**JAVA HELP**

**2º DAM**

**DESARROLLO DE INTERFACES**

**ALFONSO RINCÓN CUERVA**

Contenido

ACTIVIDAD 1

ACTIVIDAD 2

ACTIVIDAD 3

ACTIVIDAD 4

ACTIVIDAD 5

ACTIVIDAD 6

ACTIVIDAD 7

[**BIBLIOGRAFÍA** 1](#_Toc189776066)

# ACTIVIDAD 1

BUSCA EN INTERNET AL MENOS TRES EJEMPLOS DE PAQUETES EXE Y MSI. COMPARA LAS SEMEJANZAS Y DIFERENCIAS EN EL PROCESO DE INSTALACIÓN DE CADA UNO DE ELLOS

**Ejemplos de EXE:**

Estos nos siguen un estándar específico de instalación, y pueden llegar a instalar archivos adicionales. Permiten más personalización.

1. **Google Chrome Installer.**
2. **Adobe Reader Installer.**
3. **WinRAR.**

**Ejemplos de MSI:**

Permiten menos personalización y siguen un estándar de instalación de Windows.

1. **Microsoft Office.**
2. **Mozilla Firefox MSI.**
3. **Skype for Business.**

**Semejanzas en el proceso de instalación:**

* Ambos formatos están diseñados para instalar software en sistemas Windows.
* Ambos pueden ofrecer interfaces gráficas que guían al usuario a través del proceso de instalación.

**Diferencias en el proceso de instalación:**

1. **Control y Flexibilidad:** los instaladores EXE son muy personalizables y permiten a los desarrolladores definir procesos de instalación específicos, ejecutar scripts adicionales o verificar condiciones previas antes de la instalación.

Los archivos MSI siguen un estándar establecido por Microsoft y son gestionados por el servicio Windows Installer, lo que da menos flexibilidad para personalizaciones complejas.

1. **Seguridad:** los archivos EXE pueden ser más susceptibles a contener código malicioso, mientras que los MSI son considerados más seguros, ya que el proceso de instalación es supervisado por Windows Installer.
2. **Despliegue en Entornos Corporativos:** la implementación de archivos EXE en redes corporativas puede ser más desafiante debido a la falta de estandarización y opciones limitadas para instalaciones silenciosas o desatendidas, mientras que los archivos MSI son ideales para despliegues masivos en entornos empresariales, ya que se integran fácilmente con herramientas de administración de software.
3. **Gestión de Instalaciones Concurrentes:** los EXE pueden permitir múltiples instalaciones simultáneas, a diferencia de los MSI, que suelen permitir una sola instalación a la vez para evitar conflictos.

# ACTIVIDAD 2

REALIZA UN EMPAQUETADO CON FORMATO DEB. PARA ELLO DEBES SEGUIR ESTAS INSTRUCCIONES: 1. INSTALA LA DISTRIBUCIÓN CHECKINSTALL PARA PODER REALIZAR ESTE TIPO DE DISTRIBUCIÓN. • UTILIZA LA LÍNEA DE COMANDOS: SUDO APTGET INSTALL CHECKINSTALL 2. DESDE LA CARPETA EN LA QUE SE ENCUENTRAN TODOS LOS FICHEROS DE PROYECTOS, EJECUTA LA INSTRUCCIÓN: • .CONFIGURE • MAKE • CHECKINSTALL 3. POR ÚLTIMO, SE ABRIRÁ UN ASISTENTE PARA DEFINIR CIERTOS PARÁMETROS PARA LA CREACIÓN DE PAQUETES. SE PULSARÁ LA TECLA ENTER Y SE INICIARÁ LA COMPILACIÓN DEL PAQUETE QUE ESTARÁ LISTO PARA SU DISTRIBUCIÓN.

# ACTIVIDAD 3

ENUMERA Y DESCRIBE CADA UNA DE LAS PANTALLAS QUE SERÍA NECESARIO IMPLEMENTAR EN UN ASISTENTE DE INSTALACIÓN PARA UNA APLICACIÓN QUE PRESENTA UNA CORRECTA INTERACCIÓN CON EL USUARIO.

