Daniel Losada Rueda 3ºB

PRÁCTICA 1:

PLATAFORMA GITHUB & GIT

Procesamiento de Aplicaciones Telemáticas

29/01/2022

#1 clone

```
gitpod /workspace/hello-world $ git clone https://github.com/losadarr/hello-world
Cloning into 'hello-world'...
remote: Enumerating objects: 38, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 38 (delta 1), reused 31 (delta 0), pack-reused 0
Receiving objects: 100% (38/38), 58.97 KiB | 5.90 MiB/s, done.
Resolving deltas: 100% (1/1), done.
```

Git clone se utiliza para copiar un repositorio de Git que ya existe, en un nuevo directorio local. Esta acción crea un nuevo directorio local para el repositorio copiando el contenido de éste. Asique dentro de “hello-world” tenemos otro “hello-world”.

- Creo un documento llamado practica1 en el repositorio, y dentro un mensaje.

#2 status

```
gitpod /workspace/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)

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    Practica1/Practica1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        hello-world/
        practica1
```

Status me indica que el archivo “practica1” ha sido modificado. Éste comando muestra el estado del directorio de trabajo y el área staging o de preparación. Te deja ver qué cambios se han realizado.

#3 add

```
gitpod /workspace/hello-world $ git add practica1
```

Add no devuelve nada, agrega archivos nuevos o modificados en el directorio de trabajo al área de preparación o staging. Éste es un comando muy importante aunque tenga la reputación de lo contrario ya que sin el ningún commit haría nada.

#4 commit

```
gitpod /workspace/hello-world $ git commit -m "Esto es una prueba, mandaremos el txt"
[main f3e3b0f] Esto es una prueba, mandaremos el txt
 1 file changed, 1 insertion(+)
 create mode 100644 practica1
```

Commit sirve para hacer una “snapshot” de tu repositorio en un momento especifico. Si se piensa en el directorio de trabajo como nuestra área de trabajo “en progreso”, significa que los cambios no se reflejan aun en el repositorio de Git. Commit pone tus modificaciones en tu repositorio local.

#5 push

```
gitpod /workspace/hello-world $ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 308 bytes | 308.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/losadarr/hello-world.git
   48fe276..f03e96a  main -> main
```

El comando **Push** se usa para cargar contenido del repositorio local al repositorio remoto. Push es cómo se transfieren confirmaciones de un repositorio local a un repositorio remoto. Es la contraparte de git fetch, que descarga todos los branches al repo local.

#6 checkout

```
gitpod /workspace/hello-world $ git checkout f03e96a
Note: switching to 'f03e96a'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at f03e96a Esto es una prueba, mandaremos el txt
```

Checkout es el acto de cambiar entre diferentes versiones de una entidad de destino, le dice a GIT qué rama o confirmación quieres que vea los cambios aplicados. Aplica cambios a la rama correcta.

+ 7 log

```
gitpod /workspace/hello-world $ git log --oneline
f03e96a (HEAD, origin/main, origin/HEAD, main) Esto es una prueba, mandaremos el txt
48fe276 (upstream/main) Merge pull request #1 from gitt-3-pat/feature/1
5b68377 Primera iteracion
5038239 Initial commit
```

De forma predeterminada, Git Log devuelve una entrada de registro completa para cada confirmación realizada en un repositorio. Puede recuperar una lista de ID de confirmación y sus mensajes de confirmación asociados mediante el indicador - - oneline. Podemos ver los ID de confirmación y la primera línea de los mensajes asociados con una confirmación.

- ### Instalación Java17

```
C:\Users\danie>java -version
java version "17.0.2" 2022-01-18 LTS
Java(TM) SE Runtime Environment (build 17.0.2+8-LTS-86)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.2+8-LTS-86, mixed mode, sharing)

C:\Users\danie>
```

- ### Instalación Maven

```
Microsoft Windows [Versión 10.0.19044.1466]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\danie>mvn -v
Apache Maven 3.8.4 (9b656c72d54e5bacbed989b64718c159fe39b537)
Maven home: C:\apache-maven-3.8.4
Java version: 17.0.2, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-17.0.2
Default locale: es_ES, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\danie>
```

- ### Instalacion Intellij