PRÁCTICA 1

PROGRAMACIÓN DE APLICACIONES TELEMÁTICAS

Rubén Rodríguez Gómez
ICAI | 3ºB

Previo

Antes de proceder a completar las cuestiones propuestas para esta práctica, como prerrequisito será necesario realizar un fork sobre el repositorio propuesto por la misma: https://github.com/gitt-3-pat/hello-world.

Un fork consiste principalmente en hacer una copia exacta de un proyecto generando dos URL distintas. Esto se suele hacer por seguridad de tal forma que los cambios en el original no se trasladan de forma automática a la copia. Así en caso de fallo, siempre se conservará la versión original.

1. Prueba de comandos

   Una vez realizado el previo se va a proceder con la primera parte de esta práctica.
   En primer lugar, se investigará sobre el comando ***git clone***:

```
@Rubenzoo →/tmp $ git clone  https://github.com/gitt-3-pat/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 | 791.00 KiB/s, done.
@Rubenzoo →/tmp $ ls
codespaces_logs  hello-world           vscode-git-0804143195.sock
dockerd.log      hsperfdata_codespace  vscode-ipc-5b1e579f-e1fb-42e0-877a-eab6a1ca39f6.sock
editors          sshd.log              vscode-ipc-dd694552-e9cc-46a3-9155-0cf226345bb4.sock
```

   Este comando lo que hace es clonar el repositorio correspondiente al enlace https://github.com/gitt-3-pat/hello-world. De esta forma, podremos trabajar en la copia y no directamente en el original.
   Se comrpueba que en /tmp se ha clonado el repositorio hello-world.

   Una vez realizado esto, se procederá a trabajar el resto de la práctica en el directorio de workspaces.
   Ahora se procederá a probar ***git status:***

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

nothing to commit, working tree clean
```

   Este comando informa acerca de los cambios realizados en el repositorio. Se relaciona con los comandos ***git add, git commit y git push***. Estos dos se probarán a continuación para ver su efecto sobre el ***git status.***

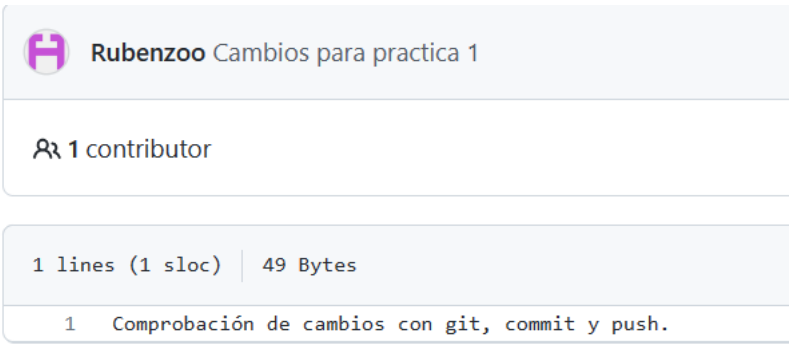
Para poder modificar un archivo, es necesario emplear los tres comandos especificados arriba.
Una vez se modifique un archivo aparecerá lo siguiente con el git status:

```
@Rubenzoo ➜/workspaces/hello-world/pr1 (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

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

no changes added to commit (use "git add" and/or "git commit -a")
```

Como se puede observar, una vez se realicen cambios, el status te recomienda utilizar ***git add*** para poder crear un borrador con lo modificado en el documento. Una vez se haya creado el borrador, con ***git commit*** se guardará dicho borrador. En este caso, se procederá a utilizar un ***git commit -m "texto"*** para poder indicar un mensaje de cambio (se comprobará con la utilización del comando ***git log***). Por último, se lanzarán a la nube, o en este caso, a mi repositorio, todos los cambios realizados en el documento:

```
@Rubenzoo ➜/workspaces/hello-world/pr1 (main) $ git add comandos.txt
@Rubenzoo ➜/workspaces/hello-world/pr1 (main) $ git commit -m "Cambios para practica 1"
[main 766b6ec] Cambios para practica 1
 1 file changed, 1 insertion(+), 1 deletion(-)
@Rubenzoo ➜/workspaces/hello-world/pr1 (main) $ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 385 bytes | 385.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Rubenzoo/hello-world.git
   e61f289..766b6ec  main -> main
```

```
@Rubenzoo ➜/workspaces/hello-world/pr1 (main) $ git log
commit 766b6ec751f0d15e8f9d0cd628c0aff91d630a25 (HEAD -> main, origin/main)
Author: Rubenzoo <102907885+Rubenzoo@users.noreply.github.com>
Date:   Fri Jan 27 12:48:15 2023 +0000

    Cambios para practica 1
```



Una vez visto los anteriores comandos, se procede ahora a finalizar la prueba de comandos con el análisis del comando ***git checkout***:
Este comando sirve principalmente para navegar

```
@Rubenzoo →/workspaces/hello-world (766b6ec) $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

Principalmente sirve para cambiar entre ramas y versiones especificas de un mismo repositorio. También se puede emplear para deshacer cambiar locales no guardados.