**1. Pantalla de Bienvenida**

🔹 **Descripción:** contiene el nombre de la aplicación y una breve introducción sobre la instalación. También contendrá elementos como botones de Siguiente, Cancelar y un mensaje de bienvenida.

**2. Pantalla de Términos y Condiciones (EULA):** mostrará el acuerdo de licencia del software, el cual el usuario debe aceptar los términos para continuar con la instalación.Contendrá el texto del acuerdo de licencia y las opciones de "Acepto" o "No acepto".

**3. Pantalla de Selección de Ubicación de Instalación**: permite al usuario elegir la carpeta donde se instalará la aplicación.Contendrá el botón para Examinar y seleccionar una carpeta.

**4. Pantalla de Opciones de Instalación (Personalización):** permite seleccionar qué componentes del software se desean instalar, con opciones como instalación completa, mínima o personalizada.Contendrá casillas de verificación para activar/desactivar componente y el espacio para mostrar el tamaño requerido en disco.

**5. Pantalla de Confirmación de Instalación:** muestra un resumen de las opciones seleccionadas antes de iniciar la instalación.

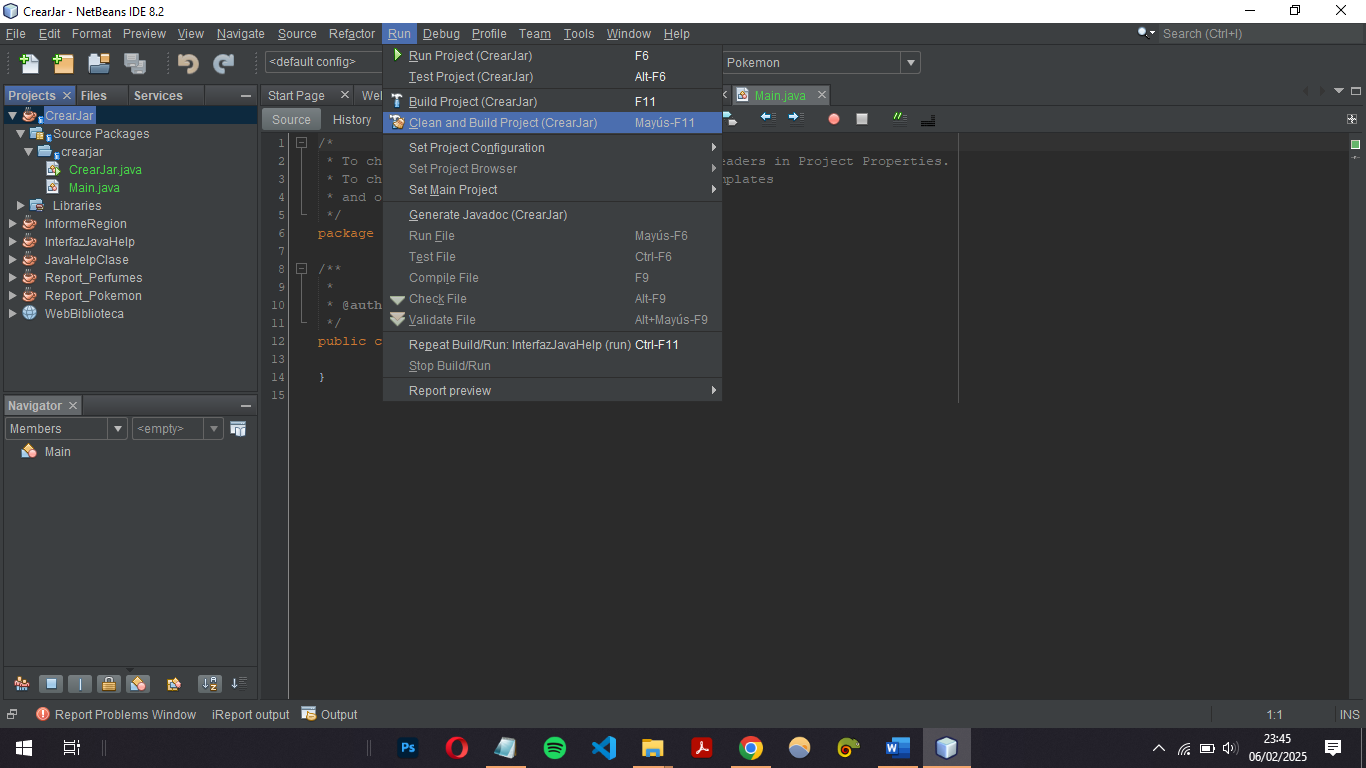
**6. Pantalla de Progreso de Instalación:** muestra el progreso en tiempo real de la instalación del software mediante una barra de carga y mensajes sobre los archivos que se están copiando.

