

29 DE ENERO DE 2023



COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

ICADE

CIHS

PRACTICA 1 PAT

PROF. ALBERTO PERALEDA DE LA LLAVE

ARTURO FERNÁNDEZ GARCÍA

En esta práctica se pretende entrar en contacto con la herramienta de GitHub así como completar la instalación de otras herramientas que se utilizarán a lo largo de la asignatura.

GitHub:

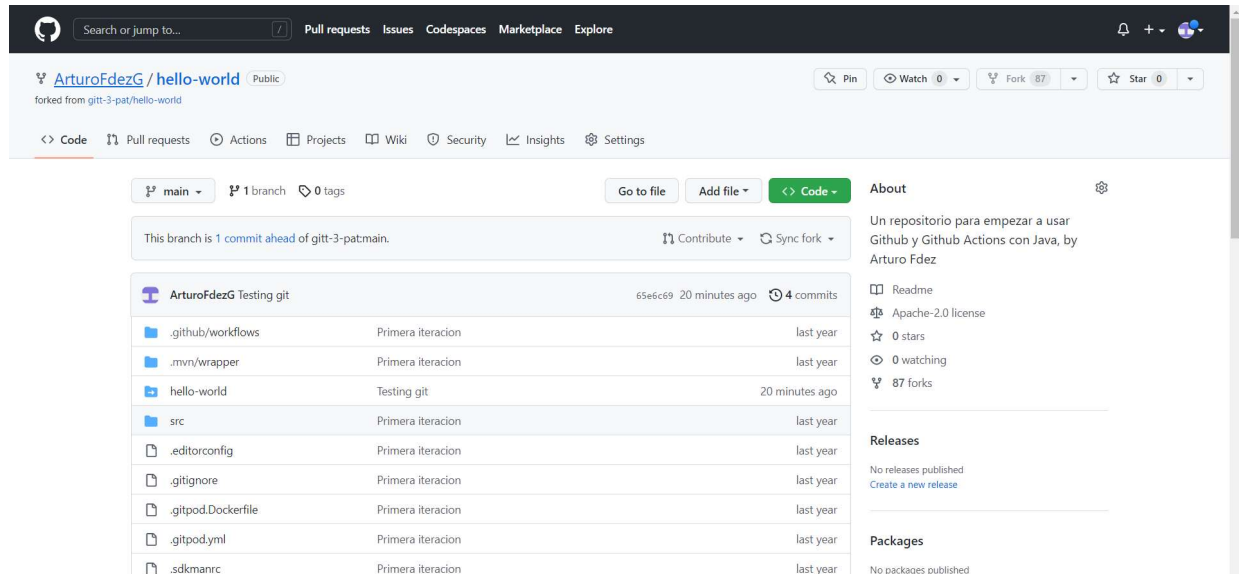


Imagen del repositorio

En primer lugar, se realiza un fork sobre el repositorio original, <https://github.com/gitt-3-pat/hello-world>, con lo que se creará una copia del repositorio en local, permitiendo la modificación de este sin que los cambios se vean reflejados en el original.

En segundo lugar y siguiendo los pasos de la práctica, se clona nuestro propio repositorio, lo que añadirá una copia de este en el main del mismo.

```
● @ArturoFdezG →/workspaces/hello-world (main) $ git clone https://github.com/ArturoFdezG/hello-world
Cloning into 'hello-world'...
remote: Enumerating objects: 38, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 38 (delta 0), reused 0 (delta 0), pack-reused 34
Unpacking objects: 100% (38/38), 59.54 KiB | 1.75 MiB/s, done.
○ @ArturoFdezG →/workspaces/hello-world (main) $ █

● @ArturoFdezG →/workspaces/hello-world (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)
    hello-world/

nothing added to commit but untracked files present (use "git add" to track)
○ @ArturoFdezG →/workspaces/hello-world (main) $ █
```

Como se puede observar con el comando `git status`, se ha creado un nuevo archivo, y por lo tanto que lo tenemos sin seguimiento. También nos informa de que nos encontramos en el main (rama principal).