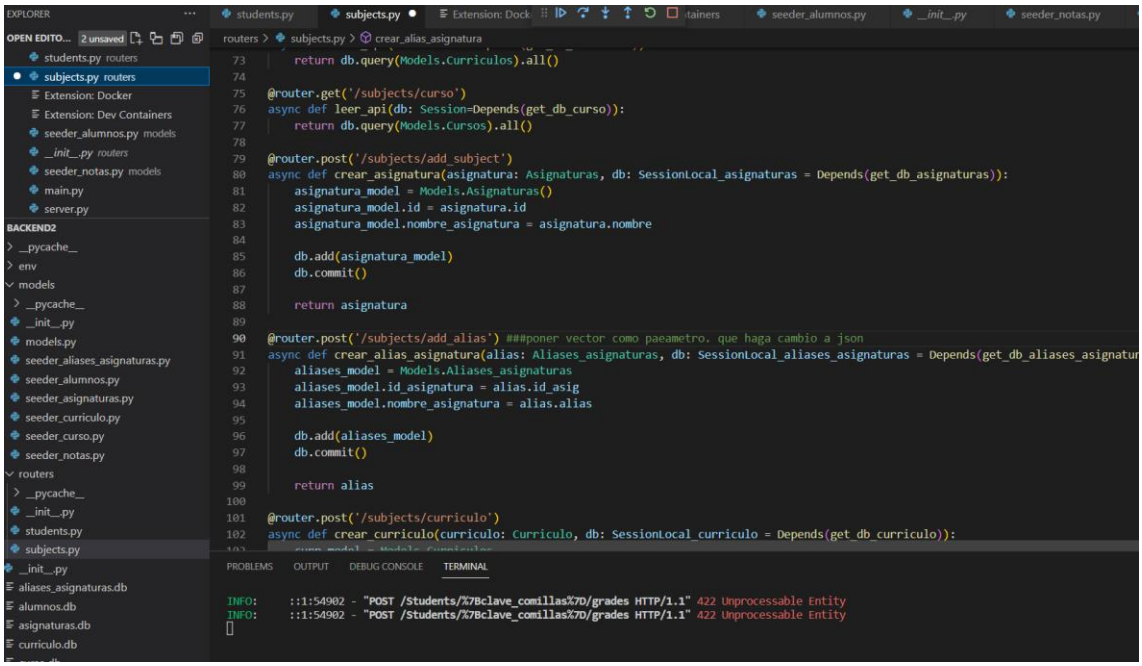
2. Instalación

En este apartado de la práctica se procederá a instalar los softwares necesarios para este curso de PAT.

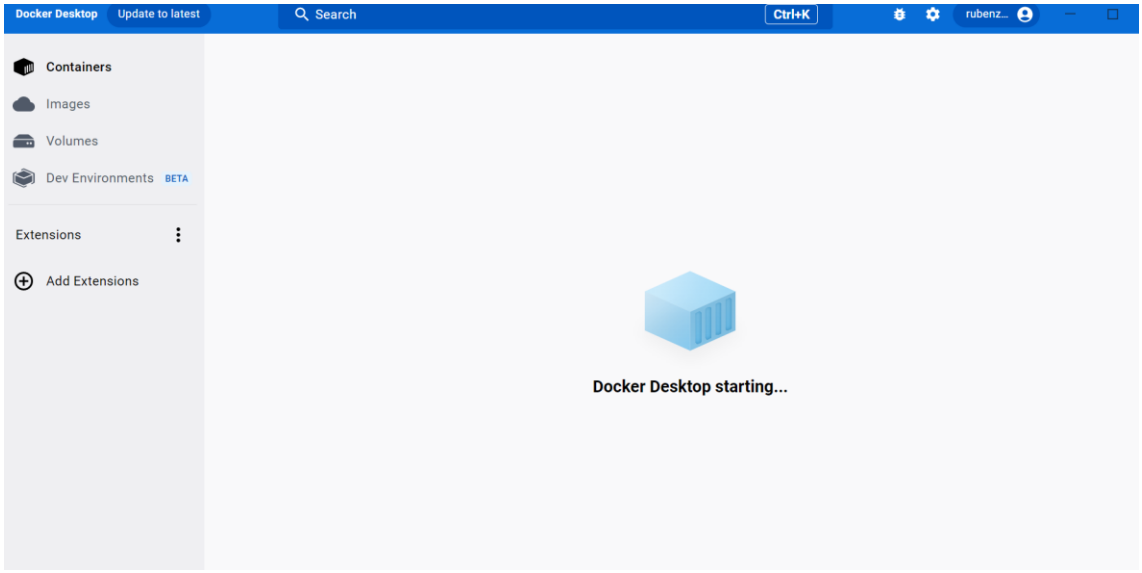
En primer lugar, se procede con la instalación de Java 17:

```
PS C:\Users\HP> java --version
java 18 2022-03-22
Java(TM) SE Runtime Environment (build 18+36-2087)
Java HotSpot(TM) 64-Bit Server VM (build 18+36-2087, mixed mode, sharing)
PS C:\Users\HP>
```

Ahora como editor de código fuente utilizare VSCode. Esta interfaz la llevo empleando para distintos proyectos desde hace aproximadamente un año:



Ahora se procede con la instalación de Docker:



Por último, se procederá con la instalación de Maven:

```
PS C:\Users\HP> mvn --version
Apache Maven 3.8.7 (b89d5959fcde851dcb1c8946a785a163f14e1e29)
Maven home: C:\Program Files\apache-maven-3.8.7-bin\apache-maven-3.8.7
Java version: 1.8.0_312, vendor: Temurin, runtime: C:\Program Files\Eclipse Adoptium\jre-8.0.312.7-hotspot
Default locale: es_ES, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```