**7. Pantalla de Finalización de Instalación:** informa al usuario que la instalación se ha completado con éxito, con opciones como "Abrir la aplicación" o "Ver el registro de cambios".

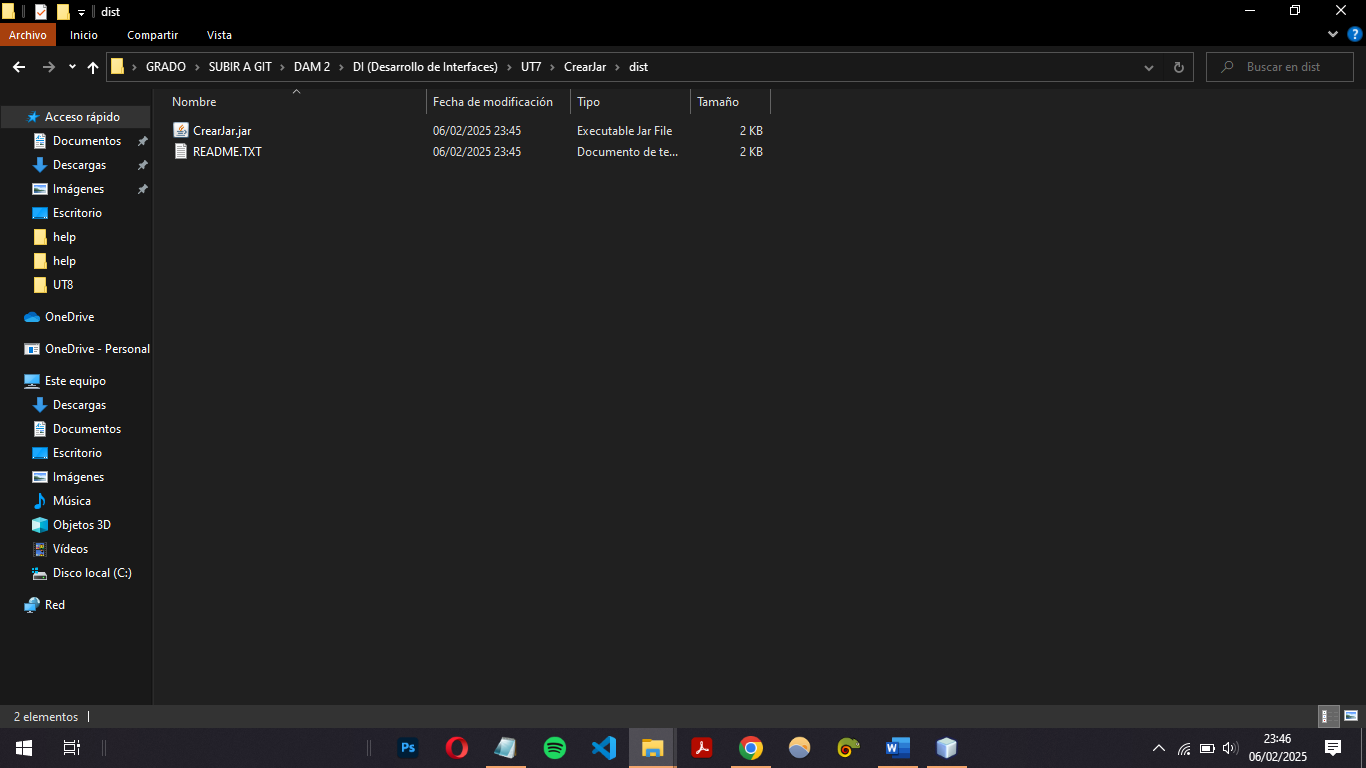
# ACTIVIDAD 4

REALIZA LA FIRMA DE UN FICHERO JAR UTILIZANDO LA HERRAMIENTA JARSIGNER.

