

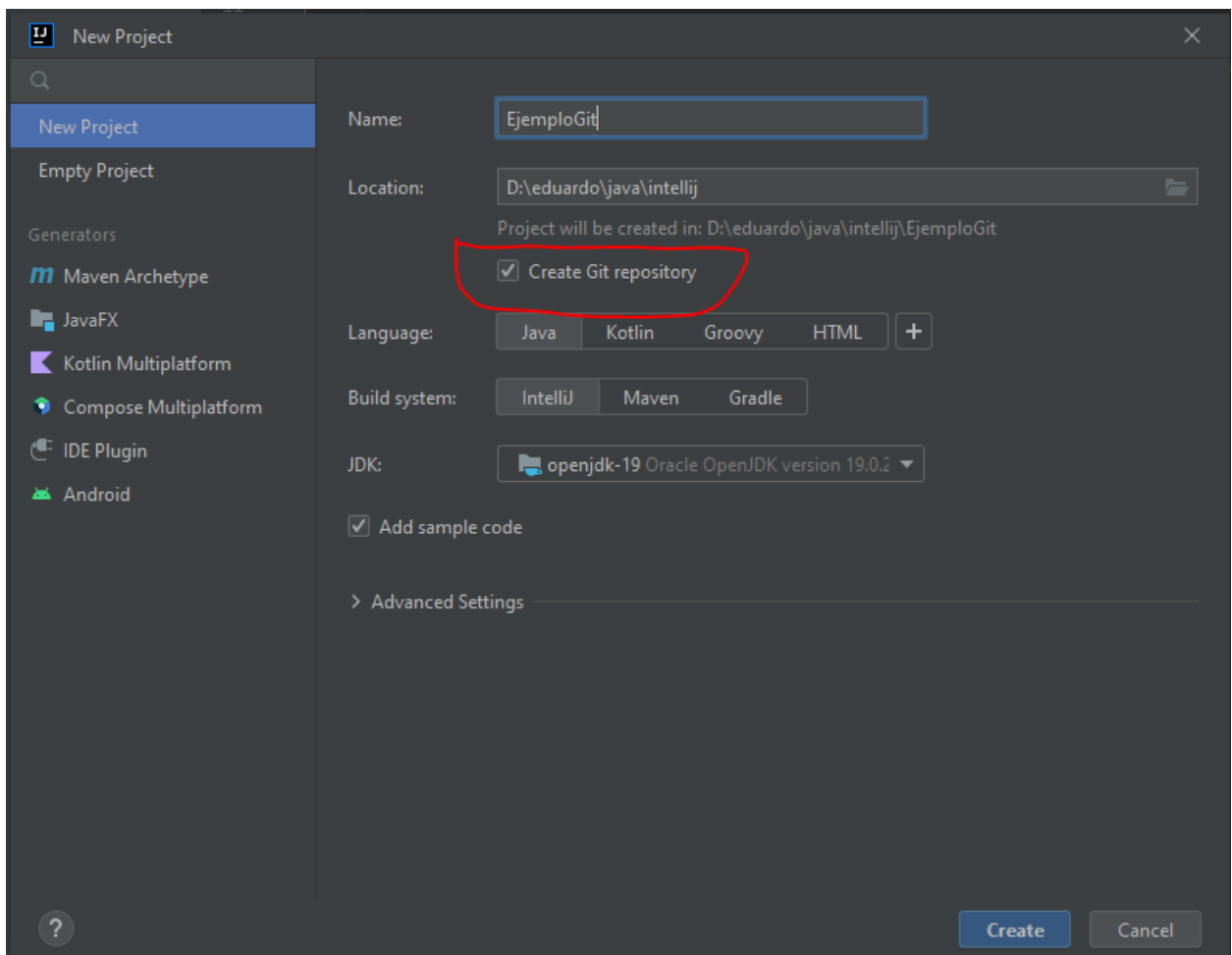
## Actividad 1 – Preparación del git

---

SINTAXIS (terminal): \$ git comando - -opciones

Órdenes:

- Configura Git con un nombre y un correo electrónico.
  - git config --global [user.name | user.mail]
- Comprueba la configuración:
  - git config --list
- Descarga el archivo java\_ejemplo.zip , descomprímelo y crea un proyecto java: EjemploGit



- Si sale la opción de “crear repositorio”, selecciónala y si no te sale puedes iniciar el repositorio más adelante situándote dentro de la carpeta del proyecto (EjemploGit) y ejecutando desde el terminal: `$ git init` .
- Visualiza el contenido de la carpeta `.git` (`$ ls -a` para ver carpetas ocultas )
- Comprueba el estado del repositorio (terminal)
  - `$ git status`

En este punto, aún no se ha realizado ningún **commit**, ni siquiera se ha pasado ningún fichero al estado de **stage**. Debería salir algo parecido a esta imagen, {puede variar el estado del Main si habéis escrito algo ..}

```
new file:   Main.java

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   ../.idea/workspace.xml
    modified:   Main.java

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Persona.java

eduardo@eduardo-osi MINGW64 /d/eduardo/java/intellij/EjemploGit/src (master)
$ |
```

- Actualiza el fichero `.gitignore` para Java según especificaciones en <https://github.com/github/gitignore> incorpora las extensiones de los archivos para que git los ignore.
- Pasa todo lo que esté en la carpeta (`src`) a un estado preparado (`stage`) { `git add [archivo | carpeta]` }
- Revisa de nuevo el estado { `git status` }
- Realiza un commit con el mensaje “Versión inicial del proyecto” { `git commit -m “mensaje”` } solo del archivo `Main.java`
- Muestra el log en el terminal.

Crea fichero con las siguientes capturas tras realizar todos los puntos:

1. Realiza una captura del estado del repositorio {`status`}
2. Contenido del fichero `.gitignore`
3. Muestra las líneas del log de Git.

4. Imagen desde el intelliJ del proyecto que sólo muestre los archivos: Main.java y persona.java

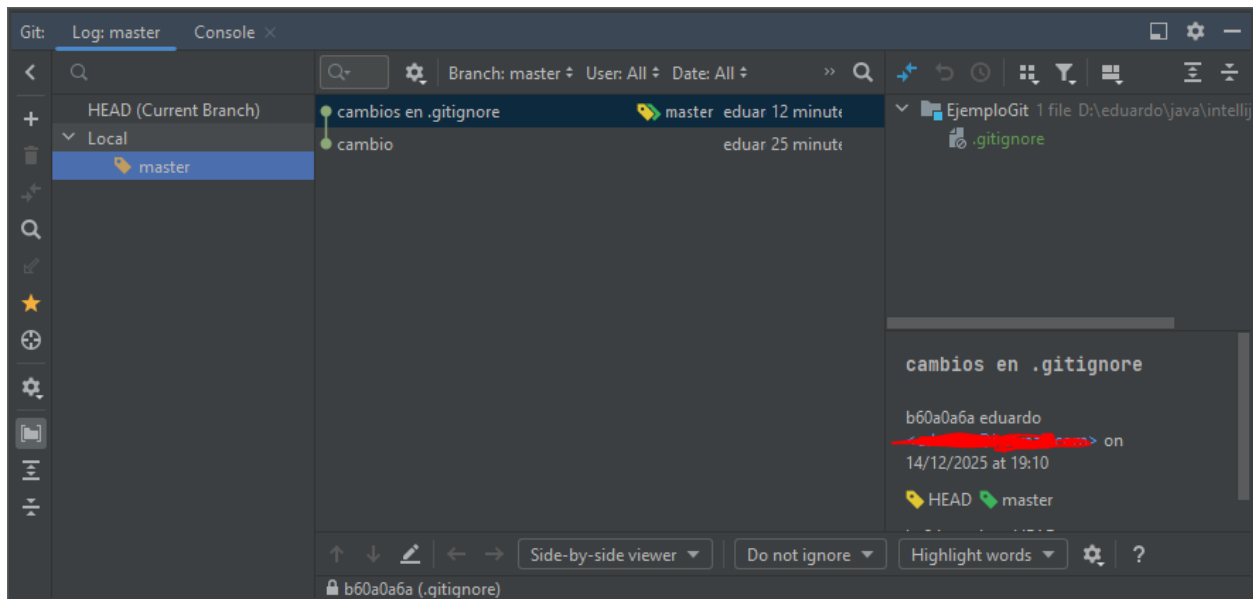
- Situación Main.java (commit); Persona.java (stage)
- Pregunta: ¿Están los ficheros impresos con el mismo color o no?

