

CREACIÓN DE PAQUETES DE INSTALACIÓN

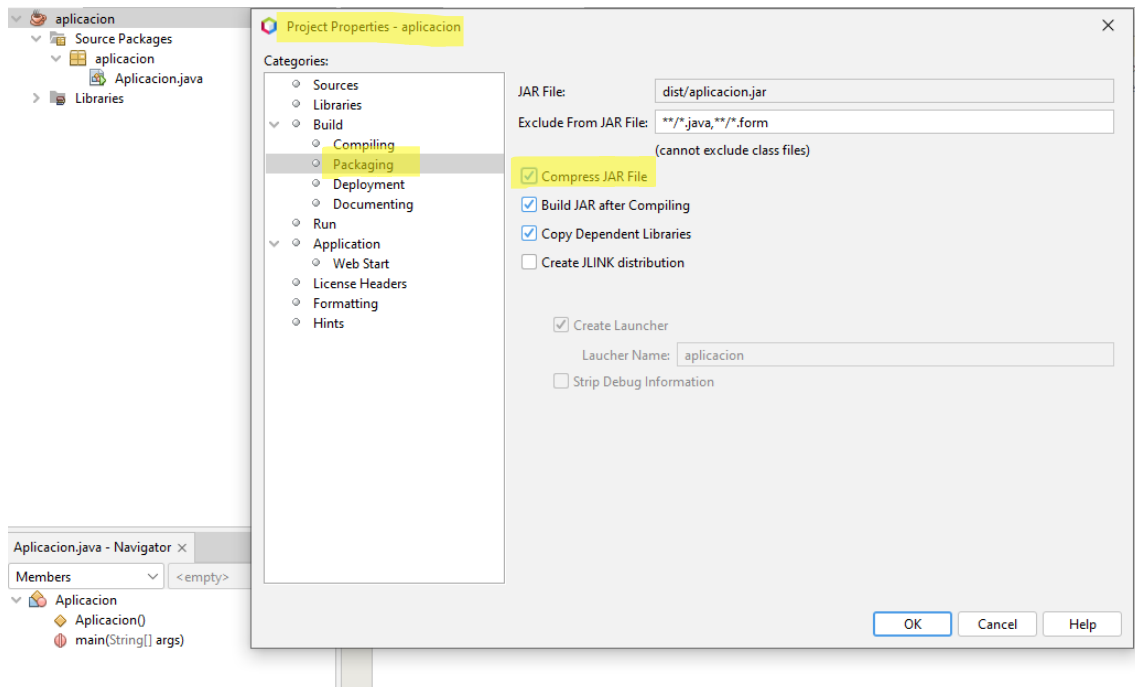
Para realizar la instalación de una aplicación, habitualmente se diseña una interfaz de usuario, también conocida como asistente de instalación, que permitirá escoger entre varias opciones de instalación, como diversos componentes que pueden ser instalados, la ruta concreta en la que se coloca el sistema de archivos, ...

Este tipo de programas realizan una serie de operaciones sobre los archivos contenidos en el paquete de distribución que permiten la instalación de cualquier software de forma automática.

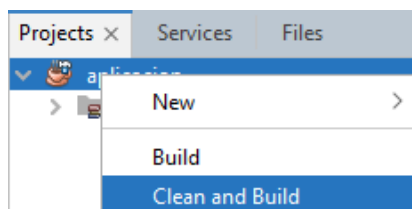
En primer lugar, para crear un asistente de instalación es necesario generar un fichero ejecutable EXE.

