

PRÁCTICA 1

Prerrequisitos:	1
1. Cuenta en GitHub y Token:	1
2. Fork del repositorio:	1
3. Configuración de Gitpod:	2
Entorno de desarrollo:	3
1. Java 17 y Maven:	3
2. IntelliJ:	3
Desarrollo de la práctica:	4
1. git clone:	4
2. git status:	4
3. git add:	5
4. git commit:	5
5. git push:	6
6. git checkout:	7
Materia adicional:	7
1. Crear otra rama:	7
2. MERGE: Unir los cambios de dos ramas:	9

Prerrequisitos:

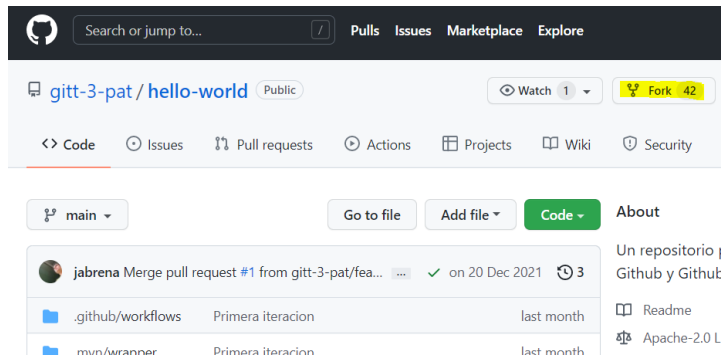
1. Cuenta en GitHub y Token:

He creado una cuenta con CarlotaCiruelos. (NombreApellido) como el nombre de usuario.

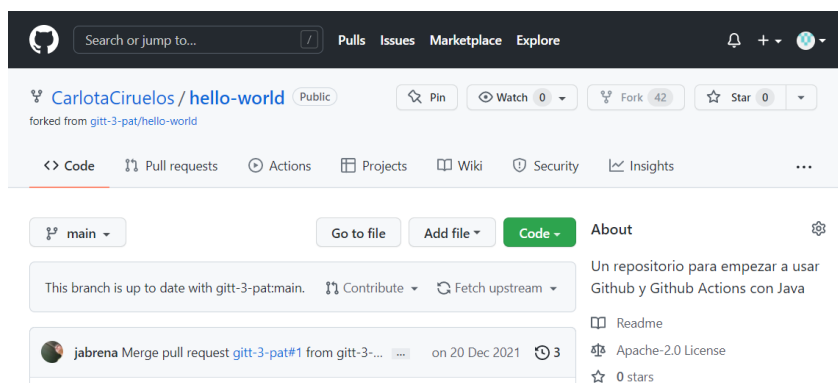
Y con este token: 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

2. Fork del repositorio:

Pinchamos en Fork y se copia el repositorio en mi cuenta de Git



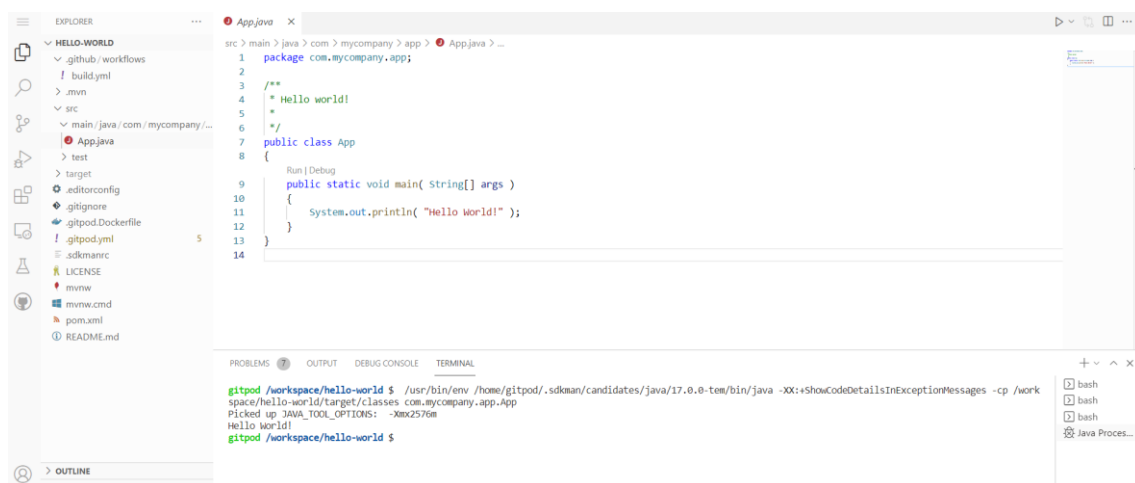
El link al ejemplo es: <https://github.com/CarlotaCiruelos/hello-world.git>



