

# Compilación JSO

## Pasos a seguir para instalar los .jar correctamente

1. Instalar JDK 11
2. Instalar Maven
3. Intalar IDE
4. Instalacion GIT
5. Clonar repositorio del proyecto
6. Importar Jar

### Instalar JDK

Esta guía te proporcionará los pasos necesarios para instalar el JDK 11 (Java Development Kit) en tu sistema.

#### Paso 1: Verificar si Java ya está instalado

Antes de comenzar, verifica si Java ya está instalado en tu sistema ejecutando el siguiente comando en tu terminal o símbolo del sistema:

```
java --version
```

Si Java ya está instalado, deberías ver la versión actual de manera similar a:

```
openjdk 11.0.21 2023-10-17
OpenJDK Runtime Environment OpenLogic-OpenJDK (build 11.0.21+9-adhoc..jdk11u)
OpenJDK 64-Bit Server VM OpenLogic-OpenJDK (build 11.0.21+9-adhoc..jdk11u, mixed mode)
```

Si Java ya está instalado, deberías ver la versión actual. Si no es así, o la version es distinta a la v11 continúa con los pasos a continuación.

#### Paso 2: Descargar el JDK 11

1. Accede al sitio web oficial de Oracle para descargar el JDK 11: Sitio de Descargas de Oracle.
2. Acepta los términos y condiciones y elige la versión del JDK 11 que sea adecuada para tu sistema operativo. Puedes encontrar versiones para Windows, macOS y Linux.
3. Haz clic en el enlace de descarga y espera a que se complete la descarga del instalador.

#### Paso 3: Instalar el JDK 11

## Windows

1. Ejecuta el instalador descargado.
2. Sigue las instrucciones del asistente de instalación.
3. Configura las variables de entorno:
  - Haz clic con el botón derecho en “Este PC” o “Mi PC” en tu escritorio o menú de inicio y selecciona “Propiedades”.
  - Haz clic en “Configuración avanzada del sistema” en el panel izquierdo.
  - En la pestaña “Opciones avanzadas”, haz clic en el botón “Variables de entorno”.
  - En la sección “Variables del sistema”, haz clic en “Nuevo”.
  - Ingresa `JAVA_HOME` como nombre de la variable y la ruta de instalación del JDK como valor (por ejemplo, `C:\Program Files\Java\jdk-11.0.1`).
  - Busca la variable `Path` en la sección “Variables del sistema” y haz clic en “Editar”.
  - Haz clic en “Nuevo” y agrega `%JAVA_HOME%\bin` al final.
  - Haz clic en “Aceptar” para cerrar todas las ventanas.
4. Abre una nueva ventana de terminal (Símbolo del sistema o PowerShell) y verifica la instalación ejecutando:

```
java -version
```

## macOS

1. Abre el archivo DMG descargado.
2. Arrastra el icono del JDK a la carpeta de Aplicaciones.
3. Abre una terminal y verifica la instalación con:

```
java --version
```

## Linux

1. Extrae el archivo `tar.gz` descargado en un directorio de tu elección.

```
tar -xvf nombre_del_archivo.tar.gz
```

2. Configura las variables de entorno. Abre el archivo `~/.bashrc` o `~/.zshrc` con tu editor de texto favorito y agrega las siguientes líneas al final del archivo:

```
export JAVA_HOME=/ruta/del/jdk
export PATH=$PATH:$JAVA_HOME/bin
```

3. Guarda y cierra el archivo.
4. Recarga la configuración del shell:

```
source ~/.bashrc    # o source ~/.zshrc si usas Zsh
```

Verifica la instalación

```
java --version
```

Con estos pasos, has instalado el JDK 11 de Java en tu sistema y configurado las variables de entorno necesarias. Ahora estás listo para desarrollar y ejecutar aplicaciones Java en tu entorno de desarrollo.

## Instalación de Apache Maven

### Paso 1: Descargar Apache Maven

1. Accede al sitio web oficial de Apache Maven para descargar la última versión: Sitio de Descargas de Apache Maven.
2. Descarga la versión binaria (Binary zip archive) del paquete en formato ZIP.

### Paso 2: Descomprimir el Archivo ZIP

1. Encuentra el archivo ZIP descargado y descomprímelo en un directorio de tu elección.
2. Este directorio será la ubicación de tu instalación de Apache Maven.