Para crear un instalador podemos utilizar la herramienta Launch4j, disponible en launch4j.sourceforge.net

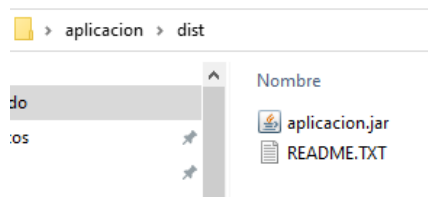
Primero se crea el paquete JAR:



Botón derecho sobre la aplicación → Clean and Build



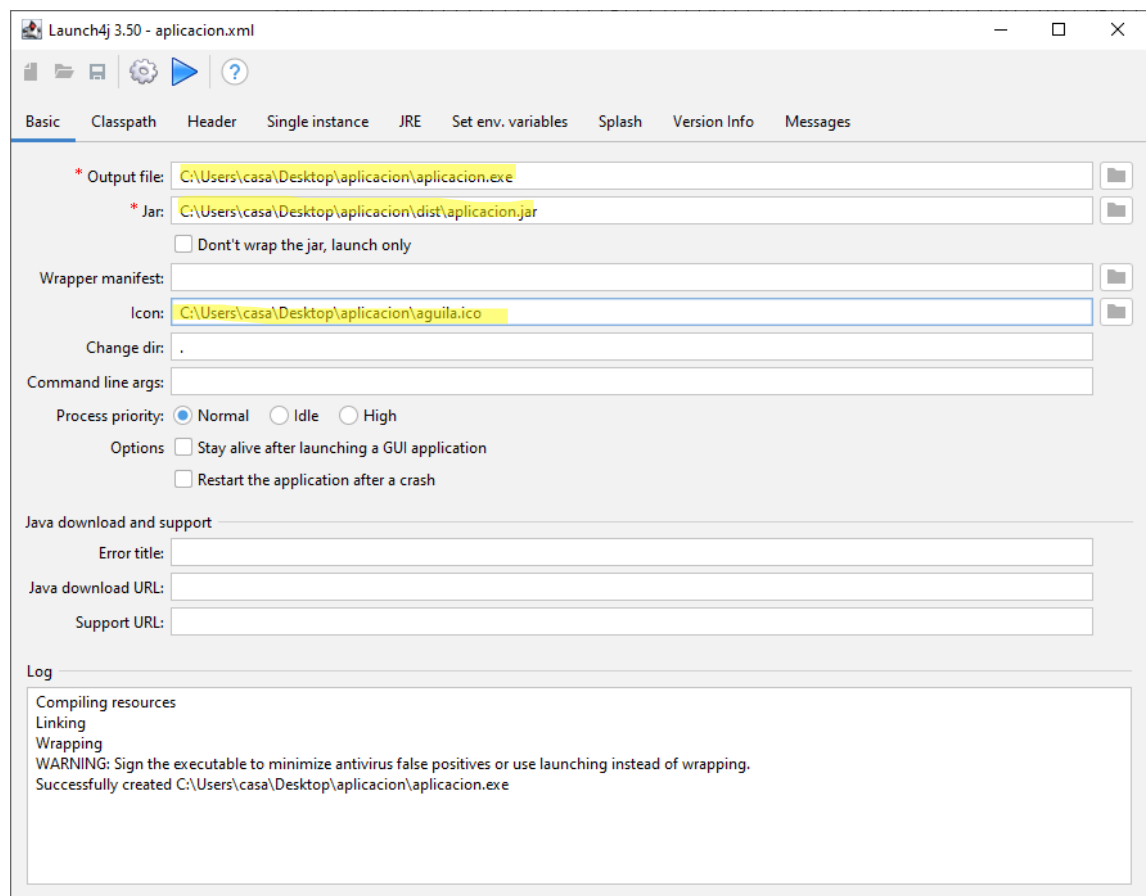
Comprobamos que se ha creado el archivo .jar en la carpeta dist de nuestro proyecto:



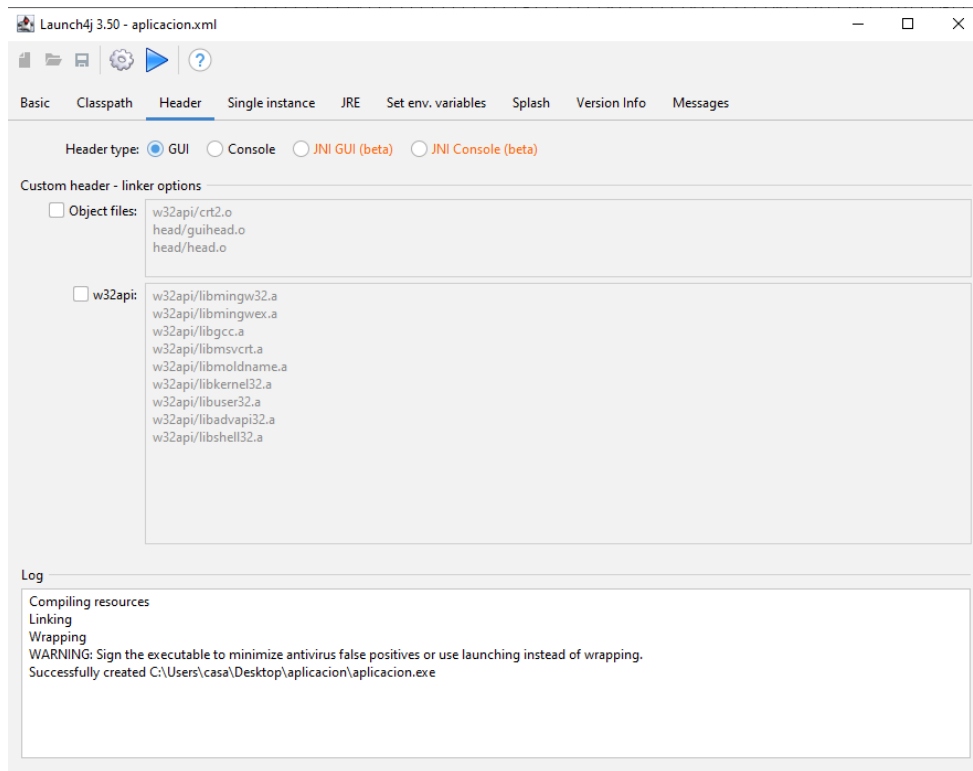
Posteriormente creamos una carpeta en la que se almacenará el resultado de salida del programa. En esta carpeta, en la que se crea el fichero EXE debe contener el paquete JAR del proyecto, librerías, imagen del icono (.ico) del proyecto, y también es aconsejable incluir otra imagen que se mostrará al inicio de la ejecución de la aplicación.

Ejecutamos la aplicación Launch4j, en la pestaña Basic:

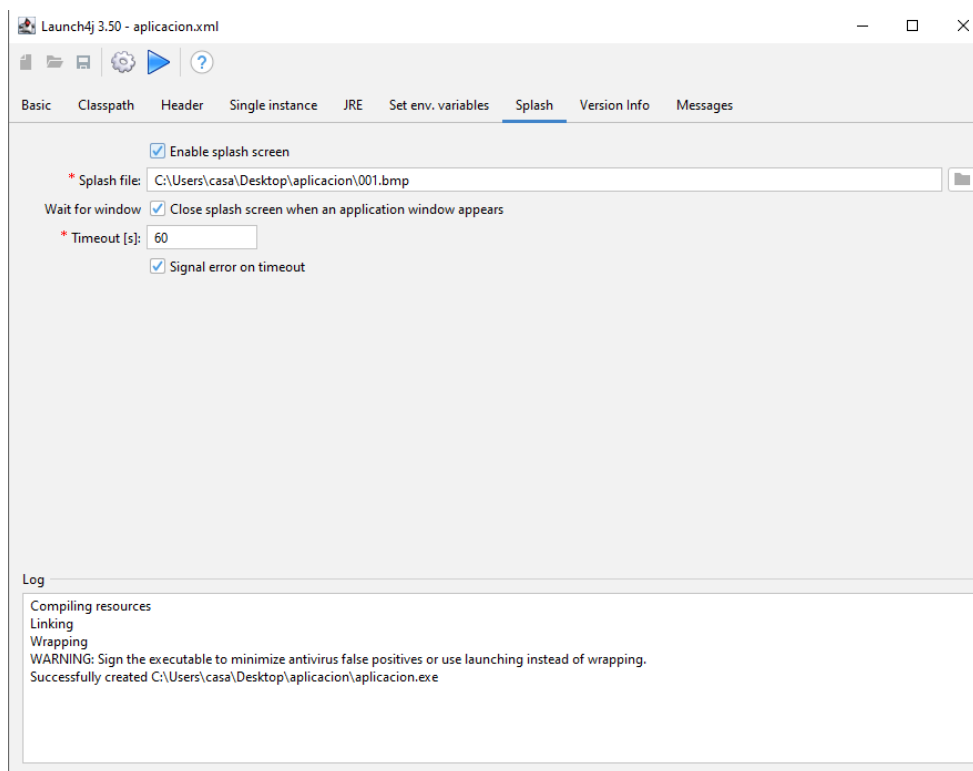
- Seleccionamos la ruta de la carpeta creada anteriormente.
- Indicamos el nombre del archivo EXE
- Indicamos el icono de nuestra aplicación.



En la pestaña Header seleccionamos GUI para que la aplicación se ejecute desde la interfaz gráfica de usuario.

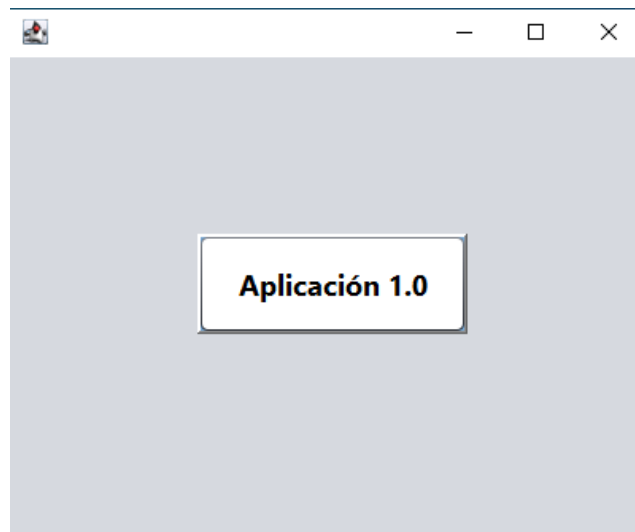


En la pestaña Splash marcamos la casilla Enable splash screen y seleccionamos la imagen que se va a mostrar al usuario cuando comience la ejecución de la aplicación. (este paso es opcional).





Finalmente pulsamos sobre el icono de construcción de la aplicación.



NOTA:

Esta práctica esta hecha con Ant.

Para Maven hay que modificar el archivo pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany</groupId>
  <artifactId>aplicacion2</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
    <exec.mainClass>com.mycompany.aplicacion2.ventana</exec.mainClass>
  </properties>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-jar-plugin</artifactId>
        <version>2.4</version>
        <configuration>
          <archive>
            <manifest>
              <addClasspath>true</addClasspath>
              <mainClass>com.mycompany.aplicacion2.ventana</mainClass>
            </manifest>
          </archive>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

Añadir Este plugin, solo hay que cambiar el nombre de la clase principal.

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <version>2.4</version>
      <configuration>
        <archive>
          <manifest>
            <addClasspath>true</addClasspath>
            <mainClass>com.mycompany.aplicacion2.ventana</mainClass>
          </manifest>
        </archive>
      </configuration>
    </plugin>
  </plugins>
</build>
```

Ejercicio:

Tenéis que hacer un manual donde explicáis todos los pasos necesarios del proceso de creación del instalador de una aplicación que tengáis creada con Netbeans, utilizando la herramienta Launch4j.