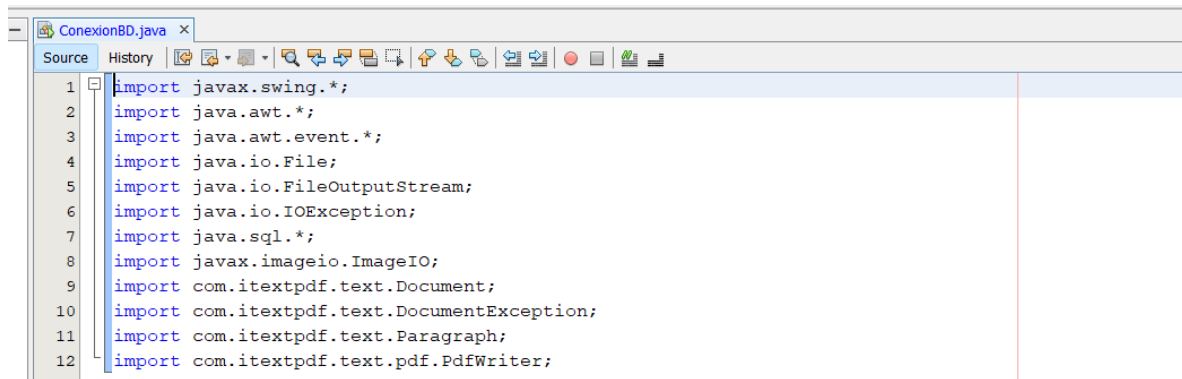


Iniciamos el código fuente con nuestra declaración de librerías

Más adelante se detalla el uso de cada una



```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.*;
4 import java.io.File;
5 import java.io.FileOutputStream;
6 import java.io.IOException;
7 import java.sql.*;
8 import javax.imageio.ImageIO;
9 import com.itextpdf.text.Document;
10 import com.itextpdf.text.DocumentException;
11 import com.itextpdf.text.Paragraph;
12 import com.itextpdf.text.pdf.PdfWriter;
```

Declaramos nuestras pantallas a utilizar

```
public class ConexionBD {
    private JFrame frame;
    private JPanel bienvenidaPanel, principalPanel, formularioPanel;
    private Connection conexion;
```

Y nuestra extension a la base de datos

```
// Constructor
public ConexionBD() {
    conectarBaseDeDatos();
    prepararVentana();
    mostrarPantallaLogin();
}

private void conectarBaseDeDatos() {
    String url = "jdbc:mysql://localhost:3306/bdiximche";
    String usuario = "root";
    String contraseña = "Niltin!17";

    try {
        conexion = DriverManager.getConnection(url, usuario, contraseña);
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Error al conectar a la base de datos: " + e.getMessage());
        e.printStackTrace();
    }
}
```

Constructor: Llama a métodos para conectar a la base de datos, configurar la ventana y mostrar la pantalla de inicio de sesión.

Conectar a base de datos: Define la URL, usuario y contraseña de la base de datos, e intente conectarse. Si falla, muestra un mensaje de error y la traza de la excepción.

Muestra un panel de inicio de sesión con campos para el usuario y la contraseña. Si el usuario ingresa sus credenciales y presiona "OK", se verifica la autenticación. Si es correcta, se muestra la ventana principal; si es incorrecto, aparece un mensaje de error y se repite el inicio de sesión. Si se cancela el inicio de sesión, el programa se cierra.

```
private void mostrarPantallaLogin() {
    JTextField usuarioField = new JTextField(15);
    JPasswordField contraseñaField = new JPasswordField(15);

    JPanel loginPanel = new JPanel(new GridLayout(3, 2, 10, 10));
    loginPanel.add(new JLabel("Usuario:"));
    loginPanel.add(usuarioField);
    loginPanel.add(new JLabel("Contraseña:"));
    loginPanel.add(contraseñaField);

    int result = JOptionPane.showConfirmDialog(null, loginPanel, "Inicio de Sesión", JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE);

    if (result == JOptionPane.OK_OPTION) {
        String usuario = usuarioField.getText();
        String contraseña = new String(contraseñaField.getPassword());

        if (autenticarUsuario(usuario, contraseña)) {
            frame.setVisible(true);
            mostrarBienvenida();
        } else {
            JOptionPane.showMessageDialog(null, "Usuario o contraseña incorrectos", "Error de Autenticación", JOptionPane.ERROR_MESSAGE);
            mostrarPantallaLogin();
        }
    } else {
        System.exit(0);
    }
}
```

Este método, autenticar Usuario verifica si las credenciales ingresadas coinciden con las almacenadas internamente. Defina un usuario correcto (Nilton) y una contraseña correcta (1234). Devuelve true si ambos valores ingresados coinciden con los correctos; de lo contrario, devuelve false.

```
private boolean autenticarUsuario(String usuario, String contraseña) {
    String usuarioCorrecto = "Nilton";//cambio de nombre de usuario
    String contraseñaCorrecta = "1234";//contraseña
    return usuario.equals(usuarioCorrecto) && contraseña.equals(contraseñaCorrecta);
}
```

