }
}
else

```

```

        {
            System.err.println("Error al ingresar los datos");
        }
        connection.close();
    }
    catch(Exception e)
    {
        System.out.println("Error al insertar");
    }
}
// nextval('sec_numerosocios') Coloca una secuencia en un atributo especificado.
//Colocar el evento para activar el botón con la tecla enter.
InputMap map = new InputMap();
map.put(KeyStroke.getKeyStroke(KeyEvent.VK_ENTER, 0, false), "pressed");
map.put(KeyStroke.getKeyStroke(KeyEvent.VK_ENTER, 0, true), "released");
jButton1.setInputMap(0, map);

```

7. Mostrar valores de una tabla

```

Connection connection=null;
ResultSet resultado=null;

connection = null;
conectarDB();
try
{
    resultado = connection.createStatement().executeQuery("SELECT
codigo_est,ci_est,nombre_est,apellido_est,fechanac_est FROM estudiante");
    DefaultTableModel modelo = new DefaultTableModel(
        new Object [][] {},
        new String [] {
            "Codigo", "CI", "Nombre", "Apellido", "Fecha Nac."
        }
    );

    jTable1.setModel(modelo);
    Object [] columnas=new Object[5];
    while(resultado.next())
    {
        for(int i = 0; i<5;i++)
        {
            columnas[i]=resultado.getObject(i+1);
        }
        modelo.addRow(columnas);
    }
    connection.close();
}
catch(Exception e)
{
    System.out.println("Error al cargar los datos");
}

```