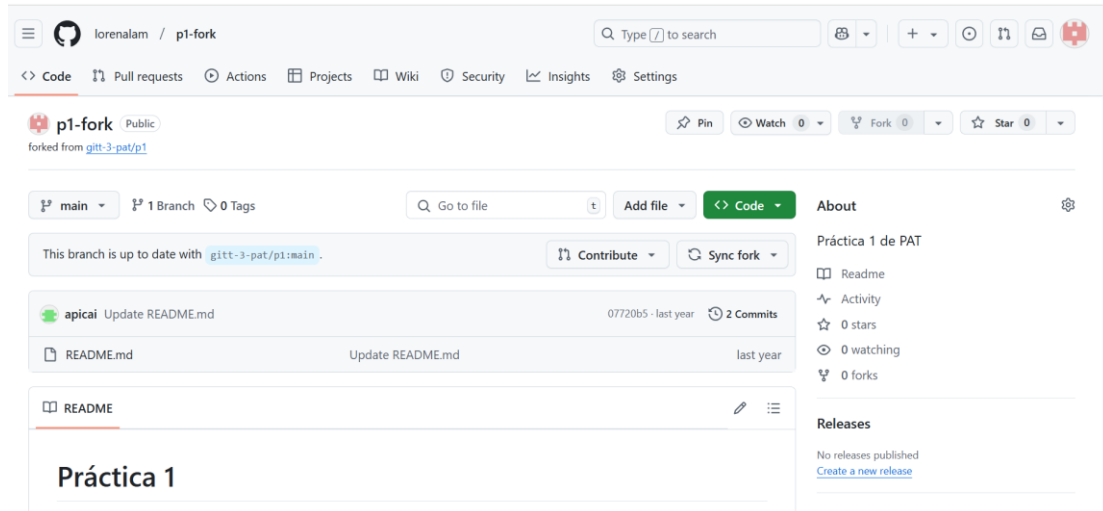


PRÁCTICA 1 PAT

COMANDOS GIT

1. Desde la cuenta de Github hacer un "fork" del repositorio: <https://github.com/gitt-3-pat/p1>



2. Prueba los siguientes comandos sobre el anterior repositorio:

git clone

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  COMENTARIOS

@lorenalam →/workspaces/p1-fork (main) $
@lorenalam →/workspaces/p1-fork (main) $ git clone https://github.com/gitt-3-pat/p1
Cloning into 'p1'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 5 (from 1)
Receiving objects: 100% (6/6), done.
@lorenalam →/workspaces/p1-fork (main) $
```

Sirve para clonar/copiar un repositorio existente en un nuevo directorio, en este caso gitt-3-pat en p1

git status

```
@lorenalam →/workspaces/p1-fork (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    p1/

nothing added to commit but untracked files present (use "git add" to track)
@lorenalam →/workspaces/p1-fork (main) $
```

Muestra el estado del directorio de trabajo, en este caso esta actualizado con el origin/main. También incluye lo que ha ocurrido con los comandos git commit y git add.

Vemos como al crear un archivo nuevo llamado prueba al hacer git status cambia

```
● @lorenalam → /workspaces/p1-fork (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   p1

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

git add

Añade todos los cambios realizados en los archivos de un directorio como por ejemplo nuevos archivos creados, cambios en existentes o archivos eliminados.

El punto (.) indica que queremos incluir todos los archivos y cambios que se han hecho del directorio actual y sus subdirectorios. Básicamente, prepara esos archivos para ser committed.

git commit

```
● @lorenalam → /workspaces/p1-fork (main) $ git add .
● @lorenalam → /workspaces/p1-fork (main) $ git commit -m "modificacion"
[main 93db88c] modificacion
 1 file changed, 1 insertion(+)
 create mode 160000 p1
○ @lorenalam → /workspaces/p1-fork (main) $
```

Con esto guardamos los cambios realizados en el repositorio mediante git add y con el -m damos una descripción del cambio.

git push

```
● @lorenalam → /workspaces/p1-fork (main) $ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 278 bytes | 278.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/lorenalam/p1-fork
 07720b5..93db88c  main -> main
```

Subimos los commits al repositorio remoto

git checkout

```
• @lorenalam → /workspaces/p1-fork (main) $ git checkout -b nuevarama
Switched to a new branch 'nuevarama'
• @lorenalam → /workspaces/p1-fork (nuevarama) $ git checkout main
M      p1
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
• @lorenalam → /workspaces/p1-fork (main) $
```

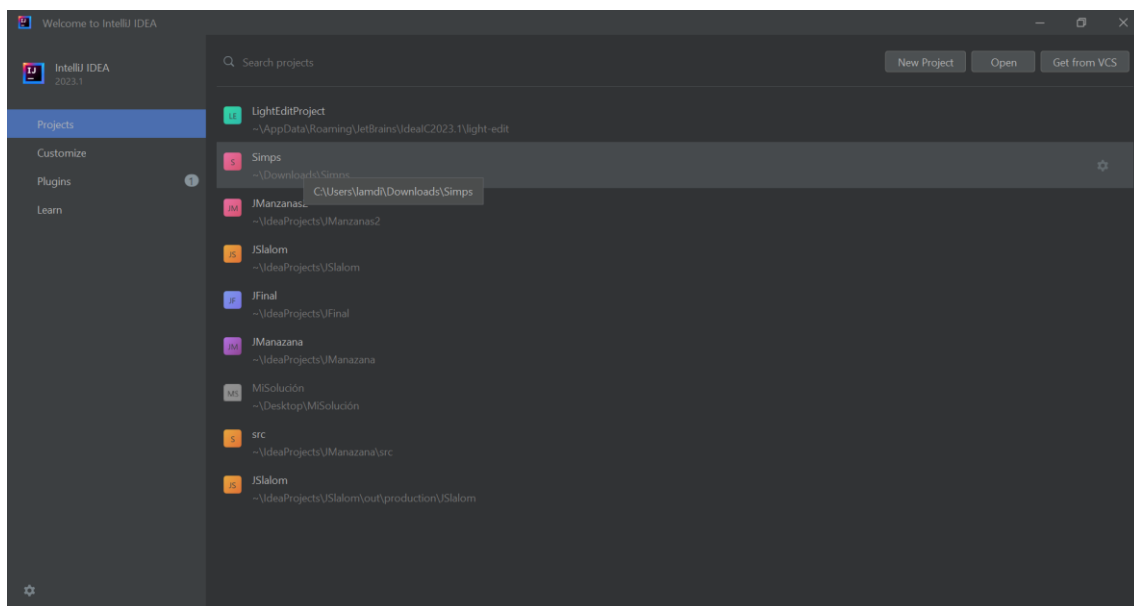
Con este comando nos podemos mover entre ramas o con la variante -b nombrerama crear una nueva rama a la que movernos y empezar a trabajar. En este caso 1 creamos y nos movemos a la rama 'nuevarama' y luego volvemos a la rama original 'main'

INSTALACIÓN ENTORNO DE DESARROLLO JAVA

Ya tenía instalada la versión 19.0.2 en mi ordenador.

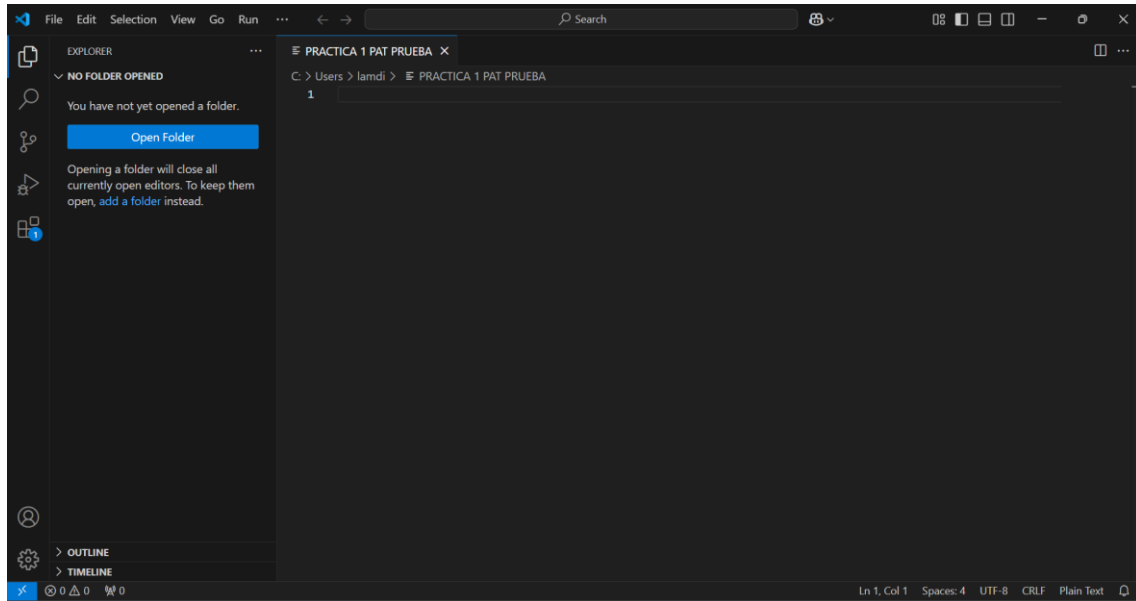
```
PS C:\Users\lamdi> java -version
java version "19.0.2" 2023-01-17
Java(TM) SE Runtime Environment (build 19.0.2+7-44)
Java HotSpot(TM) 64-Bit Server VM (build 19.0.2+7-44, mixed mode, sharing)
PS C:\Users\lamdi> |
```

También tenía instalada la versión 2023 de IntelliJ



Procedo a la instalación de Maven y VsCode.

Prueba instalación VsCode



Prueba instalación Maven