Este método crea la pantalla principal de administración de productos, donde cada botón permite realizar una acción específica: agregar, modificar, eliminar, listar, buscar productos, o generar un PDF con la lista de productos.

```
private void mostrarPantallaProductos() {
    frame.getContentPane().removeAll();

    principalPanel = new JPanel(new GridLayout(6, 1, 10, 10));
    principalPanel.setBackground(Color.lightGray);

    JButton insertarButton = new JButton("Insertar Producto");
    insertarButton.addActionListener(e -> mostrarFormulario("insertar"));

    JButton modificarButton = new JButton("Modificar Producto");
    modificarButton.addActionListener(e -> mostrarFormulario("modificar"));

    JButton eliminarButton = new JButton("Eliminar Producto");
    eliminarButton.addActionListener(e -> mostrarFormulario("eliminar"));

    JButton listarButton = new JButton("Listar Productos");
    listarButton.addActionListener(e -> listarProductos());

    JButton buscarButton = new JButton("Buscar Producto");
    buscarButton.addActionListener(e -> mostrarFormulario("buscar"));

    JButton pdfButton = new JButton("Crear PDF de Productos");
    pdfButton.addActionListener(e -> crearPDF());

    principalPanel.add(insertarButton);
    principalPanel.add(modificarButton);
    principalPanel.add(eliminarButton);
    principalPanel.add(listarButton);
    principalPanel.add(buscarButton);
    principalPanel.add(pdfButton);

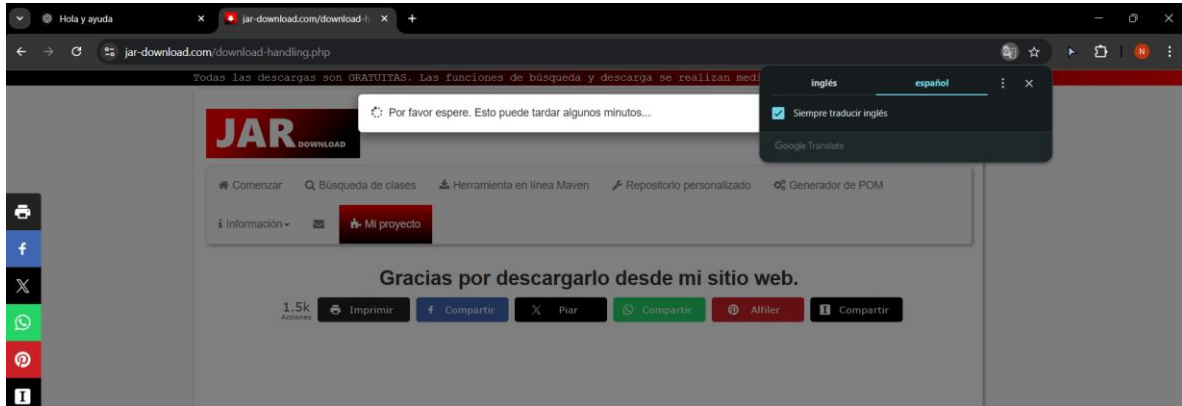
    frame.add(principalPanel, BorderLayout.CENTER);
    frame.revalidate();
    frame.repaint();
}
```

El método crearPDF() genera un archivo "ListaProductos.pdf" que contiene una lista detallada de productos obtenidos de la base de datos. Realiza los siguientes pasos:

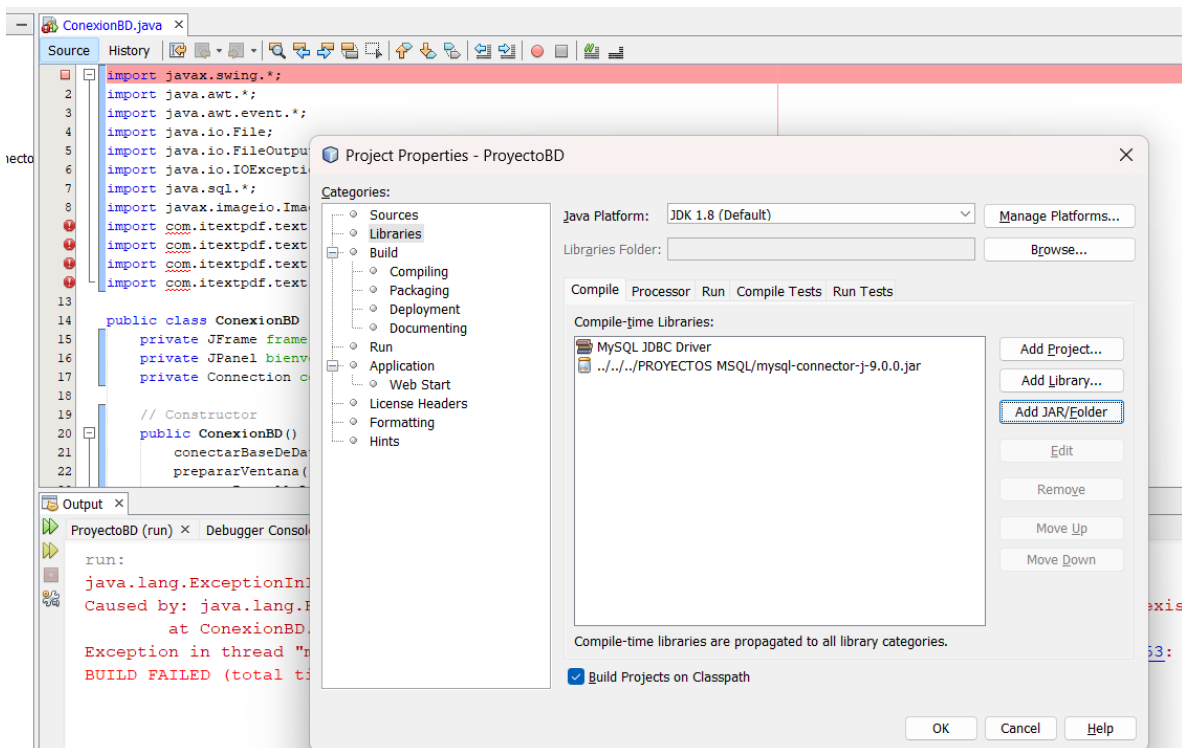
1. Crea y abre un documento PDF .
2. Agrega un título ("Lista de Productos").
3. Consulta la base de datos para obtener todos los productos.
4. Escribe en el PDF la información de cada producto (código, nombre, precio, cantidad y fecha de
5. Cierra el documento al final

