

Título de los encabezados
a dos líneas



Σavante

Tarea
Evaluación
Módulo 11

Fernando Tejado
Muñoz

0
Σavante



Título del índice

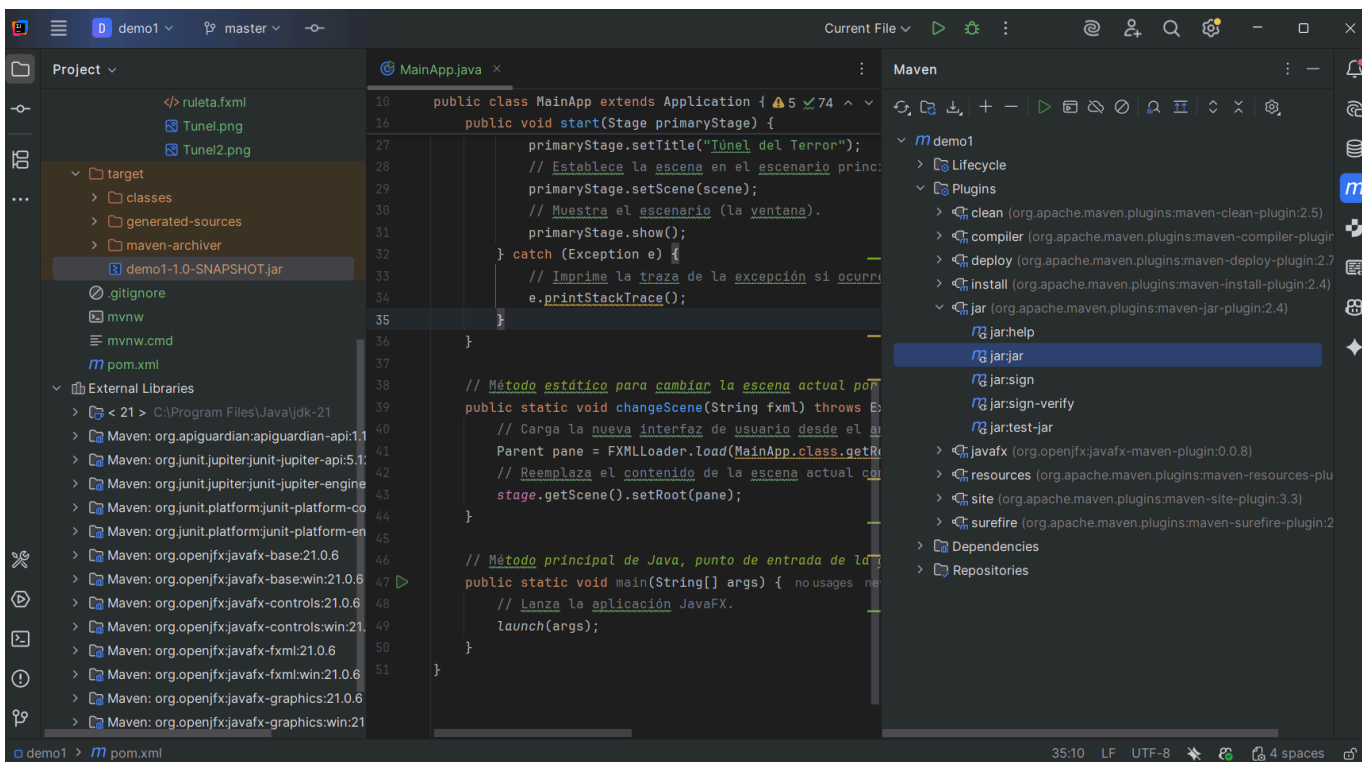
1. Generación del JAR Ejecutable	2
2. Creación del Ejecutable (.exe) con Launch4j	3
3. Instalador con Inno Setup	7
4. Problemas Encontrados y Soluciones	8



1. Generación del JAR Ejecutable

Para cumplir con el primer requisito de la rúbrica, se ha generado un "Fat JAR" utilizando Maven.

- **Proceso:** Se utilizó el plugin `maven-shade-plugin` en el archivo `pom.xml` para empaquetar todas las dependencias de JavaFX dentro de un único archivo ejecutable.
- **Clase Lanzadora:** Se creó una clase `Launcher.java` independiente que no hereda de `Application` para evitar conflictos con el entorno de ejecución de JavaFX al lanzar el JAR desde consola.
- **Verificación:** Se comprobó la correcta generación mediante el comando `java -jar demo1-1.0-SNAPSHOT.jar`, verificando que la aplicación gráfica se abre sin errores.

