



CIPFP Mislata

Centre Integrat Públic
Formació Professional Superior

Preparación del entorno

Autor: Joan Puigcerver Ibáñez

Correo electrónico: j.puigcerveribanez@edu.gva.es

Curso: 2022/2023

Licencia: BY-NC-SA

(Reconocimiento - No Comercial - Compartir Igual)



Objetivos

Configurar el entorno de desarrollo en ordenadores con sistema operativo Windows o distribuciones de Linux basadas en Debian, como Ubuntu o Linux Mint.

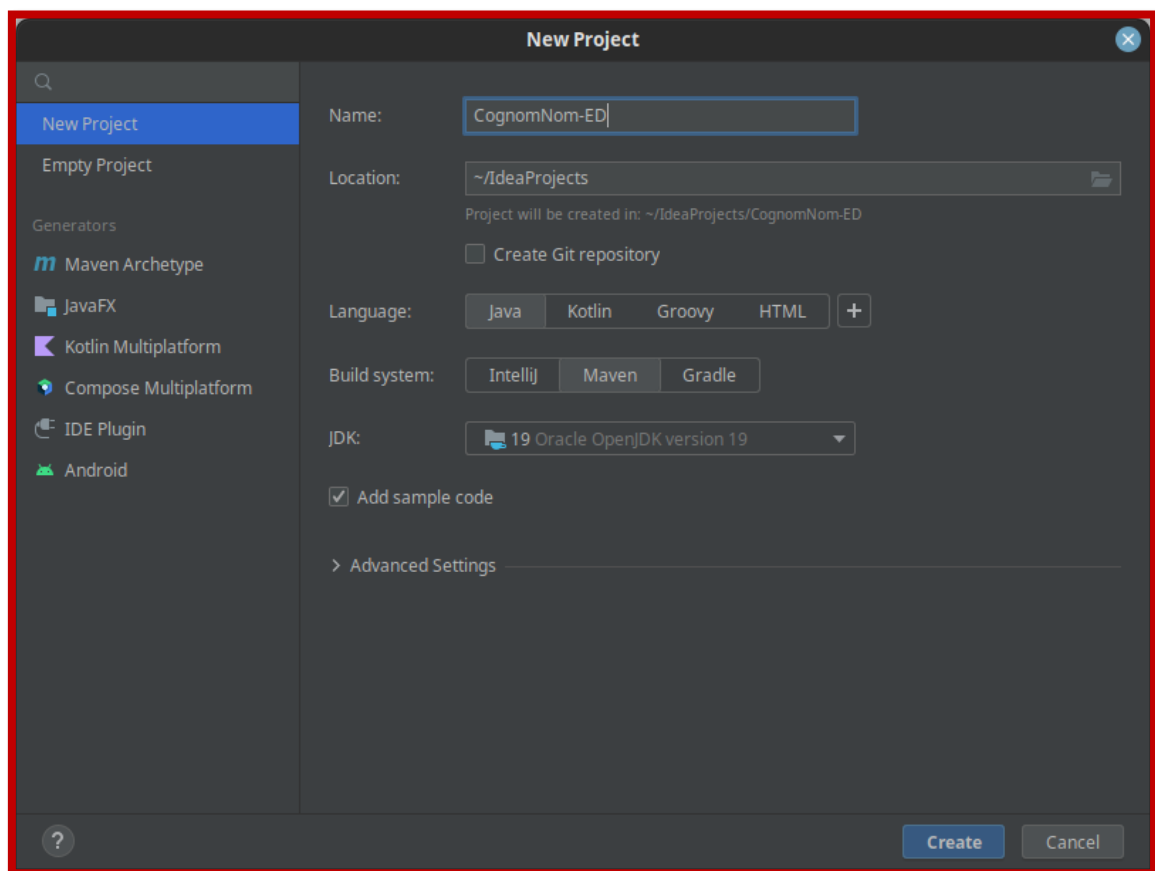
En este manual se instalará y se configurarán las siguientes herramientas:

- [IntelliJ IDEA](#) : Entorno de desarrollo integrado.
- [Git](#) : Enina de control de versiones.
- [GitHub](#) : Almacenamiento online de repositorios Git.