3. Configuración de Gitpod:

Abrimos un repositorio en Gitpod y comprobamos que le ejemplo funciona:

Este es el link: <https://carlotaciruelos-hellowor-07a6rwevdql.ws-eu29.gitpod.io/>



Entorno de desarrollo:

Esta evidencia esta también en una carpeta denominada recursos con un archivo denominado evidencias.txt e imágenes en el repositorio de git clonado en mi cuenta.

1. Java 17 y Maven:

```
C:\Users\carlo>java -version
java version "17.0.1" 2021-10-19 LTS
Java(TM) SE Runtime Environment (build 17.0.1+12-LTS-39)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.1+12-LTS-39, mixed mode, sharing)

C:\Users\carlo>mvn -version
Apache Maven 3.8.1 (05c21c65bdfed0f71a2f2ada8b84da59348c4c5d)
Maven home: C:\Users\carlo\apache-maven-3.8.1\bin\..
Java version: 17.0.1, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-17.0.1
Default locale: es_ES, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```

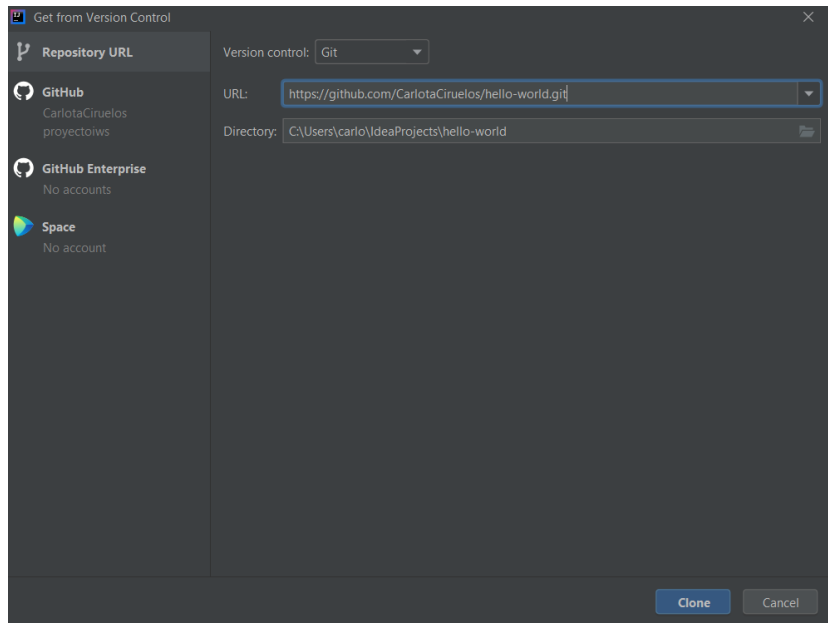
2. IntelliJ:



Desarrollo de la práctica:

1. git clone:

Este comando nos permite clonar o copiar un directorio ya existente en un nuevo directorio. En nuestro caso el repositorio original se accede por máquina remota por GitHub y lo copiamos en el sistema de archivos local.



```
12:56:23.998: [..\..\IdeaProjects] git -c ... clone --progress --recurse-submodules https://github.com/CarlotaCiruelos/hello-world.git -- hello-world1
Cloning into 'hello-world1'...
Receiving objects: 44% (17/38)
remote: Total 38 (delta 1), reused 31 (delta 0), pack-reused 0
Resolving deltas: 100% (1/1), done.
```

2. git status:

El comando git status muestra el estado de la rama de trabajo. Permite ver los cambios que no sean añadido de forma local (antes del add), los que solo se han añadido de forma local (después del add y antes del commit) y al hacer push lo publicamos a remoto.

antes del add:

```
C:\Users\carlo\IdeaProjects\hello-world>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)
      src/main/resources/prueba1.txt

nothing added to commit but untracked files present (use "git add" to track)
```

Antes del commit:

```
C:\Users\carlo\IdeaProjects\hello-world>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:   src/main/resources/prueba1.txt
```

Antes del push:

```
C:\Users\carlo\IdeaProjects\hello-world>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

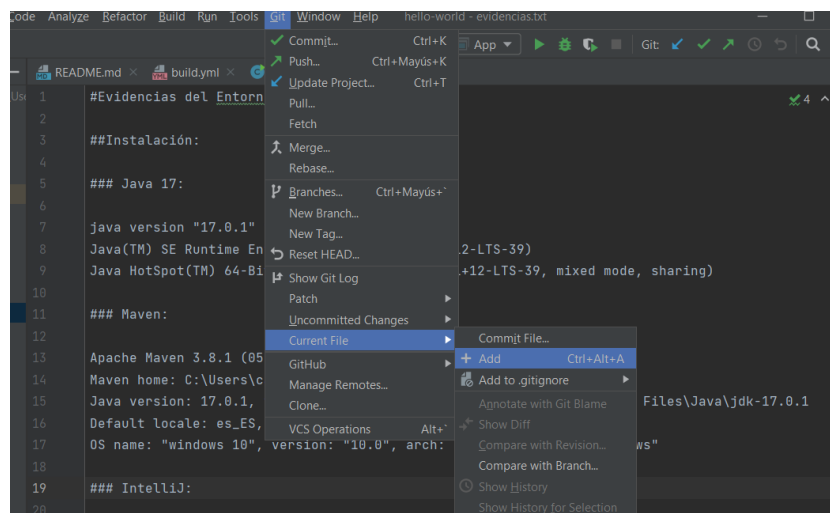
Después del push:

```
C:\Users\carlo\IdeaProjects\hello-world>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

3. git add:

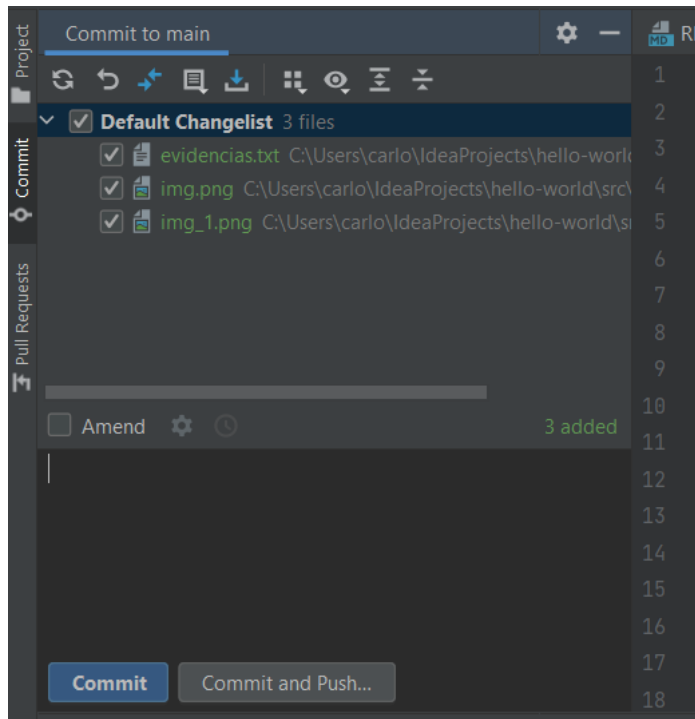
El comando git add añade archivos de forma local.



