

Manual de Usuario

Compilador JAVA

1. Introducción

El Compilador JAVA es una herramienta gráfica para empaquetar aplicaciones Java en:

- App-Image (imagen de aplicación ejecutable sin instalador).
- Instalador para Windows (.exe o .msi), con opciones de firma digital y verificación.

Su objetivo es simplificar el uso de `jpackage` y `signtool` mediante una interfaz amigable, evitando el trabajo manual en la línea de comandos.

2. Requisitos

- Windows 10/11 (x64).
- JDK con `jpackage` disponible (recomendado JDK 21+).
- Windows 10 SDK con `signtool.exe` (para firmar/verificar binarios).
- Proyecto Java compilado (JAR principal y dependencias).
- Opcional: certificado PFX y contraseña para firma. Si no se dispone de uno, la aplicación puede generar un certificado auto-firmado.

3. Preparación del proyecto

Compila tu app y genera el JAR principal (por ejemplo, con Maven o Gradle):

- Maven: `mvn clean package` → JAR en `target/`.
- Gradle: `gradle clean build` → JAR en `build/libs/`.



4. Interfaz y campos

- Jpackage.exe: ruta al binario `jpackage.exe` del JDK.
- Botón “buscar”: abre selector de archivos.
- Botón “prueba”: valida que `jpackage` si está y donde está en el pc y que se ejecute correctamente.
- Signtool.exe: ruta al binario `signtool.exe` del Windows SDK.
- “buscar”: selector de archivos.
- “prueba”: valida que `signtool` está y donde está en el pc y que esté operativo.

Jpackage.exe:	<input type="text" value="C:\Program Files\OpenLogic\jdk-21.0.8.9-hotspot\bin\jpackage.exe"/>	<input type="button" value="buscar"/>	<input type="button" value="prueba"/>
Signtool.exe:	<input type="text" value="Program Files (x86)\Windows Kits\10\bin\10.0.26100.0\x64\signtool.exe"/>	<input type="button" value="buscar"/>	<input type="button" value="prueba"/>

- En la pantalla negra se mostrará información relevante al botón prueba, tal como versión instalada y la dirección del archivo.

```
== Test package & java ==
[where jpackage]
C:\Program Files\OpenLogic\jdk-21.0.8-hotspot\bin\package.exe
[jpackage --version]
21.0.8
[where java]
C:\Program Files\Common Files\Oracle\Java\javapath\java.exe
C:\Program Files (x86)\Common Files\Oracle\Java\javapath\java.exe
C:\Program Files\OpenLogic\jdk-21.0.8-hotspot\bin\java.exe
java -version
```

- App Name: nombre público de la aplicación (p. ej., "Mi App").
- Input (target): carpeta de entrada con los artefactos (p. ej., `target`).
- Main JAR: nombre del JAR principal (p. ej., `MiApp.jar`).

App Name:	<input type="text" value="Mi App"/>	<input type="button" value="buscar"/>
Input (target):	<input type="text" value="target"/>	
Main JAR:	<input type="text" value="MiApp.jar"/>	

- Main Class: clase principal totalmente calificada (p. ej., `com.mycompany.Miapp.miapp`).
- App Version: versión de la app (p. ej., `1.0.0`).
- Owner: nombre del propietario o editor.
- Dest. (output): carpeta de salida donde se generarán los artefactos (p. ej., `out`).
- Icon (.ico): archivo .ico para el ejecutable/instalador (recomendado 256×256).

Main Class:	<input type="text" value="com.mycompany.Miapp.miapp"/>	<input type="button" value="buscar"/>
App Version:	<input type="text" value="1.0.0"/>	
Owner:	<input type="text" value="Mi Nombre"/>	
Dest. (output):	<input type="text" value="out"/>	
Icon (.ico):	<input type="text" value="icon.ico"/>	

4.1. Opciones de Windows

- win-shortcut: crea acceso directo.
- win-menu: añade entrada al menú Inicio.
- win-dir-chooser: permite elegir directorio de instalación.
- win-per-user-install: instala por usuario (en lugar de por máquina).

☒ win-shortcut
 ☒ win-menu
 ☒ win-dir-chooser
 ☒ win-per-user-install

4.2. Certificados y firma

- CN auto-firmado: “Common Name” para generar un certificado propio (p. ej., `angelvalecillos`).
- salida .pfx: ruta donde se guardará el PFX auto-firmado.
- PFX actual: ruta del certificado PFX que se usará para firmar.
- PFX password: contraseña del PFX.
- Timestamp URL: URL del servidor de sellado de tiempo (p. ej., `http://timestamp.digicert.com`).