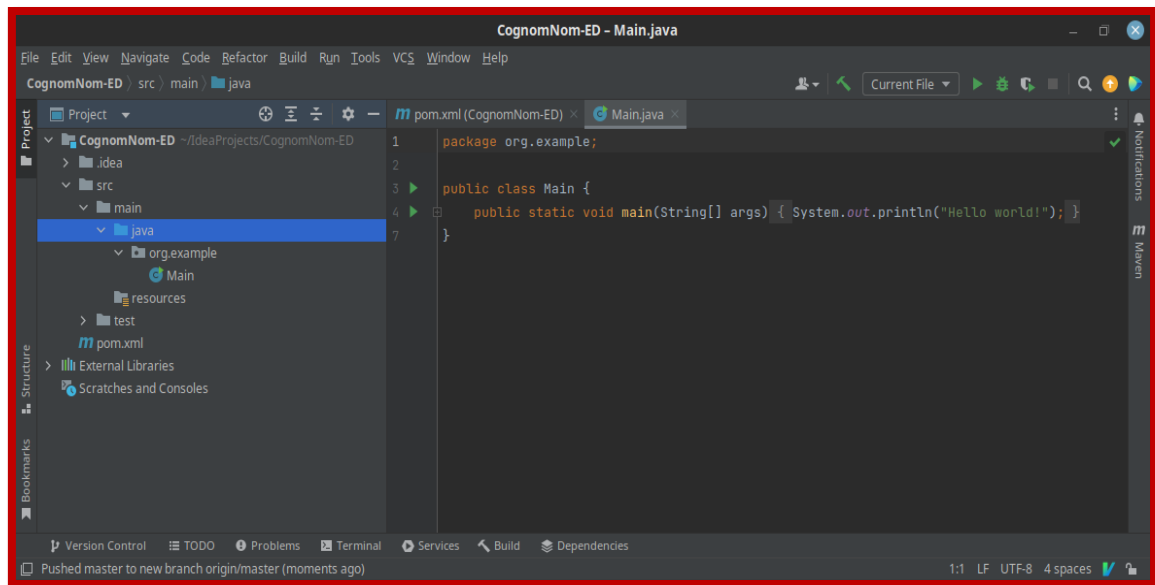
1.IntelliJ IDEA

Descarga e instala [IntelliJ IDEA](https://www.jetbrains.com/idea/download/) mediante el instalador disponible en <https://www.jetbrains.com/idea/download/> . Utilizaremos la versión **Community** del IDE.

- Inicia IntelliJ.
 - Crea un nuevo proyecto.
 - Nombre del proyecto: **ApellidoNombre-ED**
 - Cambiar *Apellido* y *Nombre* por tu primer apellido y tu nombre.
 - Lenguaje: **Java**
 - Build system: **Maven**
 - Este sistema es más complejo que *IntelliJ*, pero este proyecto lo utilizaremos durante todo el curso y estudiaremos este sistema más adelante.
 - Seleccionar **JDK**: Oracle OpenJDK 19.
 - Si no tienes ningún instalado, darle a la opción **Download JDK...**



Una vez creado, tendrá un proyecto vacío creado.