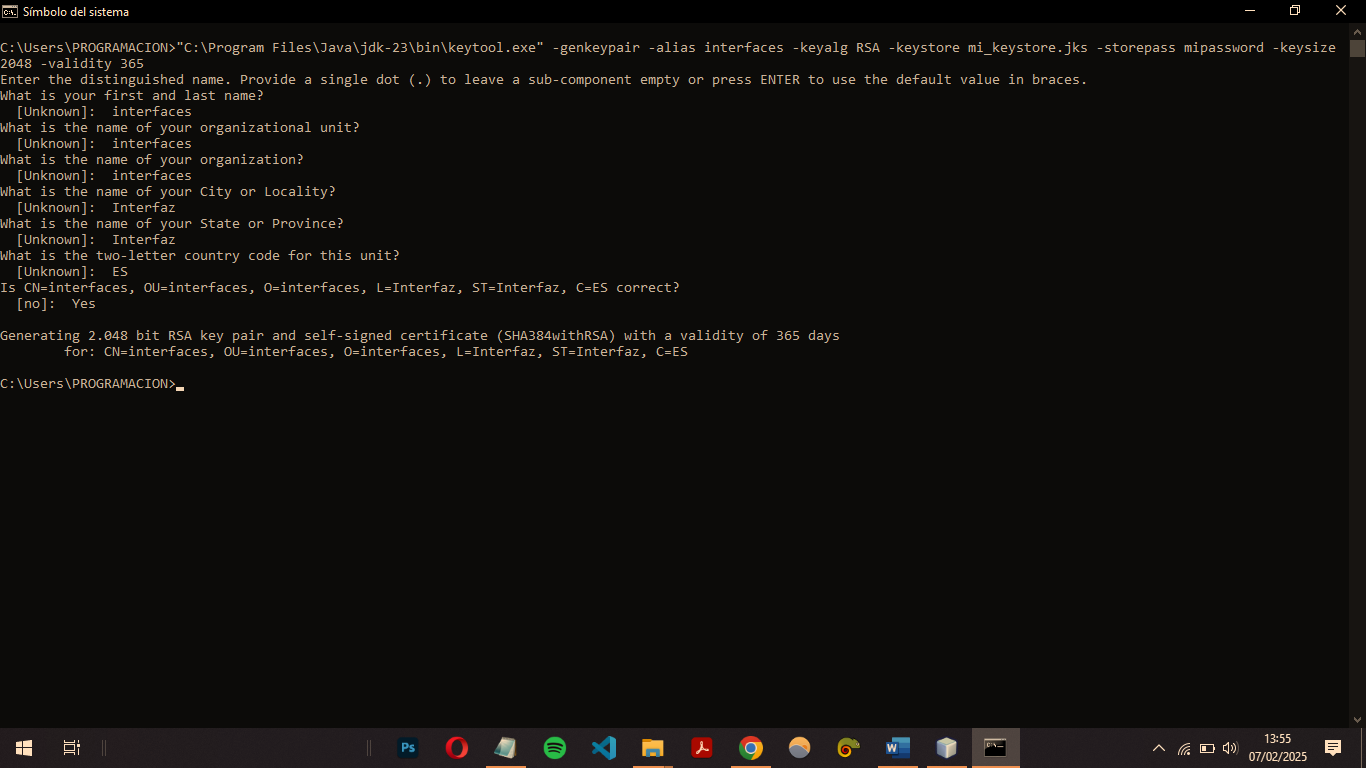
Primero vamos a crear el Jar de una aplicación creada anteriormente. Vamos a Run > Clean and Build Project.