5. ¿En qué estado está .gitignore ? Si estuviera en stage realiza un commit sólo sobre ese archivo, etiquétalo con un mensaje.

Entrega fichero: tunombre\_git1.pdf

Anexo: Interpretación de las siguientes capturas:

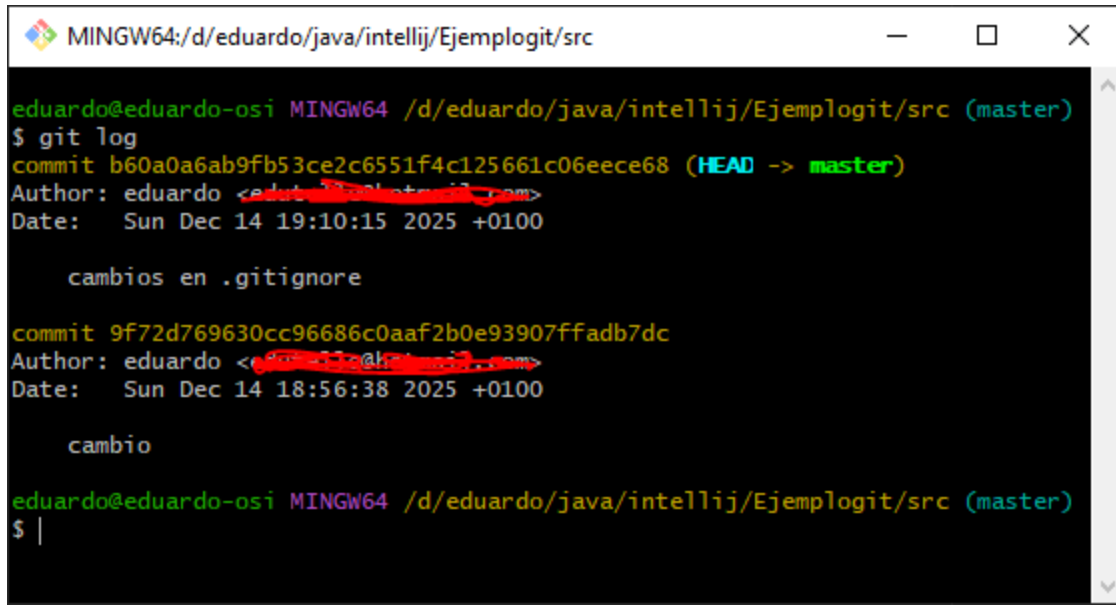
-> Clic en git (intellij):



Se han realizado dos commit:

- Hace doce minutos ha sido modificado el fichero .gitignore
- Hace 25 minutos se ha etiquetado con “cambio” para saberlo desde aquí tendríamos que hacer clic sobre cambio.

\$ git log



```
MINGW64:/d/eduardo/java/intellij/EjemploGit/src
eduardo@eduardo-osi MINGW64 /d/eduardo/java/intellij/EjemploGit/src (master)
$ git log
commit b60a0a6ab9fb53ce2c6551f4c125661c06eece68 (HEAD -> master)
Author: eduardo <eduardo@severo-ochoa.com>
Date: Sun Dec 14 19:10:15 2025 +0100

    cambios en .gitignore

commit 9f72d769630cc96686c0aaf2b0e93907ffadb7dc
Author: eduardo <eduardo@severo-ochoa.com>
Date: Sun Dec 14 18:56:38 2025 +0100

    cambio

eduardo@eduardo-osi MINGW64 /d/eduardo/java/intellij/EjemploGit/src (master)
$ |
```

Vemos 2 cambios igual que antes pero ejecutando el comando.

### Desde el intellij:

Los ficheros confirmados (commit) están en blanco y los pendientes (stage) están en verde.