### Paso 3: Configurar Variables de Entorno

## Windows

1. Configura la variable de entorno `MAVEN_HOME`:
  - Haz clic con el botón derecho en “Este PC” o “Mi PC” en tu escritorio o menú de inicio y selecciona “Propiedades”.
  - Haz clic en “Configuración avanzada del sistema” en el panel izquierdo.
  - En la pestaña “Opciones avanzadas”, haz clic en el botón “Variables de entorno”.
  - En la sección “Variables del sistema”, haz clic en “Nuevo”.
  - Ingresa `MAVEN_HOME` como nombre de la variable y la ruta de instalación de Maven como valor (por ejemplo, `C:\ruta\hacia\apache-maven-3.8.5`).
  - Haz clic en “Aceptar” para cerrar todas las ventanas.
2. Agrega `%MAVEN_HOME%\bin` al Path:
  - Busca la variable `Path` en la sección “Variables del sistema” y haz clic en “Editar”.
  - Haz clic en “Nuevo” y agrega `%MAVEN_HOME%\bin` al final.
  - Haz clic en “Aceptar” para cerrar todas las ventanas.

3. Abre una nueva ventana de terminal (Símbolo del sistema o PowerShell) y verifica la instalación ejecutando:

```
mvn --version
```

### macOS y Linux

1. Abre tu terminal y edita el archivo `~/.bashrc` o `~/.zshrc` con tu editor de texto favorito:

```
export MAVEN_HOME=/ruta/del/apache-maven-3.8.5
export PATH=$PATH:$MAVEN_HOME/bin
```

2. Guarda y cierra el archivo.
3. Recarga la configuración del shell:

```
source ~/.bashrc    # o source ~/.zshrc si usas Zsh
```

4. Verifica la instalación:

```
source ~/.bashrc    # o source ~/.zshrc si usas Zsh
```

### Elección de Entornos de Desarrollo (IDE) para Java: IntelliJ IDEA vs. NetBeans

En el mundo del desarrollo Java, la elección del entorno de desarrollo adecuado desempeña un papel crucial en la eficiencia y productividad del programador. Dos de las opciones más populares son IntelliJ IDEA y NetBeans. En esta sección, exploraremos cómo instalar ambos entornos, destacando sus características clave y brindando una visión general de por qué un IDE (Entorno de Desarrollo Integrado) es esencial para el desarrollo de aplicaciones Java.

#### ¿Por qué utilizar un IDE para Java?

Un IDE ofrece un conjunto de herramientas integradas que simplifican el proceso de desarrollo y aumentan la productividad del desarrollador. Proporciona funciones como resaltado de sintaxis, completado automático de código, depuración, pruebas integradas y gestión de dependencias. Estas características hacen que la escritura, prueba y depuración de código Java sea más eficiente y menos propensa a errores.

Esta guía te proporcionará los pasos necesarios para descargar e instalar IntelliJ IDEA Community Edition en tu sistema.

### IntelliJ IDEA y NetBeans: Dos Gigantes en el Mundo Java

Ambos IntelliJ IDEA y NetBeans son IDEs robustos, pero cada uno tiene sus características distintivas. A continuación, se presenta una tabla comparativa para ayudarte a tomar una decisión informada:

| <i>Características</i>                                  | <i>IntelliJ IDEA</i>                              | <i>NetBeans</i>  |
|---|---|--|
| <i>Licencia</i>   | Comercial (con una versión gratuita: Community)   | Apache License (completamente de código abierto)             |
| <i>Rendimiento</i>                                      | Rápido y eficiente                                | Buena velocidad, pero puede ser más pesado en algunos casos  |
| <i>Soporte de Lenguajes</i>                             | Java, Kotlin, Groovy, Scala y más                 | Principalmente Java, pero tiene soporte para otros lenguajes |
| <i>Plugins y Ecosistema</i>                             | Amplio y activo                                   | Menos extenso en comparación con IntelliJ IDEA               |
| <i>Herramientas de Refactorización</i>                  | Potente conjunto de herramientas                  | Sólido conjunto de refactorización de código                 |
| <i>Soporte de Spring y Java EE</i>                      | Excelente soporte                                 | Buen soporte, pero no tan extenso como IntelliJ IDEA         |
| <i>Integración con Sistemas de Control de Versiones</i> | Git, SVN, Mercurial, y más                        | Soporte sólido para Git, SVN y otros                         |
| <i>Aspectos de Comunidad</i>                            | Activa y próspera                                 | Comunidad activa, pero menor que IntelliJ IDEA               |
| <i>Costo</i>  | Versión Ultimate: Comercial / Community: Gratuita | Totalmente gratuito  |

Esta tabla proporciona solo un resumen de las características clave. La elección entre IntelliJ IDEA y NetBeans dependerá de tus preferencias personales y requisitos específicos del proyecto.

A continuación, exploraremos cómo instalar ambos IDEs para que puedas experimentarlos por ti mismo.

## Instalacion IntelliJ IDEA

### Paso 1: Descargar IntelliJ IDEA Community Edition

1. Accede al sitio web oficial de JetBrains para descargar IntelliJ IDEA Community Edition: Sitio de Descargas de IntelliJ IDEA.
2. Selecciona la versión “Community” y elige tu sistema operativo (Windows, macOS, o Linux).
3. Descarga el instalador.