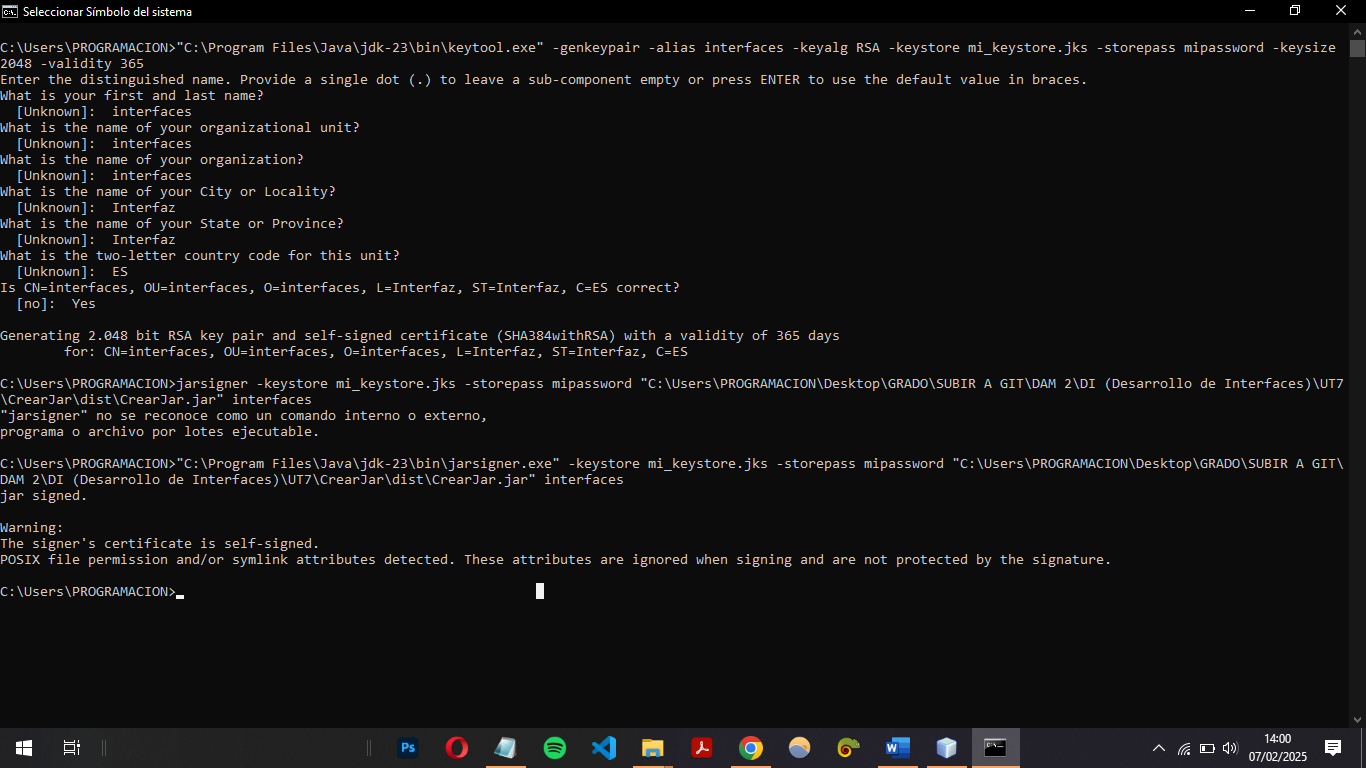
Este se generará dentro de la carpeta dist de la aplicación.