Apartado de instalación de itextpdf-5.5.13.2.jar

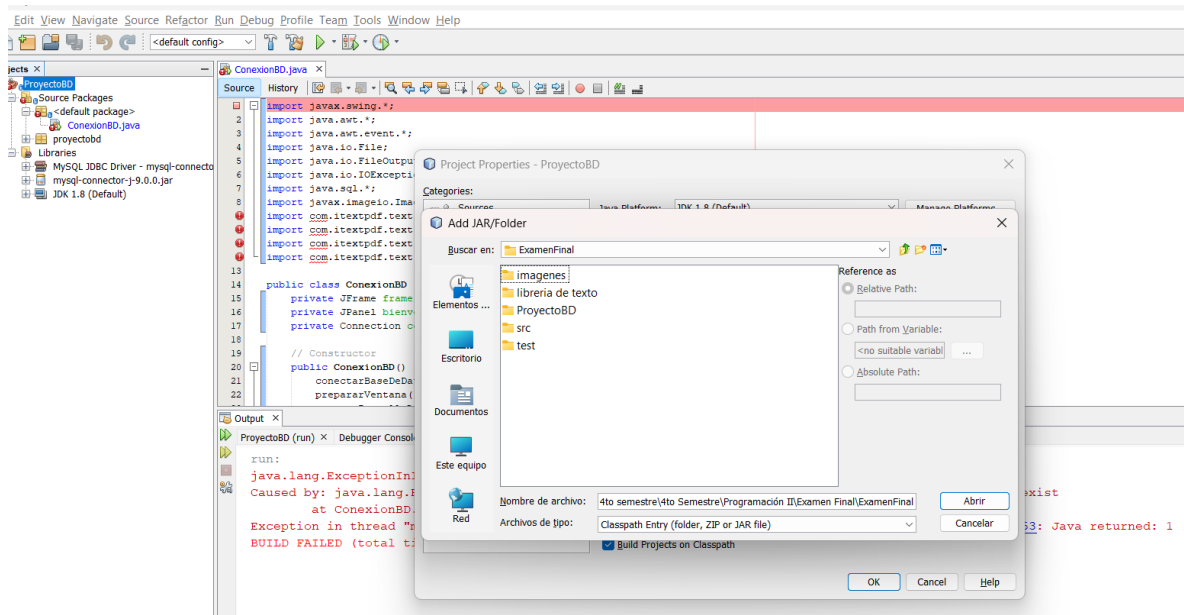
Para generar un PDF para el reporte



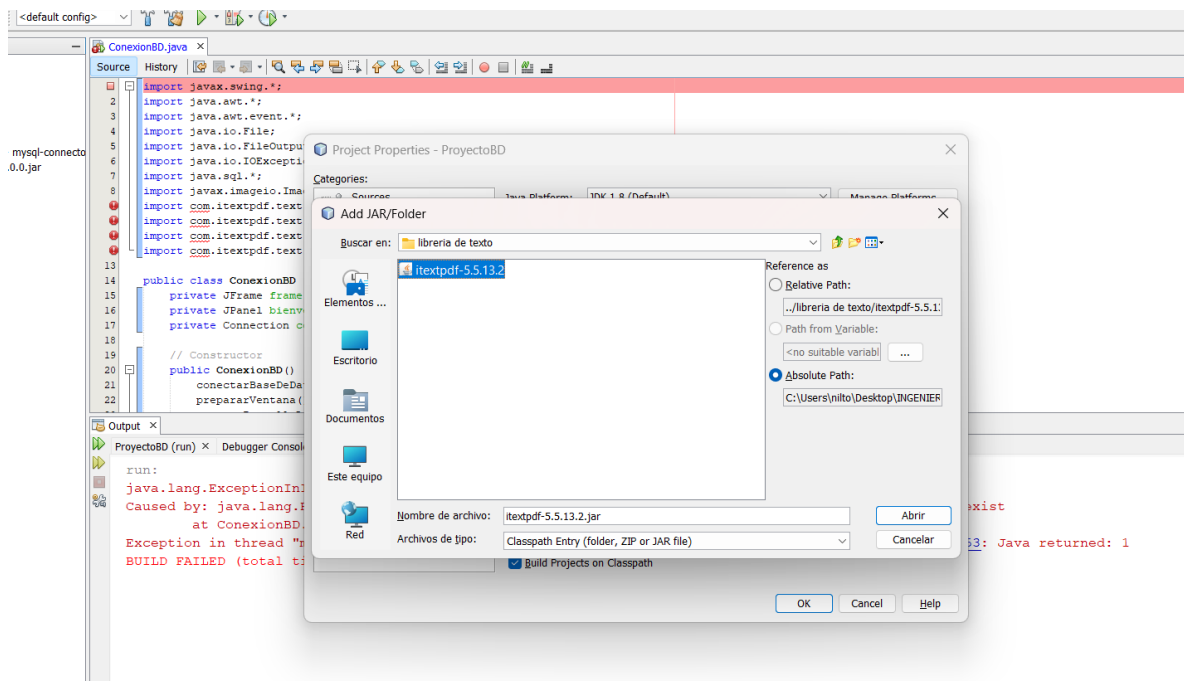
Luego de descargarlo vamos a incluirlo en el proyecto



seleccionamos add JAR/folder



Y lo buscamos en la carpeta donde guardamos la librería



seleccionamos la librería

Prueba de creación

Acá usaremos como nombre “prueba de Pdf” para asegurarnos que nos cree el documento