CN auto-firmado:	<input type="text" value="angelvalecillos"/>	
salida .pfx:	<input type="text" value="C:\tipo.pfx"/>	<input type="button" value="buscar"/>
PFX actual:	<input type="text" value="C:\tipo.pfx"/>	<input type="button" value="buscar"/>
PFX password:	<input type="password" value="••••••••"/>	
Timestamp URL:	<input type="text" value="http://timestamp.digicert.com"/>	app-image

4.3. Acciones y botones laterales

- Cargar config.: carga una configuración previa (rutas, campos y opciones).
- Guardar config.: guarda la configuración actual para reutilizarla.

<input type="button" value="Cargar config."/>
<input type="button" value="Guardar config."/>

- Sección “app-image”:

- Crear (.exe): genera la imagen de aplicación ejecutable.
- firmar: firma el binario resultante con `signtool` usando el PFX.
- verificar: valida la firma del binario.

- Sección “instalador”:

- Crear (.exe): construye el instalador de Windows con `jpackage`.
- firmar: firma el instalador.
- verificar: comprueba la firma del instalador.
- Generar auto-firmado (.pfx): crea un certificado PFX auto-firmado con el CN indicado.
- clear: limpia la pantalla negra de datos .
- Acerca: información del programa.

app-image	instalador
<input type="button" value="Crear (.exe)"/>	<input type="button" value="Crear (.exe)"/>
<input type="button" value="firmar"/>	<input type="button" value="firmar"/>
<input type="button" value="verificar"/>	<input type="button" value="verificar"/>
<input type="button" value="Generar auto-firmado(.pfx)"/>	

5. Flujos de trabajo

5.1. Generar App-Image

1. Completa “Jpackage.exe” y “Signtool.exe” y prueba ambos.
2. Rellena “App Name”, “Input (target)”, “Main JAR”, “Main Class”, “App Version”, “Owner”, “Dest. (output)” e “Icon (.ico)”.
3. Marca las opciones de Windows que necesites.
4. Pulsa “Crear (.exe)” en la columna “app-image”.
5. Opcional: “firmar” y luego “verificar” para añadir y validar firma digital.

5.2. Generar Instalador

1. Con los campos completos, pulsa “Crear (.exe)” en la columna “instalador”.
2. Opcional: “firmar” y “verificar” para garantizar confianza del instalador.

5.3. Firmar con certificado propio

1. En “PFX actual” selecciona tu `.pfx` y escribe la “PFX password”.
2. Usa “firmar” en la sección correspondiente (app-image o instalador).
3. Ejecuta “verificar” para confirmar firma y sello de tiempo.


5.4. Generar certificado auto-firmado

1. Escribe un “CN auto-firmado” y la ruta “salida .pfx”.
2. Pulsa “Generar auto-firmado (.pfx)”.
3. Asigna el nuevo PFX en “PFX actual” y su contraseña (si corresponde).
4. Firma y verifica tus binarios.

6. Buenas prácticas

- Usa `Main Class` completamente calificada.
- Mantén un icono `.ico` de alta resolución (mínimo 256×256).
- Conserva un “Dest. (output)” limpio para cada build.
- Firma siempre instaladores públicos y utiliza “Timestamp URL” para evitar caducidad de firma.

7. Solución de problemas

- No se encuentra ``jpackage``:
 - Verifica instalación del JDK y la ruta a ``jpackage.exe``.
 - Prueba con el botón “prueba”.
- No se encuentra ``signtool``:
 - Instala Windows 10 SDK y selecciona la ruta correcta (x64).
- Error en ``Main Class``:
 - Confirma el nombre de paquete/clase y que el JAR lo contiene.
- Instalador no se crea:
 - Revisa permisos de escritura en “Dest. (output)”.
 - Comprueba que  `Main JAR` y `Icon (.ico)` existan.
- Firma inválida:
 - Verifica “PFX password” y la validez del PFX.
 - Prueba otro “Timestamp URL”.

8. Estructura de salida

- app-image: ejecutable y carpeta de imagen de aplicación dentro de ``out/``.
- instalador: archivo ``exe`` (o ``msi``) dentro de ``out/``.
- logs: mensajes visibles en el panel de la aplicación; usa “clear” para reiniciar la vista.

9. Seguridad y distribución

- Al distribuir un instalador firmado, los sistemas Windows mostrarán mayor confiabilidad.
- Evita publicar certificados auto-firmados en producción; solo para pruebas.
- Verifica la integridad con “verificar” antes de compartir binarios.

10. Soporte

Para dudas o mejoras, abre un issue en el repositorio [angelvalecillos/compilador-JAVA](https://github.com/angelvalecillos/compilador-JAVA) o contacta al mantenedor.