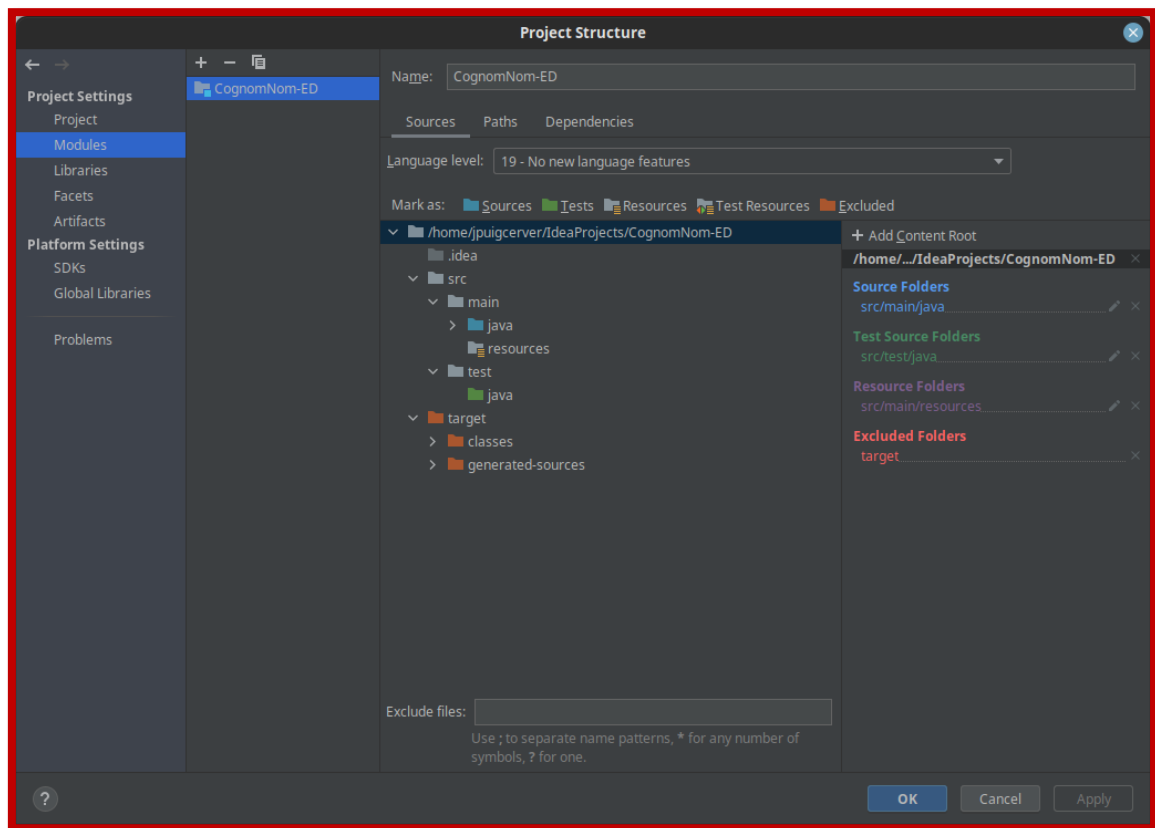


- Comprueba que el programa *Main.java* que viene por defecto se ejecuta. Para ejecutar un programa Java debe darle al botón **Run**.
 - Una vez comprobado, puede eliminarse.

1.1. Estructura de un proyecto

Como puede observar, por defecto se han creado una serie de carpetas en el proyecto.

- El **código fuente**, se guarda en la carpeta marcada como *raíz del código fuente*. Normalmente es **src**. En Maven, se ha creado la carpeta **src/main/java**. En IntelliJ, esta carpeta está marcada en azul.
- Las **pruebas**, se guarda en la carpeta marcada como *raíz del código fuente de pruebas*. Normalmente es **test**. En Maven, se ha creado la carpeta **src/test/java**. En IntelliJ, esta carpeta está marcada en verde.
- El **bytecode (.class)**, se guarda en la carpeta configurada como *destino de salida (output path)*. Normalmente es **out**. En Maven, se ha creado la carpeta **target/classes**. En IntelliJ esta carpeta se puede configurar en **Project Structure > Modules > Path**.