IXIMCHE

Insertar Producto

Modificar Producto

Eliminar Producto

Listar Productos

Buscar Producto

Crear PDF de Productos

Código: 01

Nombre: prueba de pdf

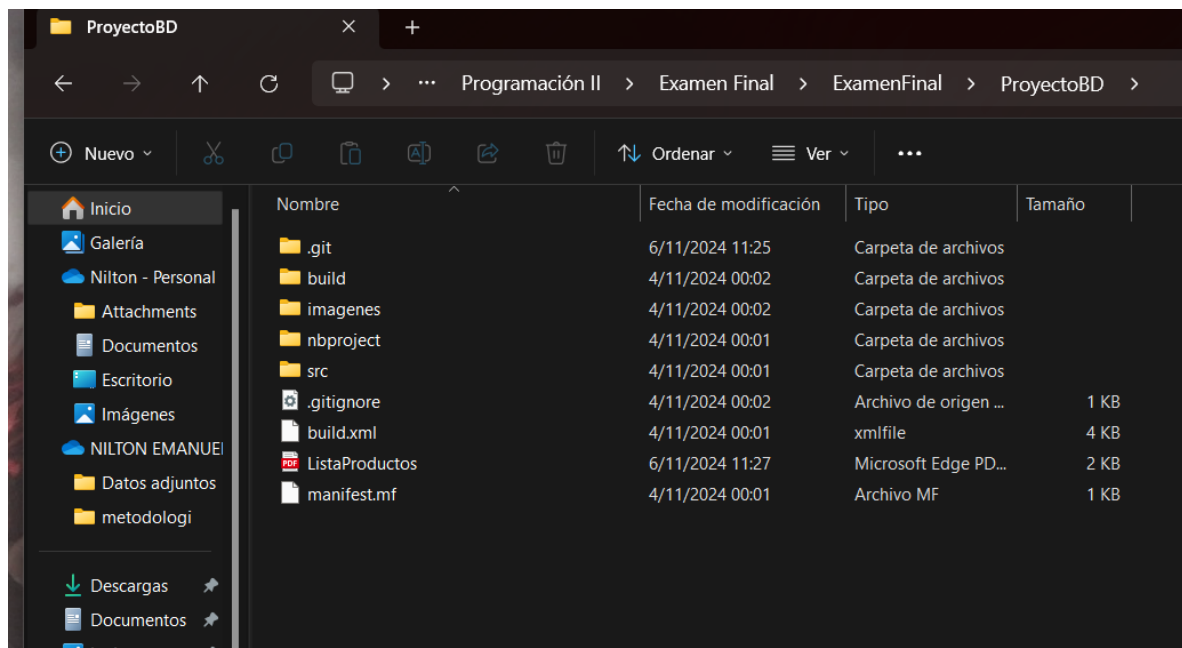
Cantidad: 2

Precio: 30

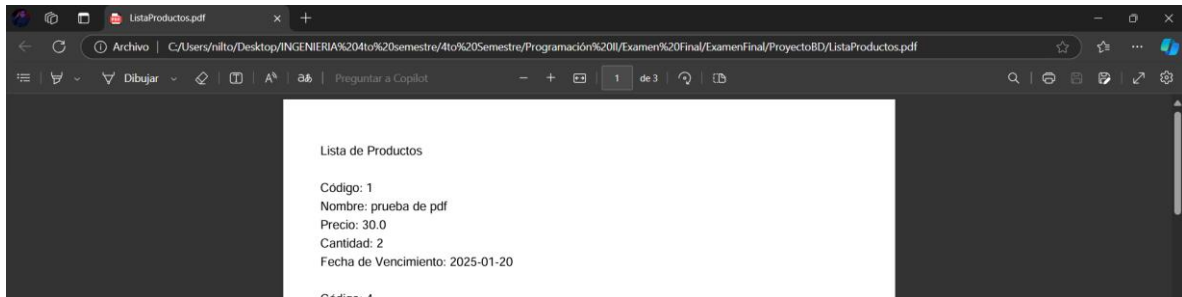
Fecha de Vencimiento (YYYY-MM-DD): 2025-01-20

Ejecutar

Luego debemos dirigirnos a la carpeta donde almacenamos el proyecto y nos aparecerá un archivo PDF llamado **Listadeproductos**