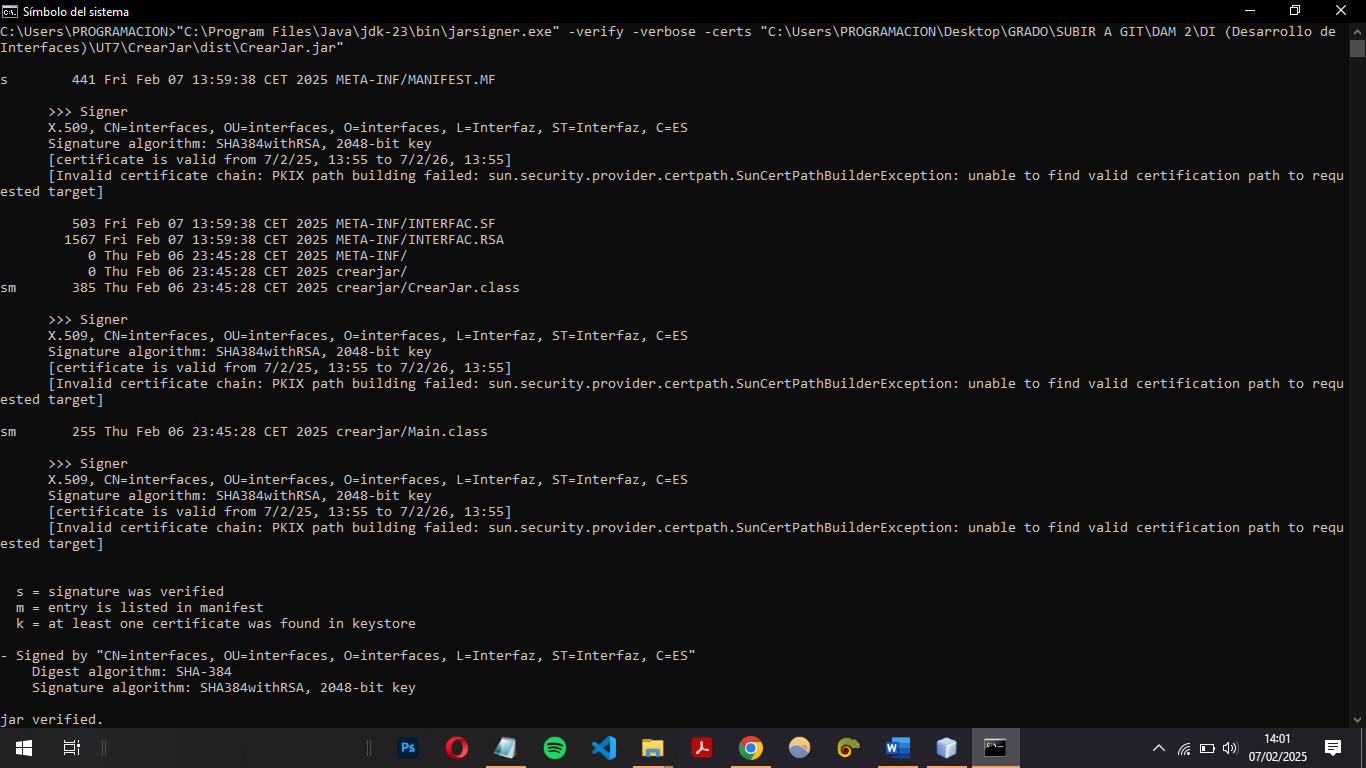
Usamos el siguiente código para generar el keystore, el cual es un almacén de claves digital.



Con el siguiente comando, realizamos la firma del .jar utilizando JarSigner.



Escribimos lo siguiente para verificar si se ha verificado, y podemos comprobar, que si es así.



ACTIVIDAD 5

En la instalación de aplicaciones software existen sistemas de gestión de paquetes que permiten automatizar el proceso y que variarán en función del sistema operativo o el tipo de fichero de instalación. ¿Qué tipo de componentes será necesario que contengan estos paquetes? ¿Será suficiente con el código de la aplicación?

ACTIVIDAD 6

Para poder crear un asistente de instalación es necesario haber generado un fichero instalador EXAE. Una de las herramientas más utilizadas para la creación de ficheros de este tipo es Launch4j. Esta herramienta requiere del JAR correspondiente al proyecto sobre el que se está trabajando. Desde NetBeans escoge alguno de los proyectos ya desarrollados y crea un paquete JAR.

ACTIVIDAD 7

Una de las herramientas más utilizadas para la creación de asistentes de instalación es Inno Setup Compiler. Tras la creación del instalador .exe necesario para el desarrollo de un asistente de instalación, es necesario completar el proceso de construcción. En este caso, se propone crear un asistente de instalación para alguna de las aplicaciones desarrolladas hasta ahora

# **BIBLIOGRAFÍA**

*Candil Rodríguez, R. G. (2024). Unidad 7: Desarrollo de interfaces. IES Ribera del Tajo.*

*NetBeans, A. (s. f.). Welcome to Apache NetBeans. https://netbeans.apache.org/front/main/index.html*