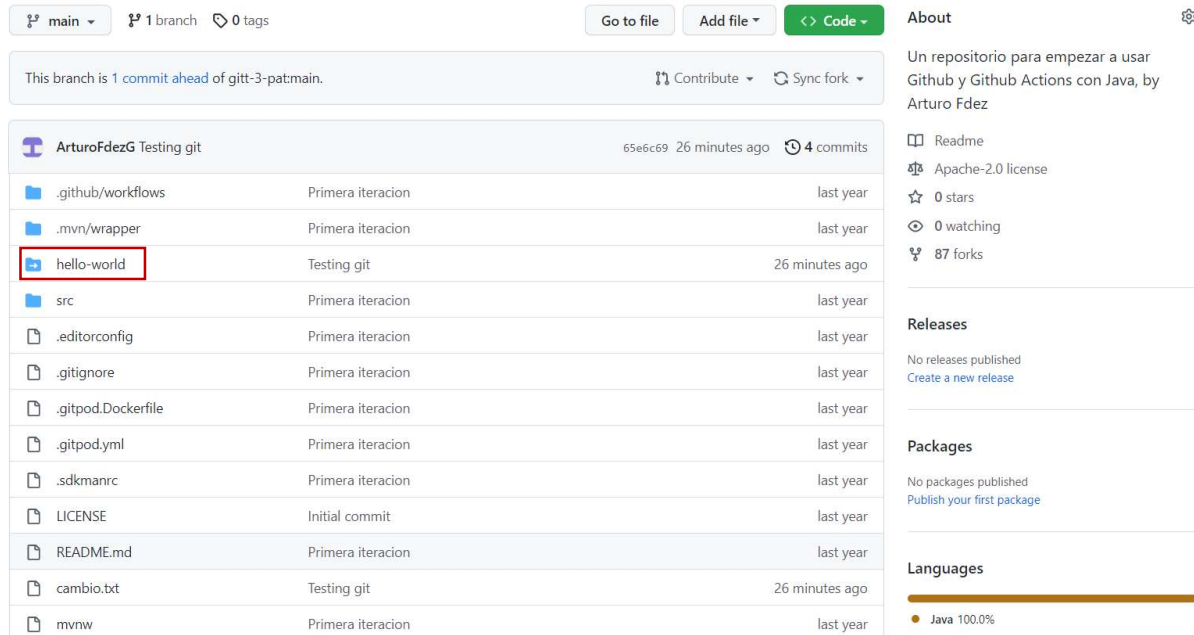


Imagen del repositorio

Como se puede apreciar, se crea una copia del repositorio dentro del mismo, con el nombre de *hello-world* (la señalada en rojo).

Nota: esta captura se ha realizado al final de la práctica, después de haber ejecutado comandos como `commit` y `push` que todavía no se han explicado.

Para observar mejor el funcionamiento del comando `git status`, se crea un fichero de prueba (`cambio.txt`) y se vuelve a ejecutar el comando, con lo que podremos apreciar la aparición de un nuevo archivo sin seguimiento.

```
@ArturoFdezG →/workspaces/hello-world (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)
  cambio.txt
  hello-world/

nothing added to commit but untracked files present (use "git add" to track)
@ArturoFdezG →/workspaces/hello-world (main) $
```

Este problema se puede solucionar con el comando `git add`. el cual da seguimiento a dicho archivos que antes no lo tenían. Volemos a comprobar y ya no hay problema, aparecen con seguimiento (en verde).

```

● @ArturoFdezG →/workspaces/hello-world (main) $ git add .
warning: adding embedded git repository: hello-world
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> hello-world
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached hello-world
hint:
hint: See "git help submodule" for more information.
● @ArturoFdezG →/workspaces/hello-world (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:   cambio.txt
    new file:   hello-world

○ @ArturoFdezG →/workspaces/hello-world (main) $ █