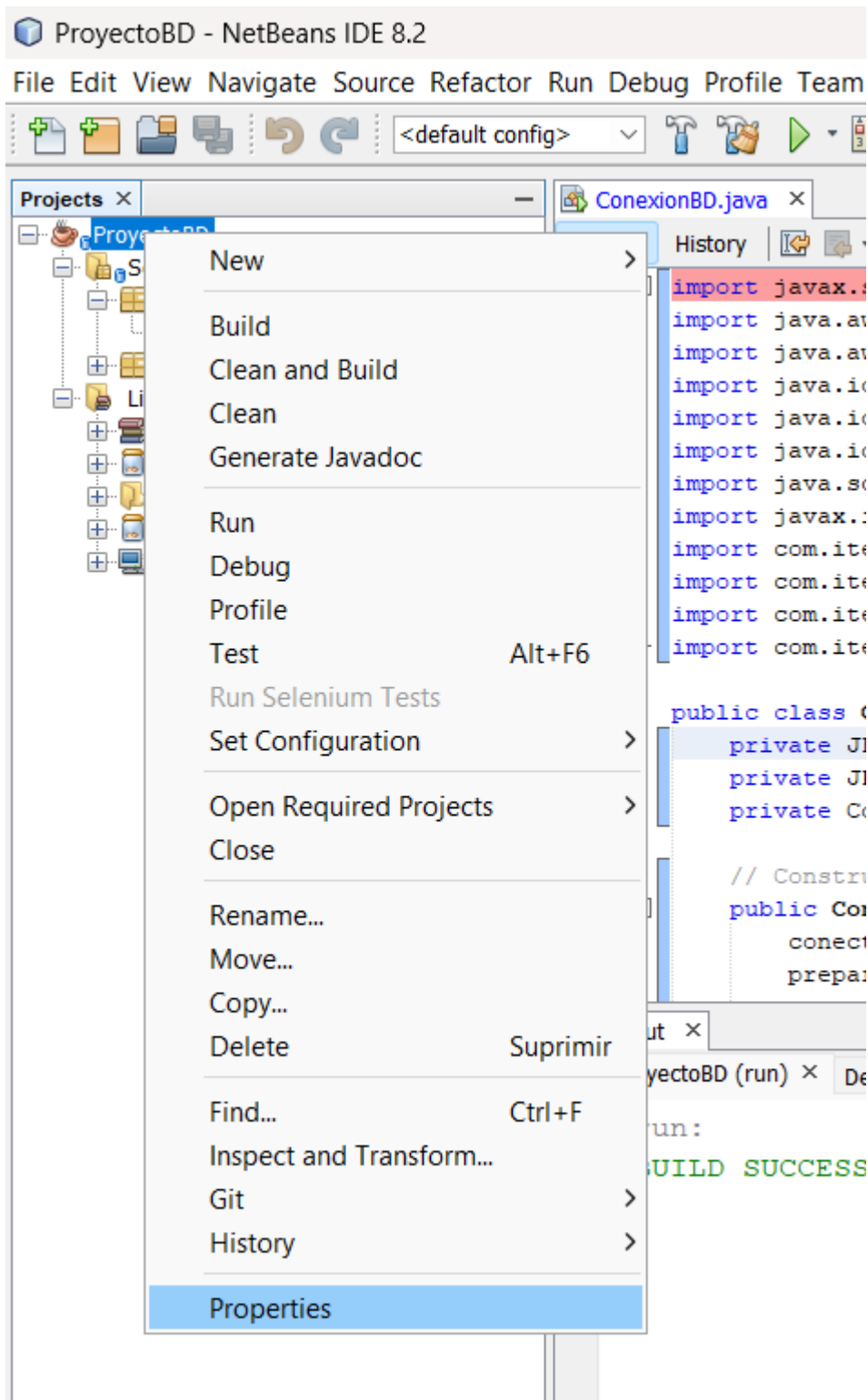


Este archivo se creara y actualizará con cada modificación al programa, es decir que en cada iteración creara un archivo nuevo y eliminara el anterior

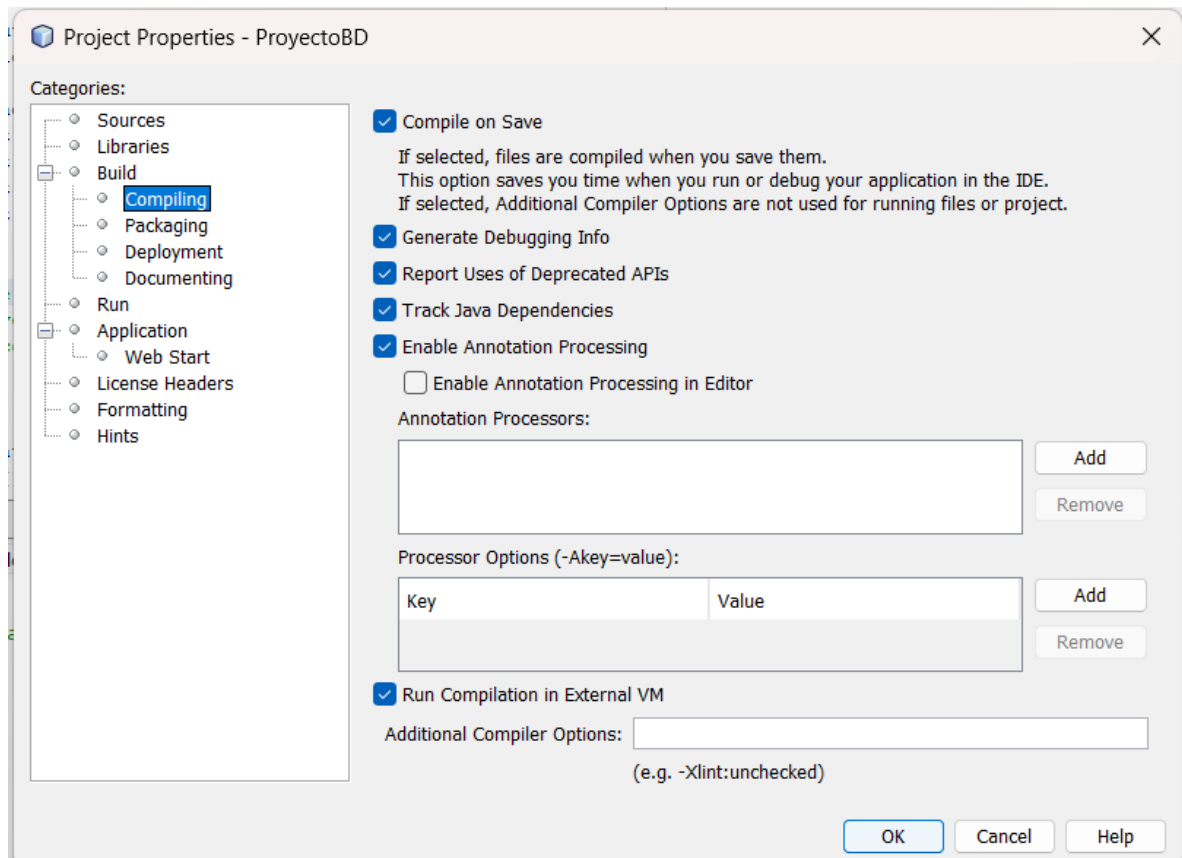


Ahora crearemos un archivo .JAR y luego un .exe para poder utilizarlo como aplicación