```
12:46:30.268: [hello-world] git -c ... add --ignore-errors -A -- src/main/resources/evidencias.txt
```

4. git commit:

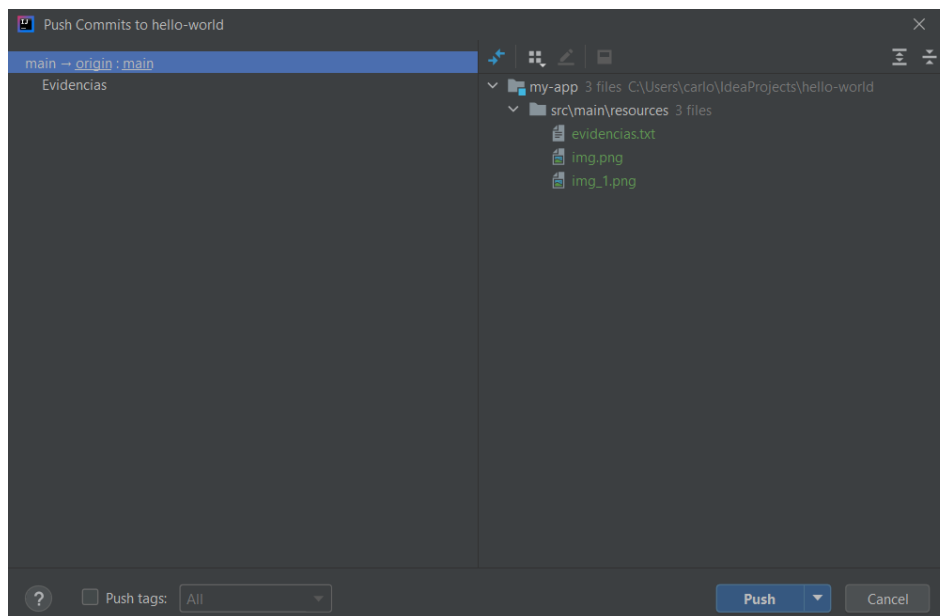
Confirma una instantánea del directorio del entorno de ensayo en el historial de confirmaciones de los repositorios.



```
13:08:28.938: [hello-world] git -c ... commit -F C:\Users\carlo\AppData\Local\Temp\git-commit-msg-.txt --
[main 66f1c6c] Evidencias
3 files changed, 42 insertions(+)
create mode 100644 src/main/resources/evidencias.txt
create mode 100644 src/main/resources/img.png
create mode 100644 src/main/resources/img_1.png
```

5. git push:

El comando git push se usa para cargar contenido del repositorio local a un repositorio remoto.



```
13:08:57.103: [hello-world] git -c ... push --progress --porcelain origin refs/heads/main:main
Writing objects: 100% (8/8), 115.70 KiB | 14.46 MiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
Remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/CarlotaCiruelos/hello-world.git
refs/heads/main:refs/heads/main 48fe276..66f1c6c
Done
```

6. git checkout:

El comando git checkout te permite desplazarte entre las ramas creadas por git branch . Al extraer una rama, se actualizan los archivos en el directorio de trabajo para reflejar la versión almacenada en esa rama y se indica a Git que registre todas las confirmaciones nuevas en dicha rama.

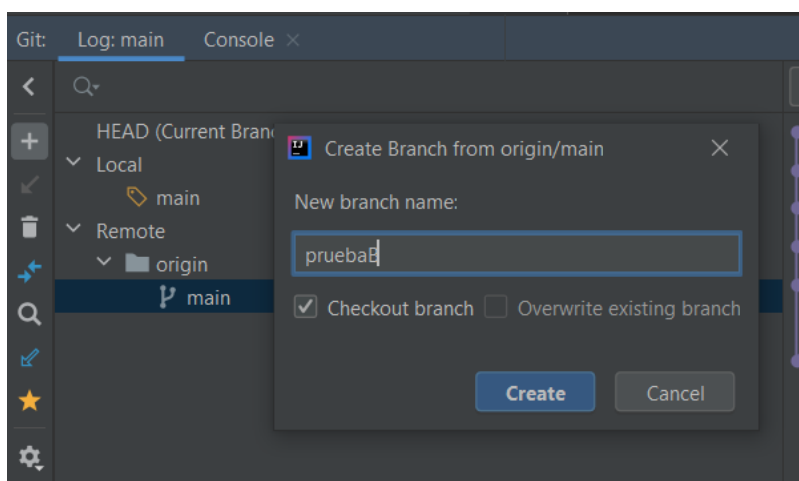
```
C:\Users\carlo\IdeaProjects\hello-world>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

Materia adicional:

1. Crear otra rama:

Esta es una funcionalidad muy útil cuando queremos trabajar más de una persona en el mismo proyecto. Puesto que permite a un equipo de desarrolladores avanzar simultáneamente en el mismo proyecto y luego unirlos.