```

El siguiente paso es utilizar el comando `commit` para confirmar los cambios realizados (en local), para lo cual se especifica una breve descripción, en nuestro caso *"Testing git"*.

```

● @ArturoFdezG →/workspaces/hello-world (main) $ git commit -m "Testing git"
[main 884157d] Testing git
2 files changed, 2 insertions(+)
create mode 100644 cambio.txt
create mode 160000 hello-world
○ @ArturoFdezG →/workspaces/hello-world (main) $ █

```

A continuación, se actualiza el repositorio y se publican los cambios mediante el comando `git push` de la siguiente manera.

```

● @ArturoFdezG →/workspaces/hello-world (main) $ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 363 bytes | 363.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ArturoFdezG/hello-world
48fe276..65e6c69  main -> main
○ @ArturoFdezG →/workspaces/hello-world (main) $ █

```

ArturoFdezG / **hello-world** Public
forked from gitt-3-pat/hello-world

<> Code Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file <> Code

This branch is up to date with gitt-3-pat/hello-world:main. Contribute Sync fork

jabrena Merge pull request gitt-3-pat#1 from gitt-3-pat/feature/1 48fe276 on Dec 20, 2021 3 commits

.github/workflows	Primera iteracion	last year
.mvn/wrapper	Primera iteracion	last year
src	Primera iteracion	last year
.editorconfig	Primera iteracion	last year
.gitignore	Primera iteracion	last year
.gitpod.Dockerfile	Primera iteracion	last year
.gitpod.yml	Primera iteracion	last year
.sdkmanrc	Primera iteracion	last year
LICENSE	Initial commit	last year
README.md	Primera iteracion	last year
mvnw	Primera iteracion	last year
mvnw.cmd	Primera iteracion	last year
pom.xml	Primera iteracion	last year

Antes de hacer push

ArturoFdezG / **hello-world** Public
forked from gitt-3-pat/hello-world

<> Code Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file <> Code

This branch is 1 commit ahead of gitt-3-pat:main. Contribute Sync fork

ArturoFdezG Testing git 65e6c69 1 minute ago 4 commits

.github/workflows	Primera iteracion	last year
.mvn/wrapper	Primera iteracion	last year
hello-world	Testing git	1 minute ago
src	Primera iteracion	last year
.editorconfig	Primera iteracion	last year
.gitignore	Primera iteracion	last year
.gitpod.Dockerfile	Primera iteracion	last year
.gitpod.yml	Primera iteracion	last year
.sdkmanrc	Primera iteracion	last year
LICENSE	Initial commit	last year
README.md	Primera iteracion	last year
cambio.txt	Testing git	1 minute ago
mvnw	Primera iteracion	last year
mvnw.cmd	Primera iteracion	last year
pom.xml	Primera iteracion	last year

Después de hacer push

Por último, se emplearán comandos para moverse entre distintas ramas (branch) del repositorio. En este caso, se crea una nueva rama llamada “testing_branch” y comprobamos su estado.

```
● @ArturoFdezG →/workspaces/hello-world (main) $ git checkout -b testing_branch
Switched to a new branch 'testing_branch'
● @ArturoFdezG →/workspaces/hello-world (testing_branch) $ git status
On branch testing_branch
nothing to commit, working tree clean
○ @ArturoFdezG →/workspaces/hello-world (testing_branch) $
```

Tras su creación, hacemos push de esta para ver los cambios reflejados en el repositorio, como se puede observar en las siguientes imágenes.

```
● @ArturoFdezG →/workspaces/hello-world (testing_branch) $ git push --set-upstream origin testing_branch
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'testing_branch' on GitHub by visiting:
remote:   https://github.com/ArturoFdezG/hello-world/pull/new/testing_branch
remote:
To https://github.com/ArturoFdezG/hello-world
 * [new branch]      testing_branch -> testing_branch
Branch 'testing_branch' set up to track remote branch 'testing_branch' from 'origin'.
```