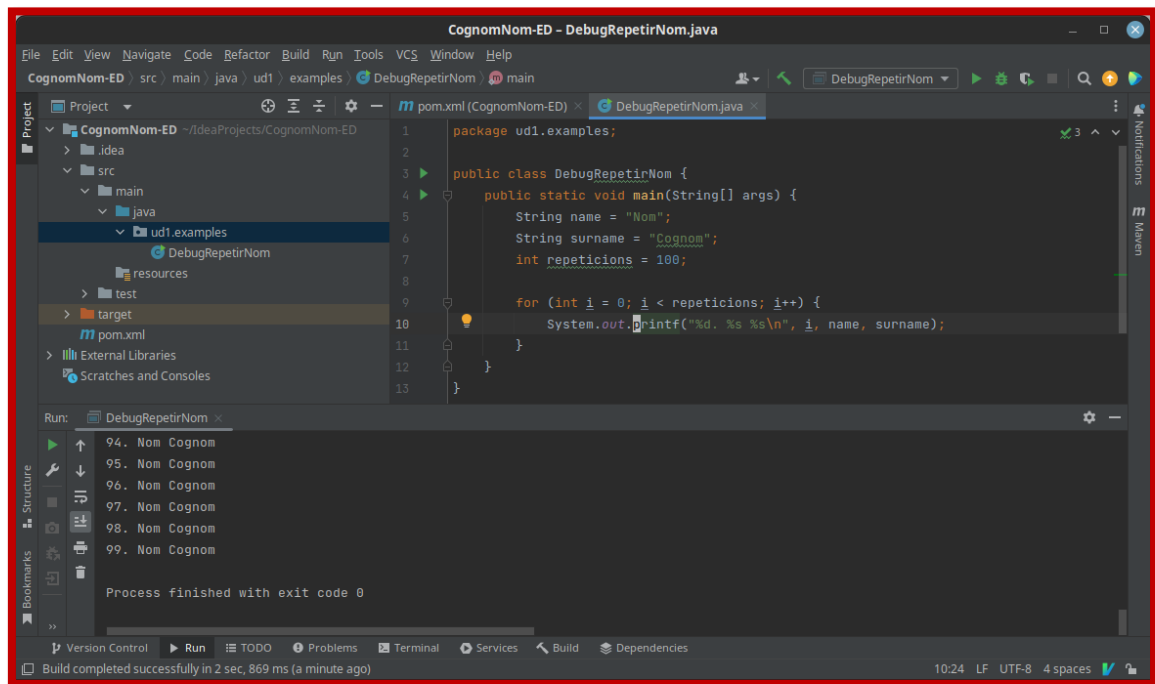


1.2. Código fuente

Java es un lenguaje orientado a objetos y el código fuente se organiza en **Clases**.

En Java, se pueden organizar las distintas *Clases* en carpetas, llamados **paquetes o packages**. Al igual que con las carpetas, un package puede contener más packages dentro. Como separador se utiliza el **punto (.)**.

- Crea el package **ud1.examples**. Botón derecho > New > Package...
- Crea una clase [DebugRepetirNombre](#) dentro de este paquete, copia el contenido del ejemplo y ejecútalo. Comprueba que funciona.



2.Git

Ante todo, hay que instalar [Git](#) .

Es necesario descargar el instalador desde la web: <https://git-scm.com/downloads>

En Ubuntu se puede instalar mediante los pedidos:

```
$ sudo apt-get update
$ sudo apt-get install git
```

3.GitHub


- Si no tienes ninguna, créate una cuenta en [GitHub](#) .
- Proporciona tu usuario al docente del módulo mediante la tarea correspondiente en el curso del módulo.
- Espera a ser invitado a la organización:
 - **DAM1S:** [fpmislata-dam1s-ed](#) .
 - **DAW1:** [fpmislata-daw1-ed](#) .
- Crea un repositorio dentro de la organización:
 - Propietario: **fpmislata-{grupo}-ed**
 - Nombre del repositorio: **ApellidoNombre-ED**
 - Descripción: Repositorio de las soluciones del módulo ED del grupo {GRUP} en CIPFP Mislata del estudiante Nombre Apellido
 - Visibilidad: **Privado**
 - **NO** añadir **README** ni **LICENSE**.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner *

Repository name *

 fpmislata-dam1s-ed ▾


 /

CognomNom-ED ✓


Great repository names are short and memorable. Need inspiration? How about [animated-funicular](#)?

Description (optional)

Respositori de les solucions del mòdul ED del grup {GRUP} en CIPFP Mislata del estudiant Nom Cognom

☐  Public

Anyone on the internet can see this repository. You choose who can commit.

☒  Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore


Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

Choose a license

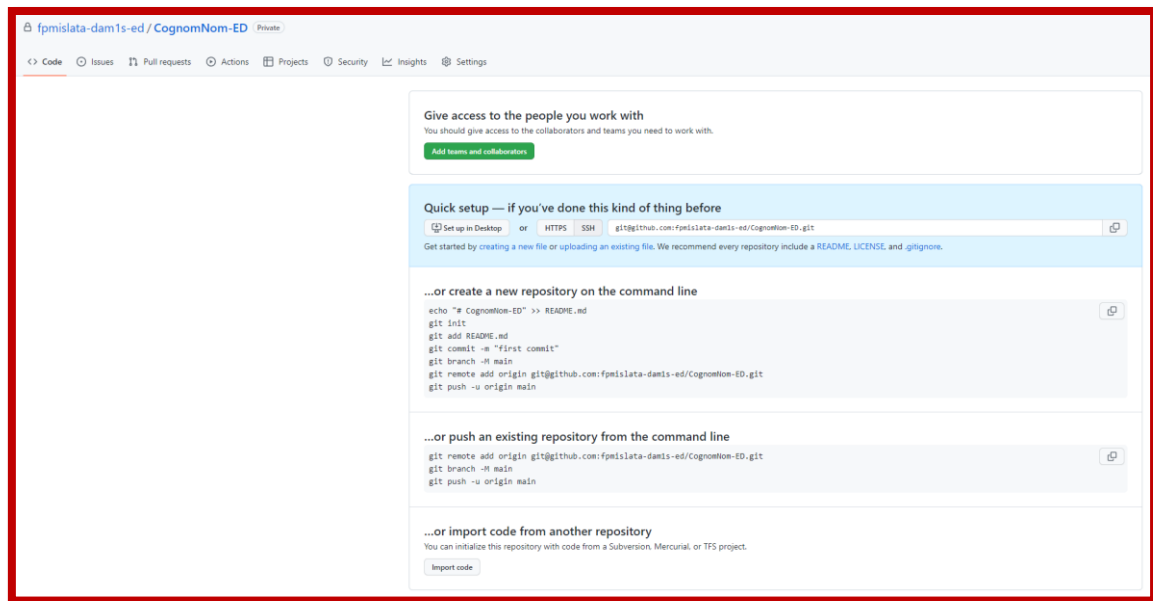
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

 You are creating a private repository in the fpmislata-dam1s-ed organization.

Create repository

Si se ha creado correctamente, debería ver una página como esta:



Proporciona la dirección HTTPS de tu repositorio al docente del módulo mediante la tarea correspondiente en el curso del módulo.

4.Integración IntelliJ + GitHub

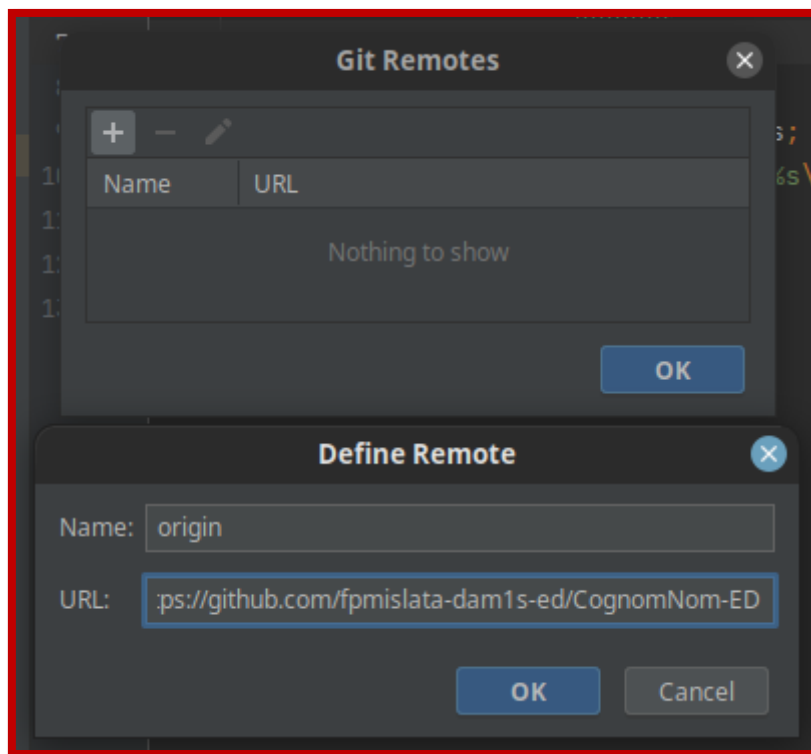
- Copiar el contenido de [.gitignore](#) en un archivo llamado **.gitignore** en la raíz de tu repositorio.

- Añadir las líneas:

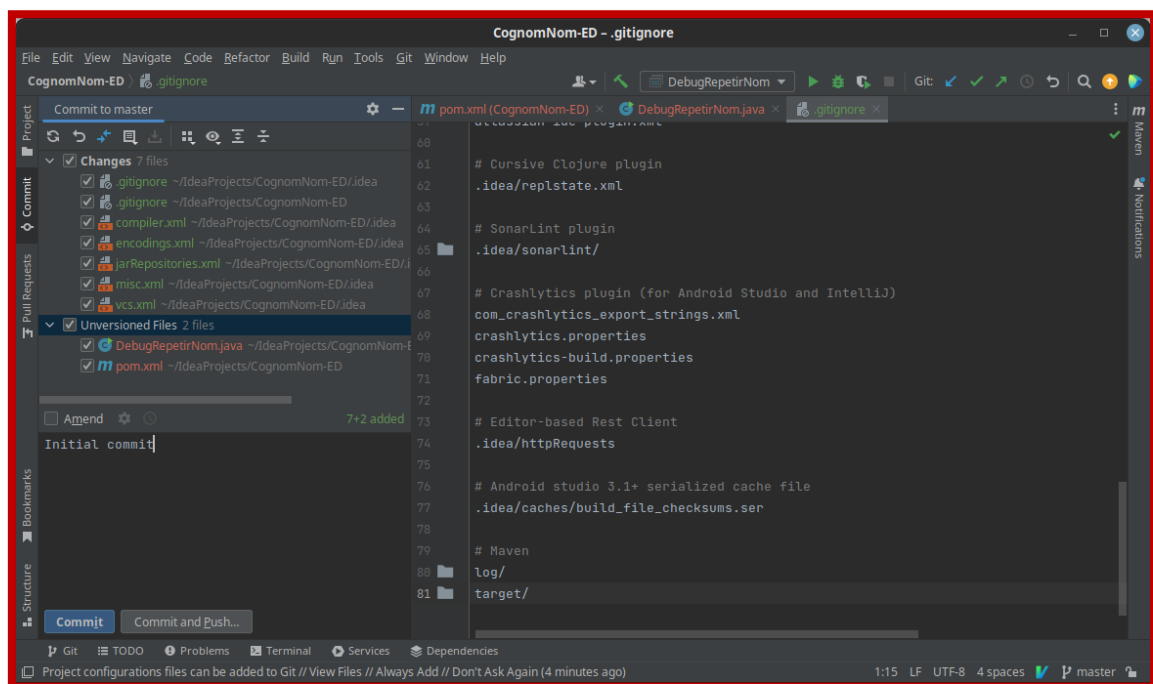
```
# Maven
log/
target/
```

- Crea el archivo **README.md** en la raíz del proyecto con el contenido:
- `# CognomNom-ED`
- Respositori de les solucions del mòdul ED de {GRUP} en CIPFP Mislata del estudiant Nom Cognom
- Activar la versión de control: **VCS > Enable Version Control Integration** y seleccionar **Git**
- Añadir el repositorio de Github: **Git > Manage Remotes...** y darle al +
 - Name: **origin**
 - URL: **El enlace HTTPS de tu proyecto de GitHub**
- Le pedirá iniciar sesión, hágalo mediante Use Token...
 - Dele a **Generate...**
- Se le abrirá una página de GitHub para generar el PAT token.
 - Ponga la expiración: No expiration
 - Genere el token

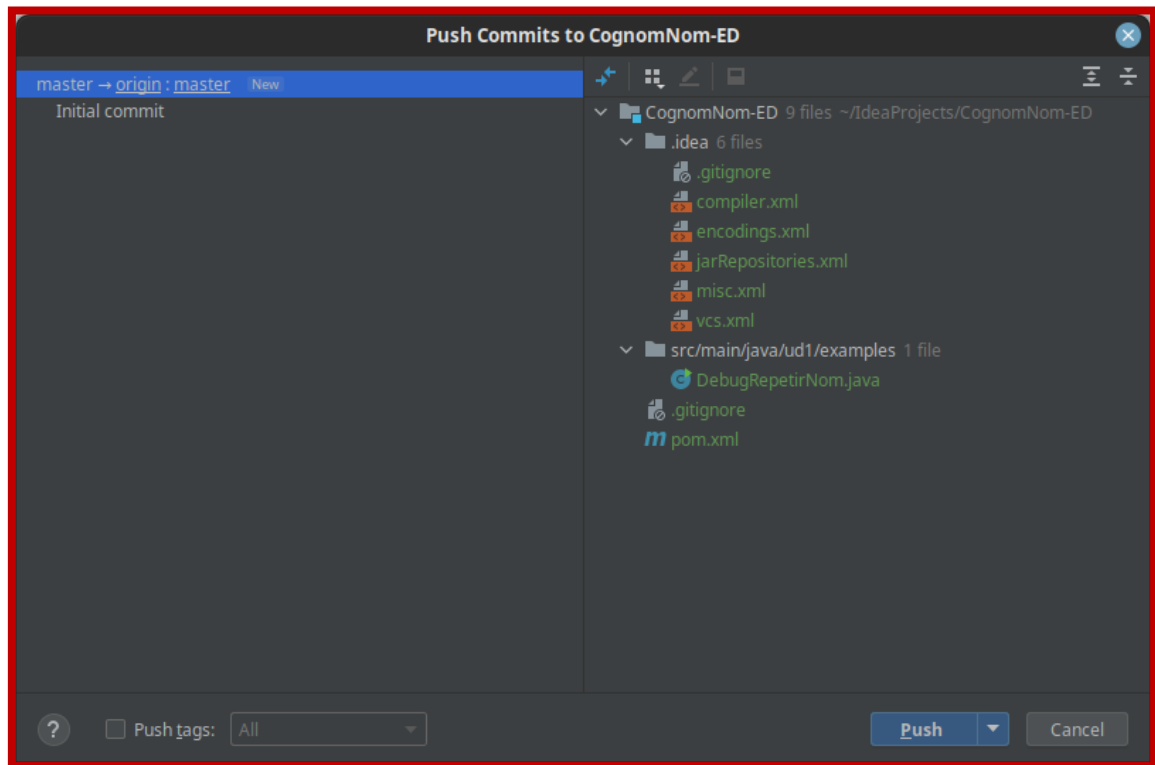
- Copie (y guárdese) el token e introdúzcalo en IntelliJ.



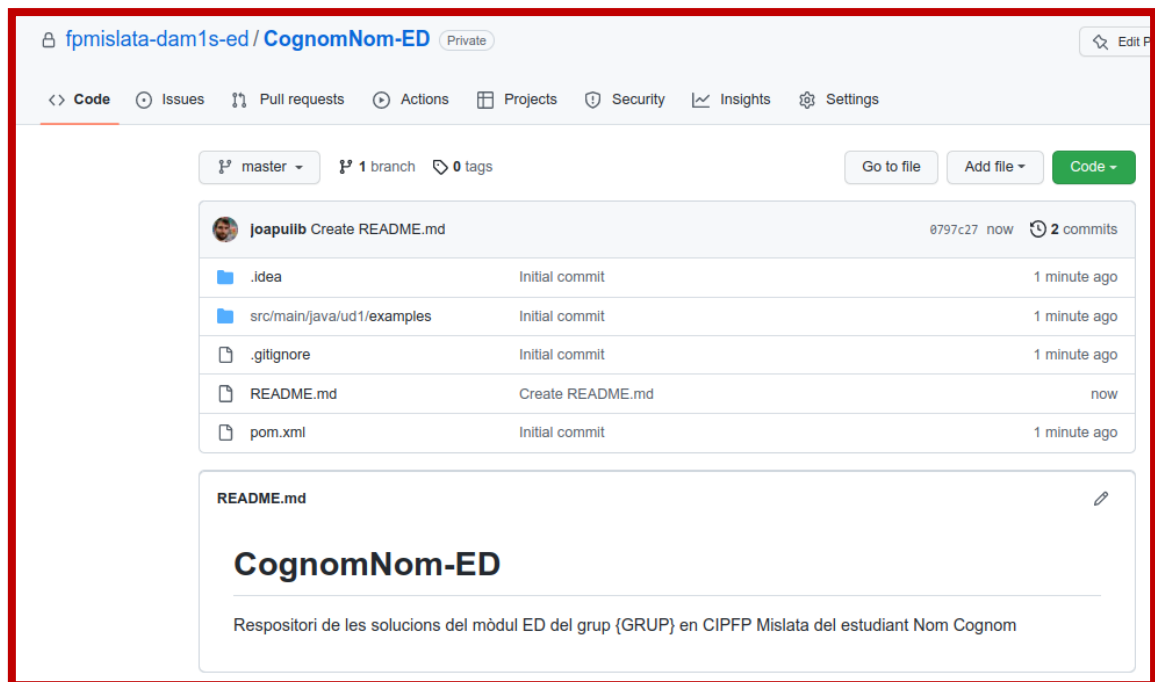
- Realizar un commit inicial: **Git > Commit...**
 - Seleccionar todos los archivos
 - Mensaje: Initial commit
 - Botón: **Commit**



- Subir el proyecto a GitHub: **Git > Push...** Debería salir en verde los archivos que va a subir.



- Verifique que se ha subido a GitHub:



5.Trabajar con Git

Una vez configurado el espacio de trabajo, ya se puede utilizar para añadir nuevas clases y sincronizarlas en distintos espacios de trabajo mediante Git. Todo el código estará dentro de la carpeta **src**.

Cuando se quiera sincronizar el código de IntelliJ IDEA en Github será necesario:

- Realizar un pull para incorporar los cambios que hayan podido haber y no crear conflictos: **Git > Pull...**
- Realizar un commit con los cambios: **Git > Commit...**
 - Mensaje del commit **significativo**
- Realizar un push: **Git > Push**
- Comprobando que se ha actualizado correctamente en GitHub.