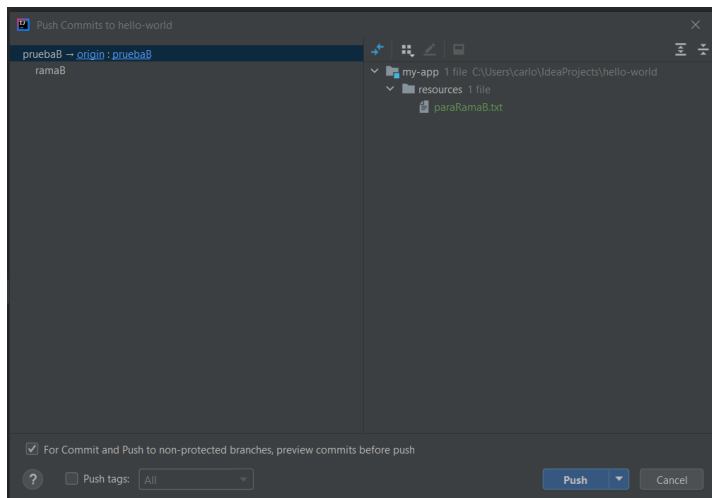
- a. Creamos la rama y hacemos un push de la rama para que aparezca en GitHub



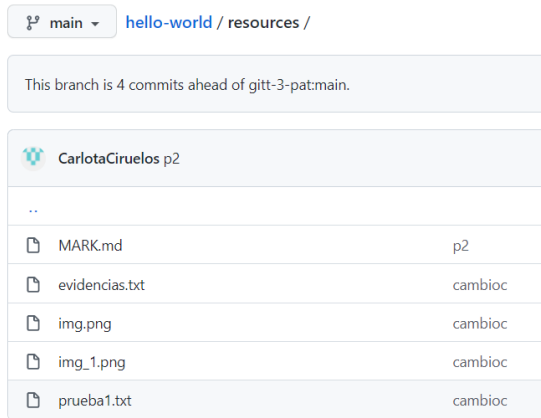
- b. Ahora estamos posicionados en la rama pruebaB. Creamos un archivo y hacemos un add+commit+push. Y podemos comprobar como solo aparece en nuestra nueva rama:

```
C:\Users\carlo\IdeaProjects\hello-world>git status
On branch pruebaB
Your branch is up to date with 'origin/pruebaB'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   resources/paraRamaB.txt
```



pruebaB → origin: pruebaB	
ramaB	
This branch is 5 commits ahead of gitt-3-pat:main.	
CarlotaCiruelos ramaB	
..	
MARK.md	p2
evidencias.txt	cambioc
img.png	cambioc
img_1.png	cambioc
paraRamaB.txt	ramaB
prueba1.txt	cambioc



main hello-world / resources /

This branch is 4 commits ahead of gitt-3-pat:main.

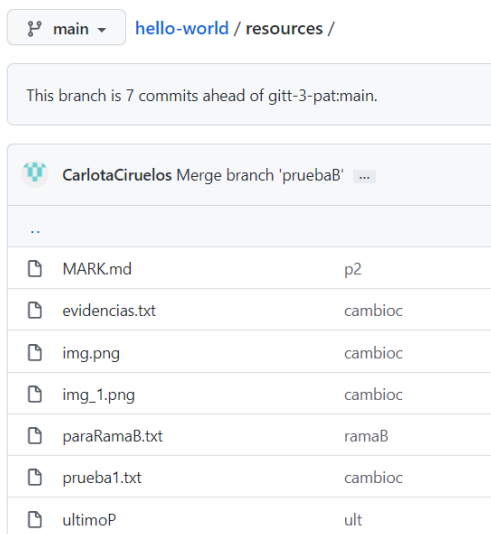
CarlotaCiruelos p2

MARK.md	p2
evidencias.txt	cambioc
img.png	cambioc
img_1.png	cambioc
prueba1.txt	cambioc

2. MERGE: Unir los cambios de dos ramas:

Con el comando `git merge`, conseguimos que los cambios que hemos hecho en la rama pruebaB (habíamos añadido un text) se actualicen en la rama main (que es en la que nos hemos posicionado con el comando `checkout`). Ahora bien, merge es en local para que los cambios aparezcan en GitHub debemos hacer un push de nuestra rama.

```
C:\Users\carlo\IdeaProjects\hello-world>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
C:\Users\carlo\IdeaProjects\hello-world>git merge pruebaB
Merge made by the 'recursive' strategy.
 resources/paraRamaB.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 resources/paraRamaB.txt
```



main hello-world / resources /

This branch is 7 commits ahead of gitt-3-pat:main.

CarlotaCiruelos Merge branch 'pruebaB' ...

MARK.md	p2
evidencias.txt	cambioc
img.png	cambioc
img_1.png	cambioc
paraRamaB.txt	ramaB
prueba1.txt	cambioc
ultimoP	ult