## **Paso 2: Instalar IntelliJ IDEA Community Edition**

### **Windows**

1. Ejecuta el instalador descargado.
2. Sigue las instrucciones del asistente de instalación.
3. Asegúrate de seleccionar la opción “Create associations” si deseas que IntelliJ IDEA se asocie con los tipos de archivos relevantes.
4. Selecciona la ubicación de instalación y elige las configuraciones adicionales según tus preferencias.
5. Haz clic en “Next” y luego en “Install” para comenzar la instalación.
6. Espera a que se complete la instalación y luego haz clic en “Finish”.

### **macOS**

1. Abre el archivo DMG descargado.
2. Arrastra el icono de IntelliJ IDEA a la carpeta de Aplicaciones.
3. Abre IntelliJ IDEA desde la carpeta de Aplicaciones.

### **Linux**

Extrae el archivo tar.gz descargado en un directorio de tu elección.

`tar -xvf ideaIC-<version>.tar.gz #####` Configurar IntelliJ IDEA

1. Abre IntelliJ IDEA después de la instalación.
2. Si es la primera vez que lo abres, se te pedirá que configure el JDK. Haz clic en “Configure” y selecciona la ubicación del JDK en tu sistema. Si aún no has instalado el JDK, puedes descargarlo desde el sitio oficial de Oracle o usar un JDK compatible con OpenJDK.
3. Haz clic en “OK” para aplicar la configuración.

## **Instalación NetBeans**

### **Paso 1: Descargar NetBeans**

1. Accede al sitio web oficial de NetBeans para descargar la última versión: Sitio de Descargas de NetBeans.

2. Selecciona la versión que desees descargar. Puedes elegir la versión de Apache NetBeans que incluye el conjunto completo de características.
3. Descarga el instalador correspondiente a tu sistema operativo (Windows, macOS o Linux).

## **Paso 2: Instalar NetBeans**

### **Windows**

1. Ejecuta el instalador descargado.
2. Sigue las instrucciones del asistente de instalación.
3. Durante la instalación, puedes seleccionar los paquetes y las opciones que desees instalar. Puedes elegir instalar las herramientas de desarrollo para Java y otras tecnologías compatibles.
4. Continúa con la instalación y espera a que se complete.

### **Linux**

1. Después de descargar el archivo, abre una terminal y navega al directorio donde se encuentra el archivo.
2. Cambia los permisos del archivo para hacerlo ejecutable:

```
chmod +x nombre_del_archivo.sh
```

3. Ejecuta el instalador:

```
./nombre_del_archivo.sh
```

## **Paso 3: Configurar NetBeans**

1. Abre NetBeans.
2. Configura el directorio del JDK:
3. Ve a “Tools” > “Options”.
4. En la categoría “Java”, selecciona la pestaña “Java” y proporciona la ruta del JDK instalado en tu sistema.
5. NetBeans estará listo para su uso.

## **instalar Git**

### **Windows**

1. Accede al sitio web oficial de Git para Windows: Descargar Git para Windows.
2. Descarga el instalador ejecutable (.exe).
3. Sigue las instrucciones del asistente de instalación.

4. Selecciona las opciones de configuración según tus preferencias. Asegúrate de seleccionar la opción “Use Git from Git Bash only” si deseas utilizar la terminal Git Bash.
5. Haz clic en “Next” y luego en “Install” para comenzar la instalación
6. Espera a que se complete la instalación y luego haz clic en “Finish”.

### **Clonar repositorio hadaJSO**

1. Abre una terminal (Git Bash en Windows) en el directorio donde deseas clonar el repositorio.
2. Ejecuta el siguiente comando para clonar el repositorio:

```
git clone https://github.com/tomasmserra/hadaJso.git
```

3. Cambia al directorio del repositorio recién clonado:

```
cd hadaJso
```

### **Actualizar repositorio local para que coincida con el remoto de la rama origin develop**

Asegúrate de estar en la rama correcta. Por ejemplo, si deseas actualizar la rama develop, ejecuta:

```
git checkout develop
```

Ahora, ejecuta el comando git pull para actualizar el repositorio local con los cambios más recientes desde el repositorio remoto:

```
git pull origin develop
```

Git intentará fusionar automáticamente los cambios remotos en tu rama local.

Si hay conflictos de fusión, Git te notificará y deberás resolver los conflictos manualmente

### **Importar Jar**

Para poder compilar el proyecto es necesario importar diferentes jars que contienen dependencias necesarias para el funcionamiento del SMP.

### **Pasos**

1. Abrir terminal de comandos dentro del directorio Jars incluido en los directorios del proyecto y ejecutar los siguientes comandos:
2. Instalar “sdk-20.00.jar”: