

Actividad A3.1: Creación de un proyecto "Hola Mundo" y control de versiones con Git desde IntelliJ IDEA

1. Introducción

En esta práctica, aprenderás a:

- Instalar el Java SDK y configurar el entorno de desarrollo.
- Compilar y ejecutar un programa básico desde la línea de comandos.
- Instalar IntelliJ IDEA y configurar un proyecto Java.
- Crear y ejecutar un programa "Hola Mundo" en IntelliJ IDEA.
- Utilizar el sistema de control de versiones Git dentro de IntelliJ para versionar su proyecto.

2. Instalación de Java SDK

Para desarrollar código en Java, necesitamos instalar el Java Development Kit (JDK). A continuación se detallan los pasos para instalar la versión más reciente.

Instrucciones de instalación para Java SDK:

1. Descargar el JDK:

- Accede a la [página oficial de descargas de Oracle](#) o bien a [OpenJDK](#).
- Elige la versión LTS (Long Term Support) más reciente (por ejemplo, Java 21 o superior) y descarga el instalador adecuado para tu sistema operativo (Windows, MacOS o Linux).

2. Instalación del JDK en Windows:

- Ejecuta el archivo descargado (.exe) y sigue los pasos de instalación. Asegúrate de recordar la ruta de instalación (por defecto suele ser `C:\Program Files\Java\jdk-xx`).
- Una vez instalado, ve a "Panel de Control" > "Sistema y Seguridad" > "Sistema" > "Configuración avanzada del sistema" > "Variables de entorno".
- En "Variables de entorno", busca la variable `Path`, selecciona **Editar** y añade la ruta a la carpeta `bin` del JDK, por ejemplo: `C:\Program Files\Java\jdk-xx\bin`.

3. Verificación de la instalación:

- Abre una terminal (CMD en Windows, o terminal en Mac/Linux) y escribe el comando:

```
java -version
```

- Deberías ver la versión instalada de Java.

3. Compilación y ejecución de "Hola Mundo" desde la línea de comandos

Crear el programa "Hola Mundo":

En la terminal, navega hasta el directorio donde deseas crear el archivo, y crea una carpeta para tu proyecto:

```
mkdir HolaMundo  
cd HolaMundo
```

Crea un archivo de código fuente de Java llamado `HolaMundo.java` usando un editor de texto (como Notepad en Windows, nano en Linux, o cualquier editor de tu preferencia):

```
public class HolaMundo {  
    public static void main(String[] args) {  
        System.out.println("¡Hola, mundo!");  
    }  
}
```

Compilar y ejecutar el programa:

Compilación: En la terminal, asegúrate de estar en el directorio donde guardaste el archivo `HolaMundo.java`, y ejecuta:

```
javac HolaMundo.java
```

Comprueba qué sucede en el directorio donde se encuentra el archivo `HolaMundo.java`

Ejecución: Una vez compilado, ejecuta el programa con el siguiente comando:

```
java HolaMundo
```

3. Si todo está correcto, deberías ver en pantalla: `¡Hola, mundo!`

4. Instalación de IntelliJ IDEA Community Edition

IntelliJ IDEA es un entorno de desarrollo integrado (IDE) ampliamente utilizado para programar en Java.

1. Descargar IntelliJ IDEA:

- Dirígete a la [página oficial de JetBrains](#) y descarga la versión Community (gratuita).

2. Instalación de IntelliJ IDEA en Windows:

- Ejecuta el instalador descargado y sigue las instrucciones.
- Durante la instalación, selecciona la opción para crear un acceso directo en el escritorio y asociar archivos `.java` con IntelliJ.
- Cuando finalice la instalación, abre IntelliJ IDEA.

3. Primer inicio de IntelliJ:

- Al abrir IntelliJ por primera vez, puede que te pregunte si quieres importar configuraciones previas. Elige la opción "Do not import settings" si es la primera vez que lo usas.
- Puedes omitir la instalación de plugins adicionales en este momento.

5. Creación y ejecución del programa "Hola Mundo" en IntelliJ IDEA

1. Crear un nuevo proyecto en IntelliJ IDEA:

- Abre IntelliJ y selecciona **New Project**.
- En **Project SDK**, selecciona la versión de Java que instalaste. Si no aparece automáticamente, haz clic en **Add JDK** y selecciona la ruta de instalación.
- Asegúrate de que el proyecto está configurado para **Java** y haz clic en **Next** hasta completar la configuración.

2. Crear una clase Java para "Hola Mundo":

- En la estructura de tu proyecto, haz clic derecho en `src` y selecciona **New > Java Class**.
- Nombra la clase `HolaMundo`.

En el editor de código, escribe el siguiente código:

```
public class HolaMundo {  
    public static void main(String[] args) {  
        System.out.println("¡Hola, mundo desde IntelliJ IDEA!");  
    }  
}
```

3. Ejecutar el programa:

- Haz clic en el botón verde de reproducción (**Run**) en la esquina superior derecha, o haz clic derecho en la clase **HolaMundo** y selecciona **Run 'HolaMundo.main()'**.
- En la consola, deberías ver la salida: `¡Hola, mundo desde IntelliJ IDEA!`

6. Uso de Git para versionar "Hola Mundo" desde IntelliJ IDEA

IntelliJ IDEA integra Git para facilitar el control de versiones directamente desde el IDE. A continuación se detallan los pasos para inicializar un repositorio y realizar los primeros commits.

1. Inicializar Git en el proyecto:

- En IntelliJ, ve a **VCS > Enable Version Control Integration**.
- Selecciona **Git** como sistema de control de versiones y haz clic en **OK**. Ahora el proyecto tiene un repositorio Git inicializado.

2. Realizar el primer commit:

- IntelliJ detectará automáticamente los archivos nuevos (por ejemplo, `HolaMundo.java`).
- Ve a **VCS > Commit** o usa el atajo de teclado (Ctrl+K en Windows/Linux o Cmd+K en Mac).
- Añade un mensaje de commit, como `"Inicializado el proyecto con la clase HolaMundo"`, y haz clic en **Commit**.
- Si aparece un aviso de revisión de código, puedes hacer clic en **Commit** nuevamente para confirmar.

3. Crear una nueva rama y realizar cambios adicionales:

- Ve a **Git > Branches** y selecciona **New Branch**.
- Nombra la nueva rama `mejora-hola-mundo`.
- Realiza cambios en `HolaMundo.java` (por ejemplo, añade un segundo `System.out.println("Bienvenidos al curso de DAM")`).
- Haz un commit en la rama `mejora-hola-mundo` con el mensaje `"Añadido mensaje de bienvenida"`.

4. Fusionar la rama con la rama principal (main):

- Cambia a la rama principal desde **Git > Branches > main**.
- Ve a **Git > Merge** y selecciona la rama `mejora-hola-mundo` para fusionarla en `main`.
- Si no hay conflictos, se completará la fusión. Haz un commit de la fusión si IntelliJ lo solicita.

5. Verificar el historial de commits:

- Ve a **Git > Show Git Log** para ver el historial de commits, incluyendo los cambios realizados y la fusión de la rama.

6. Crear repositorio en Github y subir cambios

- Ve a **Git > Github > Share Project On Github**, para crear el nuevo repositorio en Github desde el IDE. Tendrás que añadir y autorizar tu cuenta, además de darle un nombre al repositorio (por defecto te pondrá el nombre del proyecto Java) y el nombre de remote (por defecto, origin). Una vez configurado, pulsa Share, y se hará automáticamente un push del proyecto. Comprueba en tu github que el proyecto ha sido correctamente sincronizado.

7. Instrucciones de entrega

Una vez realizado todos los pasos indicados en la práctica, debes crear una memoria, y entregarla en formato pdf con el nombre

apellidos_nombre_ED03_1.pdf

Con capturas indicando la realización de todas las actividades. Además, debes contestar a las siguientes preguntas:

- ¿Crees que el proceso de compilación y ejecución de un programa en Java es más sencillo a través del IDE que a través de la consola de comandos?
- ¿Crees que el uso de git desde el IDE es más sencillo que desde la consola de comandos?
- Explora todas las opciones que hay en el menú Git de IntelliJ Idea e indica si hay alguna que te llame la atención o que no conozcas.

Incluye también un enlace al repositorio en Github creado desde IntelliJ Idea.