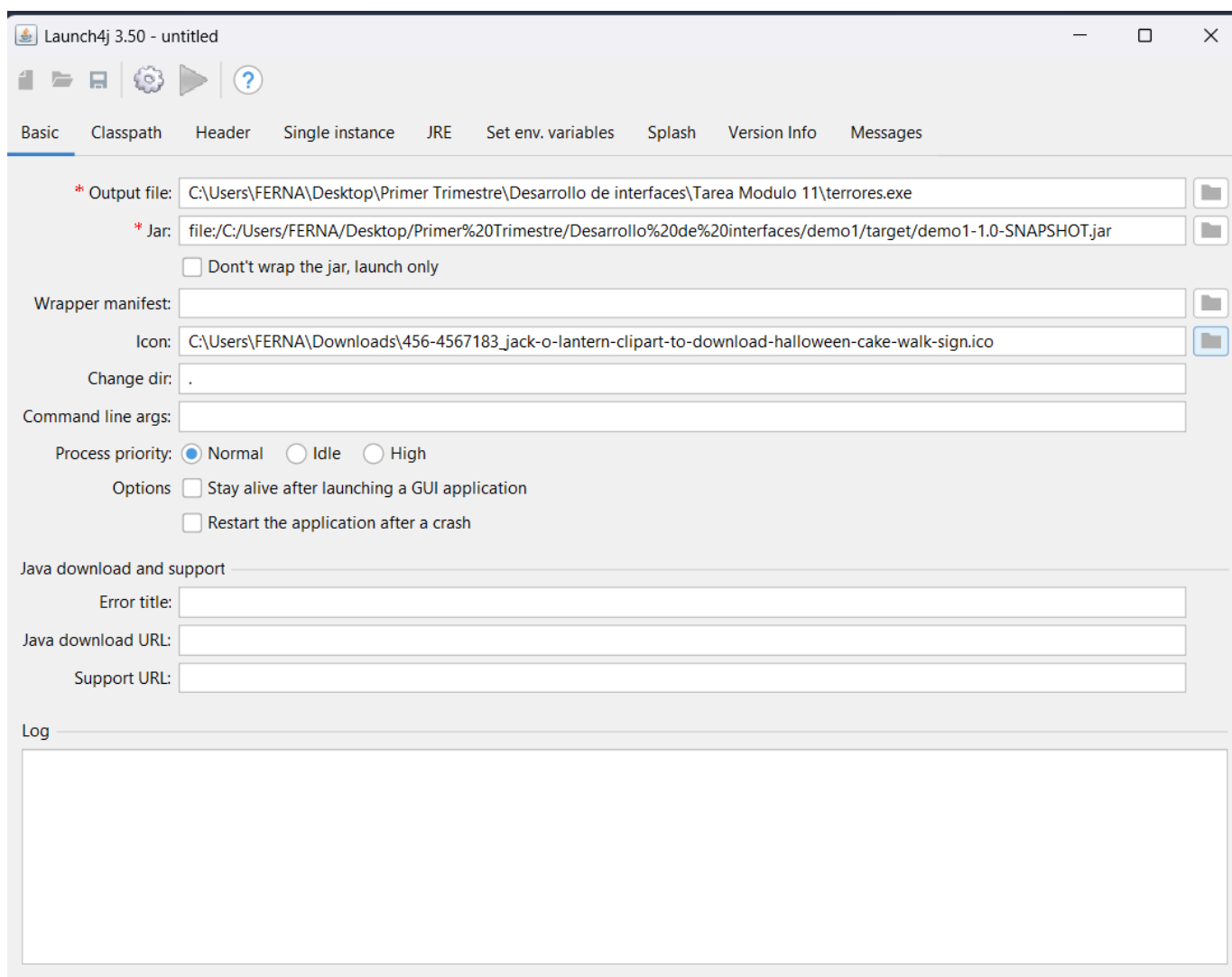




2. Creación del Ejecutable (.exe) con Launch4j

En esta fase se transformó el JAR en un ejecutable nativo de Windows.

- **Configuración:** Se configuró Launch4j en modo **"GUI"** para asegurar que la aplicación abra directamente sin mostrar la consola de comandos, mejorando la experiencia de usuario.
- **Portabilidad (Bundled JRE):** Para garantizar que la aplicación funcione en equipos sin Java instalado, se utilizó la opción **"Bundled JRE path"**, vinculando una carpeta **runtime** que contiene el JDK 21 necesario para la ejecución.



Título de los encabezados a dos líneas



Launch4j 3.50 - untitled

Basic Classpath Header Single instance **JRE** Set env. variables Splash Version Info Messages

JRE paths: %JAVA_HOME%;%PATH%

Search options

Min JRE version: 21 ☐ JDK required

Max JRE version: ☐ 64-Bit required

Options

Initial heap size: MB % of available memory

Max heap size: MB % of available memory

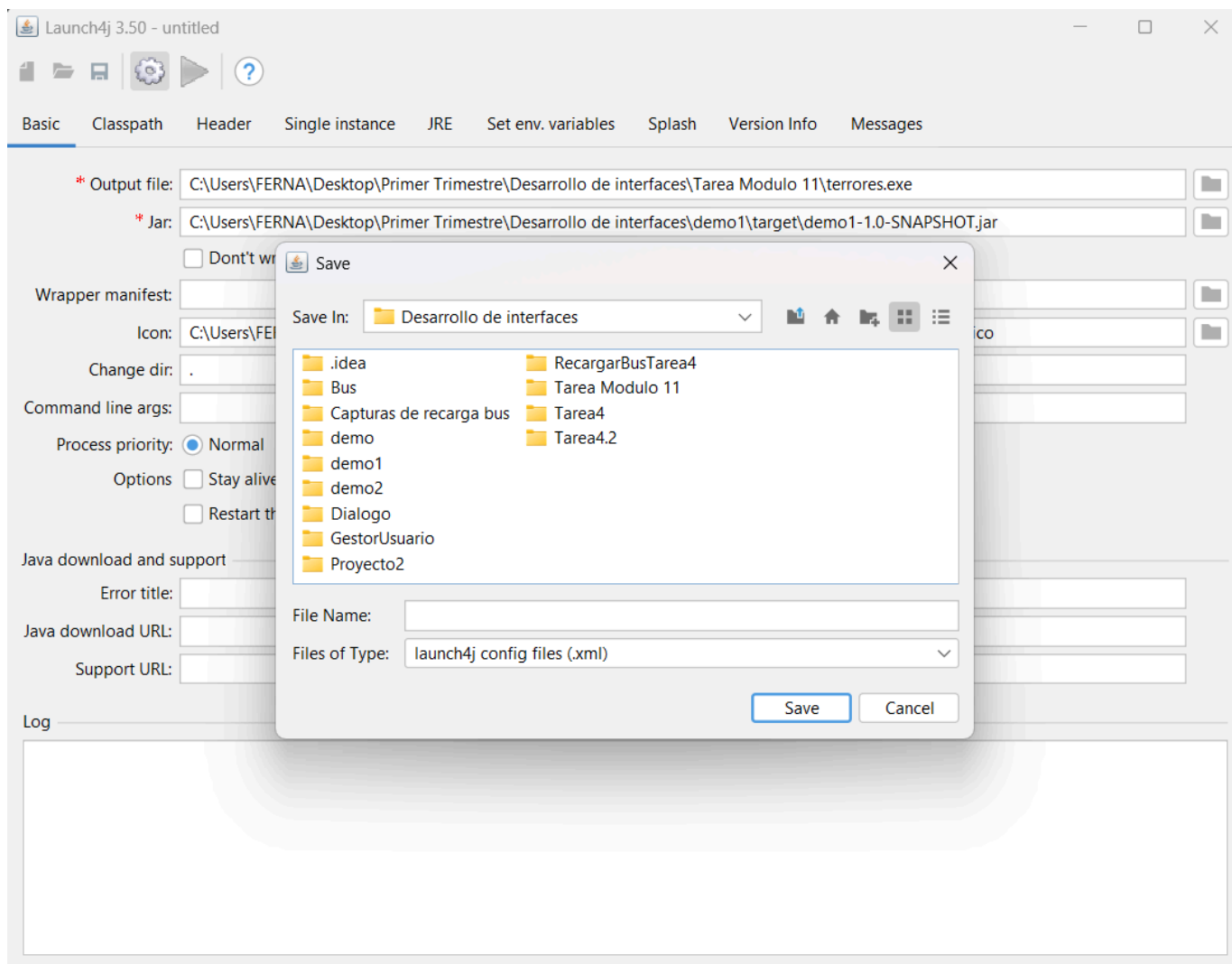
JVM options:

Variables / registry: EXEDIR

* Environment var:

Log

Título de los encabezados a dos líneas



Título de los encabezados a dos líneas



Launch4j 3.50 - launchterror.xml

Basic Classpath Header Single instance JRE Set env. variables Splash Version Info Messages

* Output file: C:\Users\FERNA\Desktop\Primer Trimestre\Desarrollo de interfaces\Tarea Modulo 11\terrores.exe

* Jar: C:\Users\FERNA\Desktop\Primer Trimestre\Desarrollo de interfaces\demo1\target\demo1-1.0-SNAPSHOT.jar

☐ Don't wrap the jar, launch only

Wrapper manifest:

Icon: C:\Users\FERNA\Downloads\456-4567183_jack-o-lantern-clipart-to-download-halloween-cake-walk-sign.ico

Change dir: .

Command line args:

Process priority: ☒ Normal ☐ Idle ☐ High

Options ☐ Stay alive after launching a GUI application

☐ Restart the application after a crash

Java download and support

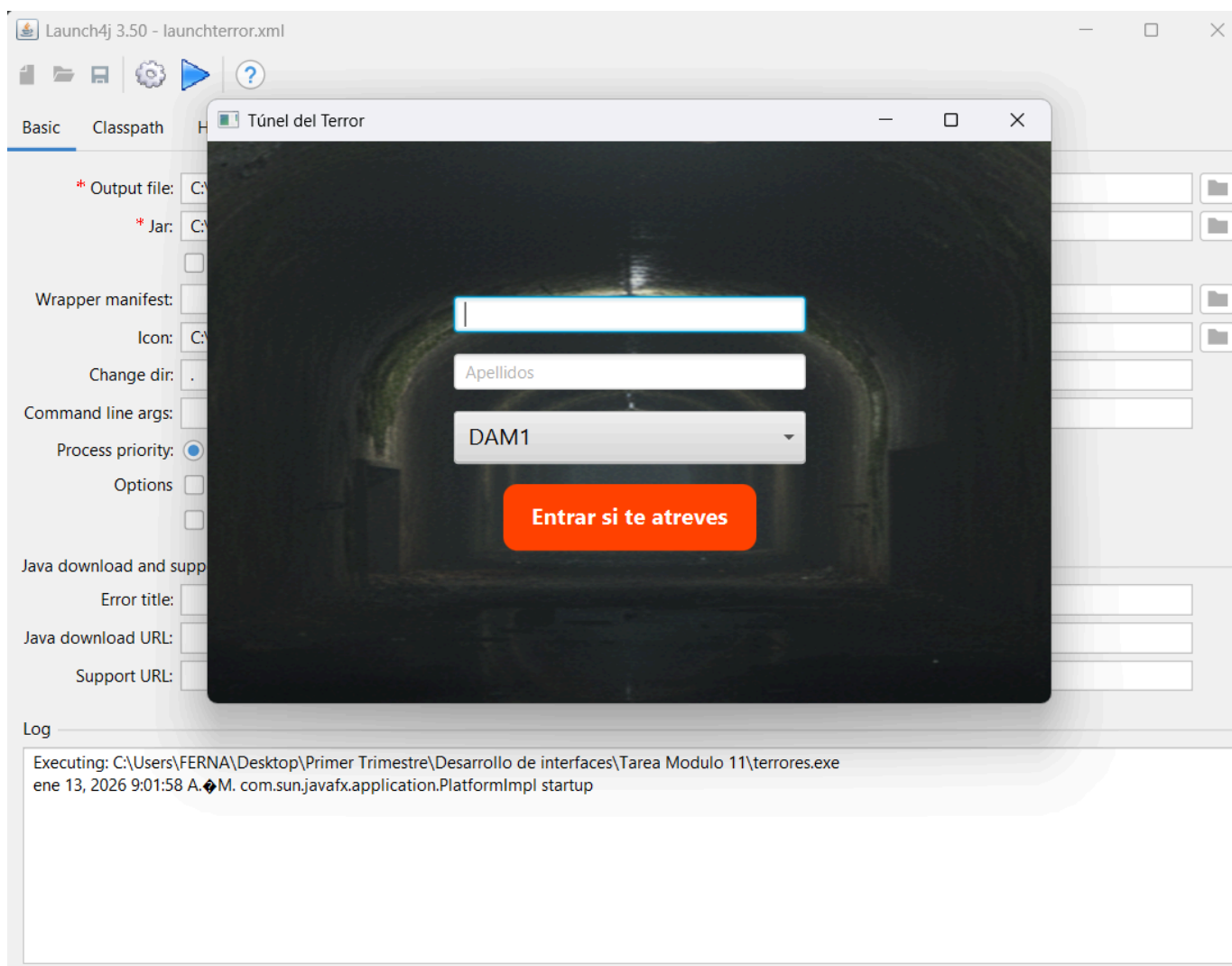
Error title:

Java download URL:

Support URL:

Log

Compiling resources
Linking
Wrapping
WARNING: Sign the executable to minimize antivirus false positives or use launching instead of wrapping.
Successfully created C:\Users\FERNA\Desktop\Primer Trimestre\Desarrollo de interfaces\Tarea Modulo 11\terrores.exe



3. Instalador con Inno Setup

Se ha desarrollado un instalador profesional que gestiona todo el ciclo de vida del software.

- **Ubicación:** El instalador sitúa los archivos en `C:\Program Files\TerrorresApp`, cumpliendo con los estándares del sistema.
- **Funcionalidades:** El asistente permite crear accesos directos en el escritorio y gestiona la desinstalación completa desde el panel de control de Windows, eliminando archivos y entradas de registro.
- **Originalidad:** Se personalizó el instalador con un icono temático de Halloween (`.ico`) y se incluyó un archivo de licencia para dar un acabado profesional al producto.



The screenshot shows the Inno Setup Compiler 6.7.0 interface. The main window displays the 'Main Script' editor with the following code:

```
Root: HKA; Subkey: "Software\Classes\{#MyAppAssocKey}"; ValueType: string; ValueName: ""; ValueData: "{#MyAppAssocName}"; Flags: uninsdeletekey
Root: HKA; Subkey: "Software\Classes\{#MyAppAssocKey}\DefaultIcon"; ValueType: string; ValueName: ""; ValueData: "{app}\{#MyAppExeName},0"
Root: HKA; Subkey: "Software\Classes\{#MyAppAssocKey}\shell\open\command"; ValueType: string; ValueName: ""; ValueData: "\"{app}\{#MyAppExeName}\" \"%*%\"";

[Icons]
Name: "{autoprograms}\{#MyAppName}"; Filename: "{app}\{#MyAppExeName}"
Name: "{autodesktop}\{#MyAppName}"; Filename: "{app}\{#MyAppExeName}"; Tasks: desktopicon

[Run]
Filename: "{app}\{#MyAppExeName}"; Description: "{cm:LaunchProgram,{#StringChange(MyAppName, '&', '&&')}}"; Flags: nowait postinstall skipifsilent
```

The Compiler Output window at the bottom shows the following messages:

```
Parsing [Run] section, line 58
Parsing [Files] section, line 44
Creating setup files
Verification successful
Compressing: C:\Users\FERNA\Desktop\Primer Trimestre\Desarrollo de interfaces\Tarea Modulo 11\terrores.exe
Compressing Setup program executable
Updating version info (Setup.exe)
Updating manifest (Setup.exe)

*** Finished. [11:47:19, 00:02,735 elapsed]
```

4. Problemas Encontrados y Soluciones

- **Error de Atributo Manifest:** Inicialmente, Launch4j no detectaba la clase principal. Se solucionó configurando correctamente el **ManifestResourceTransformer** en el plugin de Maven.
- **Conflicto de Módulos:** La presencia del archivo **module-info.java** generaba errores de "module not found" en entornos sin módulos configurados. Se optó por eliminar este archivo y realizar una configuración "plana" de dependencias para asegurar la compatibilidad total del JAR empaquetado.