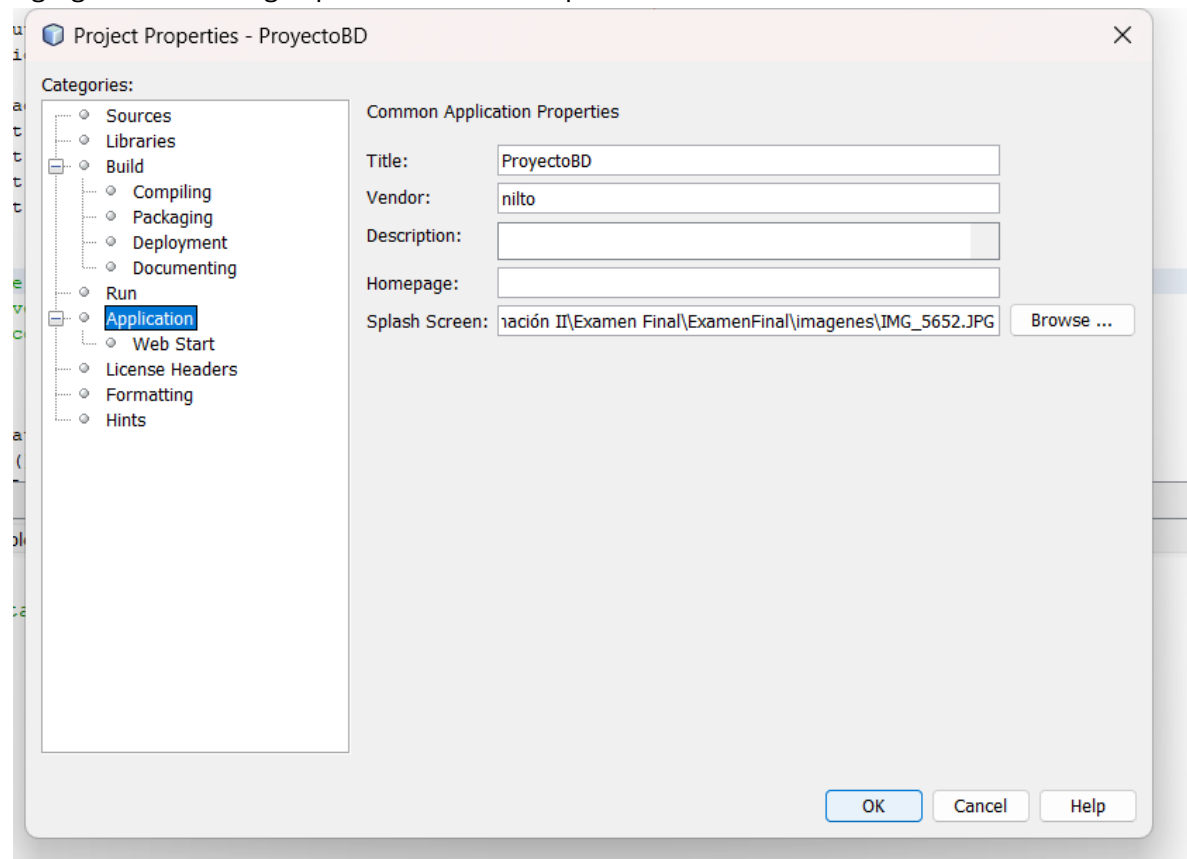
Seleccionamos propiedades



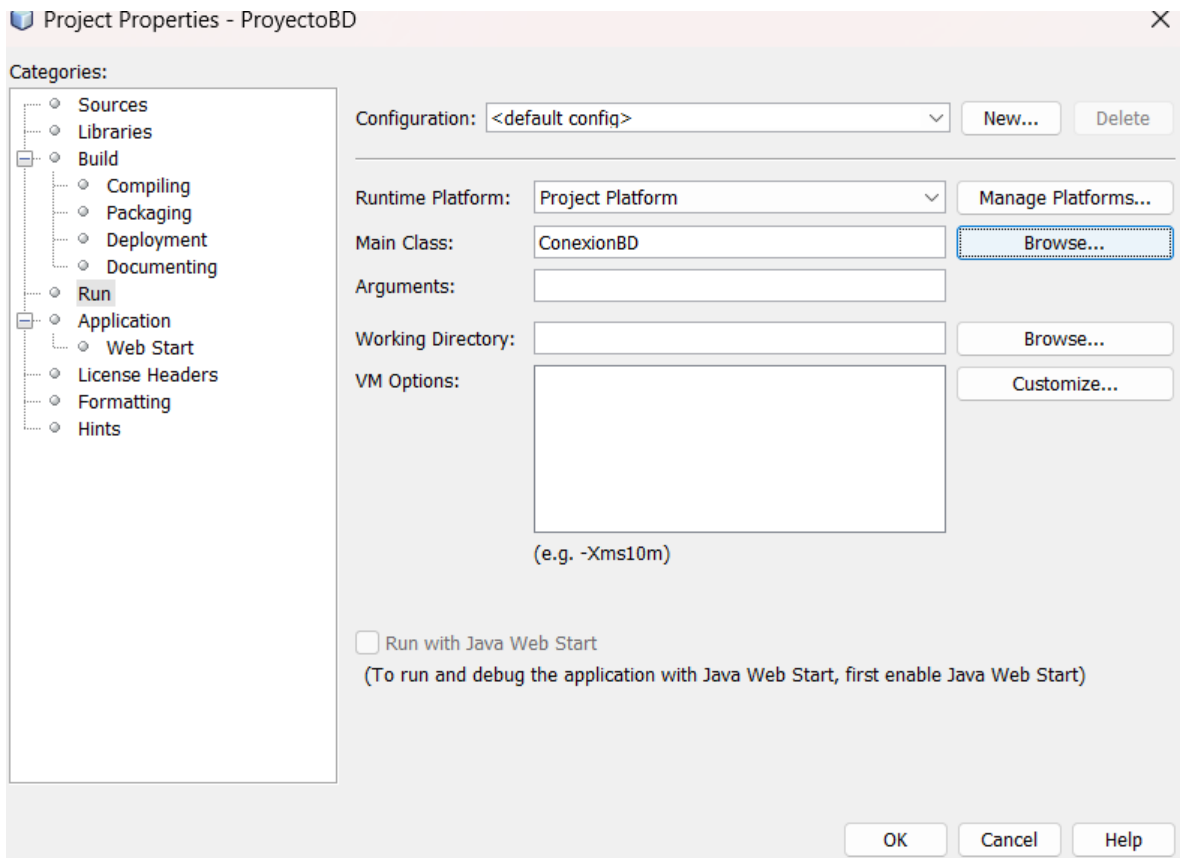
Dejamos todo seleccionado por defecto



Agregamos una imagen para el inicio de la aplicación



Y seleccionamos **OK** para finalizar



Seleccionamos la clase principal

Luego nos creara una nueva carpeta que dentro de ella tendrá el archivo .JAR

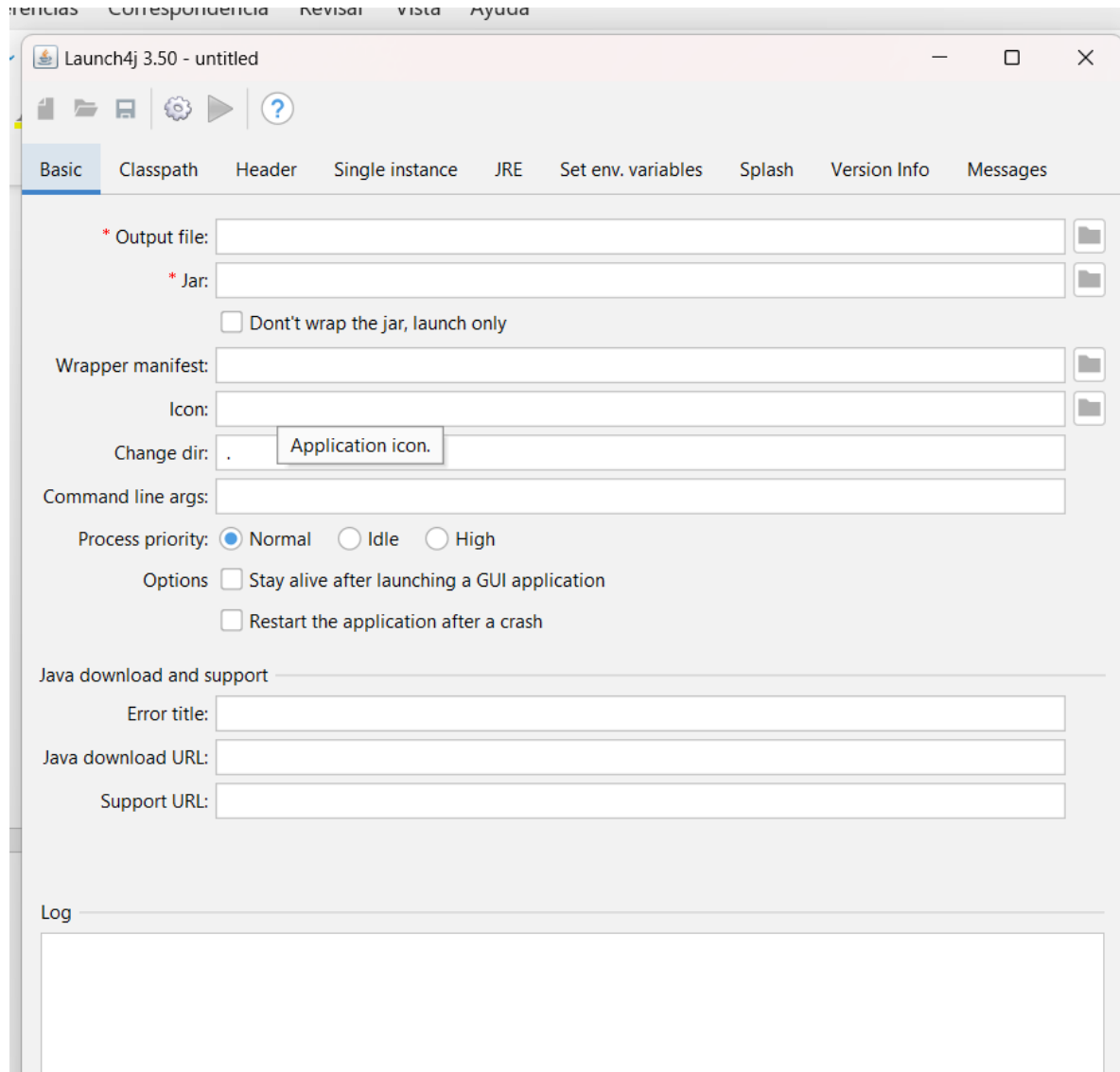
Nombre	Fecha de modificación	Tipo	Tamaño
.git	6/11/2024 11:25	Carpeta de archivos	
build	6/11/2024 11:47	Carpeta de archivos	
dist	6/11/2024 11:48	Carpeta de archivos	
imagenes	4/11/2024 00:02	Carpeta de archivos	
nbproject	4/11/2024 00:01	Carpeta de archivos	

Nombre	Fecha de modificación	Tipo	Tamaño
lib	6/11/2024 11:47	Carpeta de archivos	
ListaProductos	6/11/2024 11:48	Microsoft Edge PD...	2 KB
ProyectoBD	6/11/2024 11:47	Executable Jar File	454 KB
README	6/11/2024 11:47	Documento de tex...	2 KB