Volvemos al main y comprobamos su estado. Como se puede observar en la terminal, cambiamos entre las distintas ramas mediante el mismo comando, git checkout <rama>.

```
● @ArturoFdezG →/workspaces/hello-world (testing_branch) $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
● @ArturoFdezG →/workspaces/hello-world (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
○ @ArturoFdezG →/workspaces/hello-world (main) $
```

Una vez hechas las pruebas, eliminamos la rama anteriormente creada.

```
● @ArturoFdezG →/workspaces/hello-world (65e6c69) $ git branch -d testing_branch
Deleted branch testing_branch (was 65e6c69).
○ @ArturoFdezG →/workspaces/hello-world (65e6c69) $
```

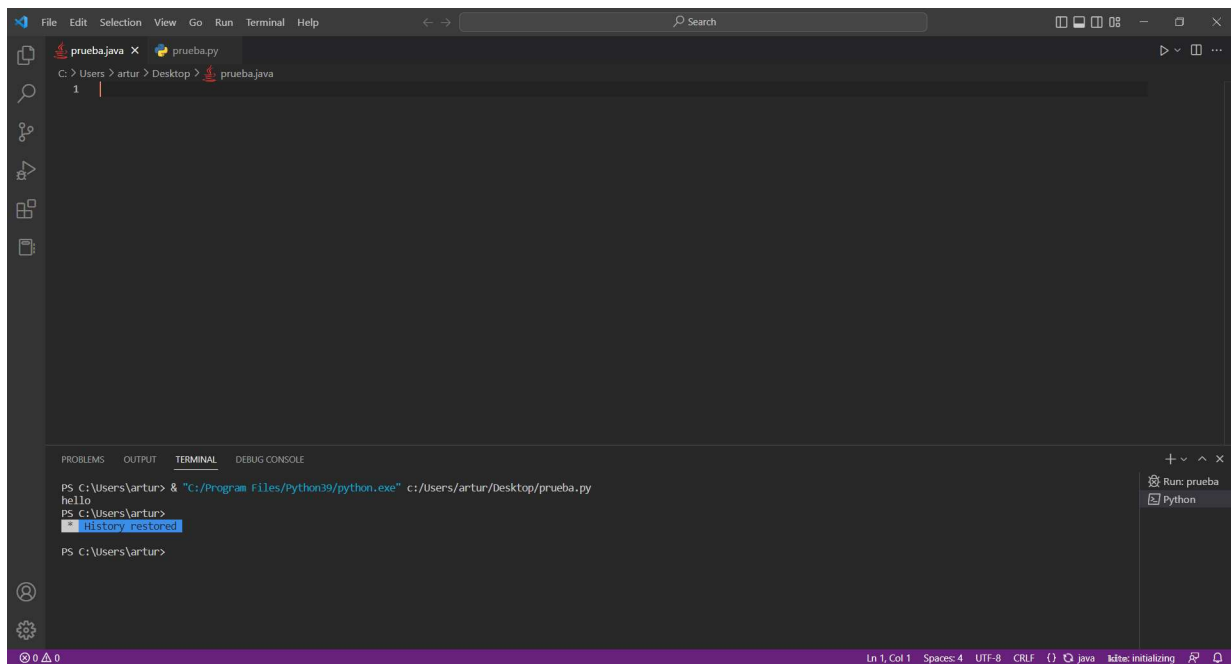


Otras herramientas:

```
C:\Users\artur>java --version
java 17.0.6 2023-01-17 LTS
Java(TM) SE Runtime Environment (build 17.0.6+9-LTS-190)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.6+9-LTS-190, mixed mode, sharing)

C:\Users\artur>
```

```
Apache Maven 3.8.5 (3599d3414f046de2324203b78ddcf9b5e4388aa0)
Maven home: C:\maven
Java version: 17.0.6, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-17
Default locale: es_ES, platform encoding: Cp1252
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"
```



```
C:\Users\artur>docker run hello-world
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

<https://hub.docker.com/>

For more examples and ideas, visit:

<https://docs.docker.com/get-started/>

```
C:\Users\artur>
```