Este archivo nos servirá para crear nuestro .exe

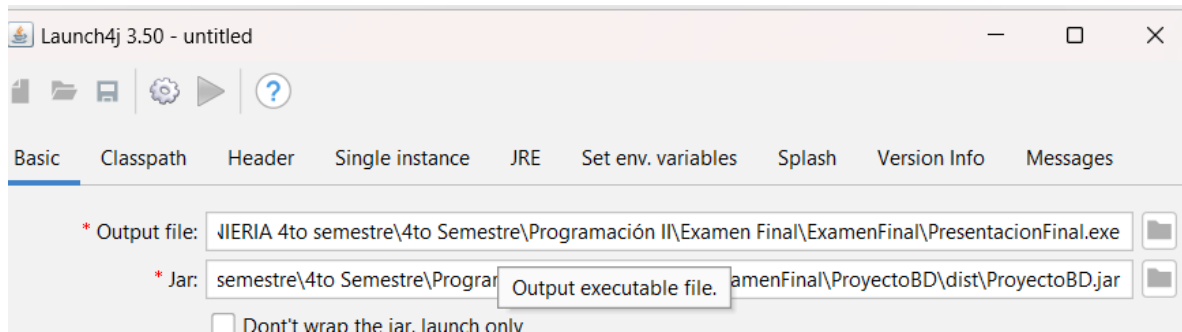
Nota: El pdf no es el mismo, son archivos diferentes ya que uno es del código fuente y el otro es sobre el ejecutable

Para crear un .exe se necesita una aplicación extra

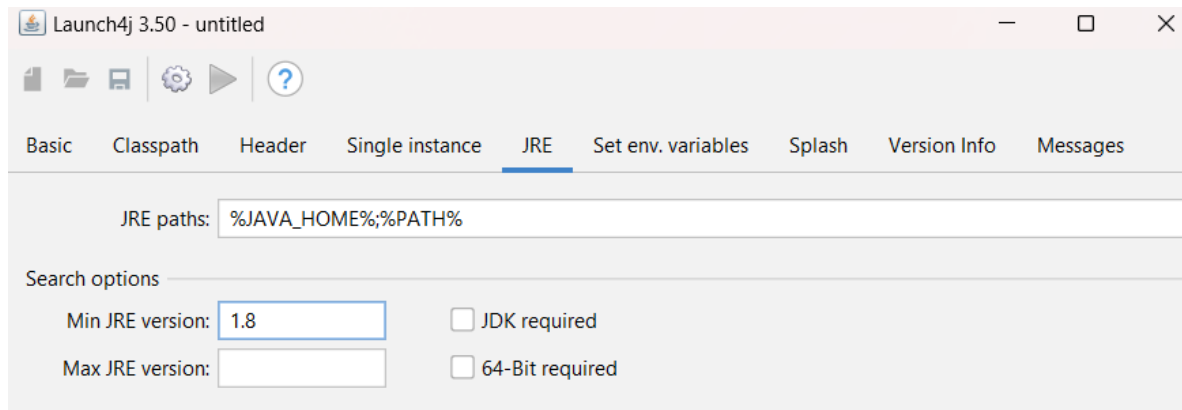


En **output file** agregamos el nombre que tendrá nuestro archivo final

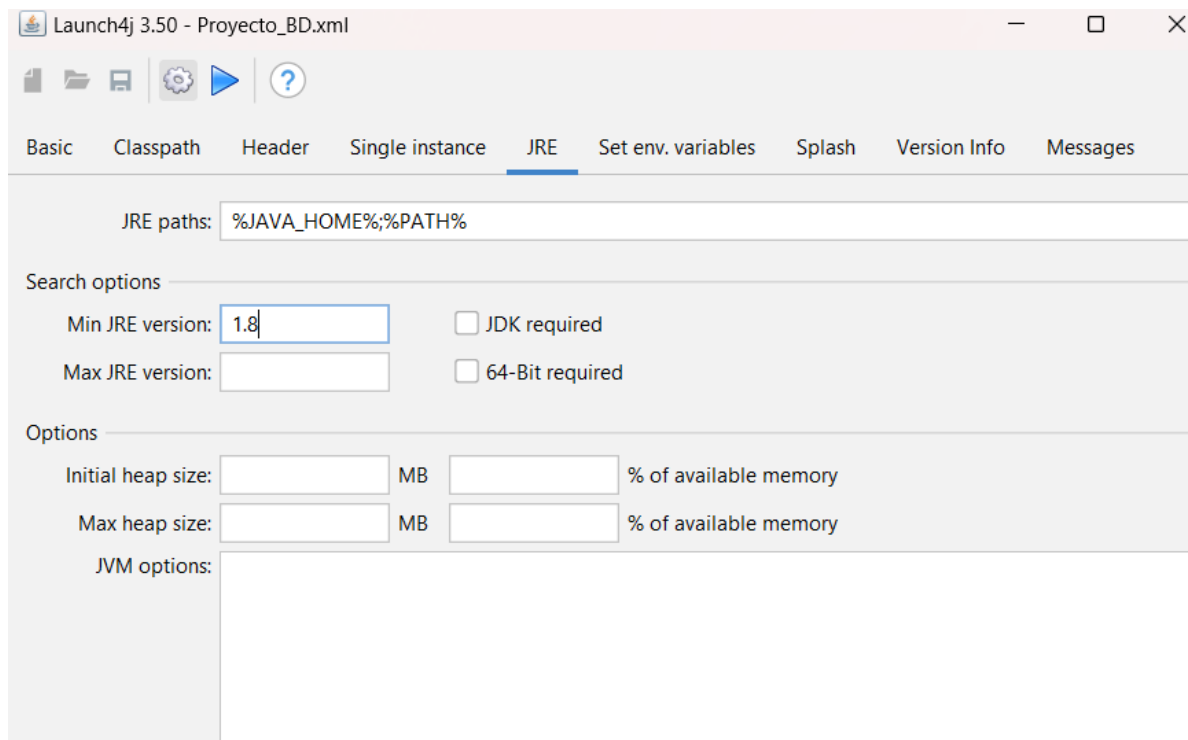
Y en .JAR solo agregamos el archivo que creamos



En JRE agregamos la versión con la que creamos el proyecto



en este caso 1.8



Y creamos el ejecutable