

Práctica 1: Plataforma GitHub & Git

Intro

El objetivo de esta práctica era la puesta a punto del entorno de trabajo que se va a usar durante el curso, que comprende la instalación de los recursos requeridos y el uso de comandos básicos de Git con el objetivo de familiarizarnos con ellos.

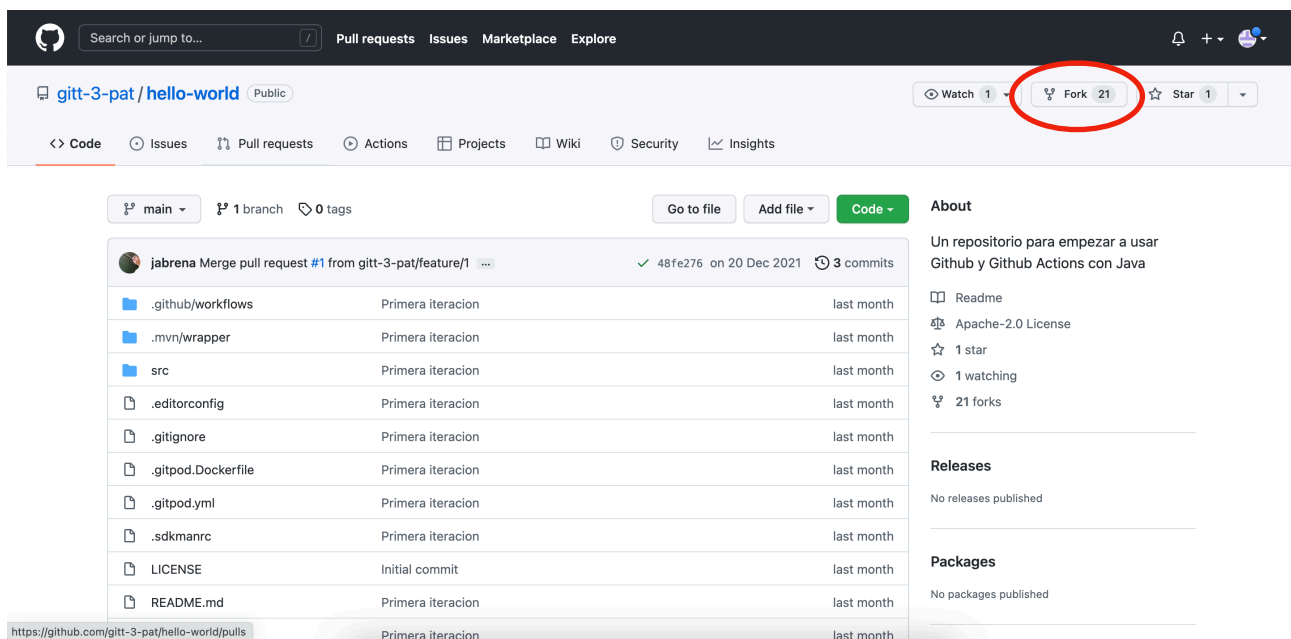
Desarrollo

El primer paso ha sido instalar la versión de Java requerida (17) , Maven e IntelliJ. Estas dos últimas ya estaban en el equipo por la asignatura de Ingeniería de Software del semestre pasado.

```
gitpod /workspace/hello-world $ java -version
Picked up JAVA_TOOL_OPTIONS: -Xmx2576m
openjdk version "17" 2021-09-14
OpenJDK Runtime Environment Temurin-17+35 (build 17+35)
OpenJDK 64-Bit Server VM Temurin-17+35 (build 17+35, mixed mode, sharing)
gitpod /workspace/hello-world $ mvn -v
Picked up JAVA_TOOL_OPTIONS: -Xmx2576m
Apache Maven 3.8.3 (ff8e977a158738155dc465c6a97ffaf31982d739)
Maven home: /home/gitpod/.sdkman/candidates/maven/current
Java version: 17, vendor: Eclipse Adoptium, runtime: /home/gitpod/.sdkman/candidates/java/17.0.0-tem
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "5.13.0-1012-gcp", arch: "amd64", family: "unix"
```

Para el desarrollo de esta práctica, se ha usado Gitpod en vez de IntelliJ para estar más familiarizado con otras herramientas.

Entrando ya en los comandos Git, lo primero ha sido realizar un fork en el repositorio indicado:



The screenshot shows the GitHub interface for the repository 'gitt-3-pat/hello-world'. The 'Fork' button is highlighted with a red circle. The repository page displays a list of files and folders, including .github/workflows, .mvn/wrapper, src, .editorconfig, .gitignore, .gitpod.Dockerfile, .gitpod.yml, .sdkmanrc, LICENSE, and README.md. The 'About' section on the right provides information about the repository, including the license (Apache-2.0) and the number of forks (21).

Una vez hecho, comenzamos con los comandos. Primero, **git clone** nos permite replicar un repositorio, en este caso para hacer la prueba clonamos el mismo repositorio hello-world dentro de sí mismo:

```
gitpod /workspace/hello-world $ git clone https://github.com/alexalzaga/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 | 6.55 MiB/s, done.
Resolving deltas: 100% (1/1), done.
```

Si hacemos algún cambio en algún elemento del repositorio, por ejemplo añadir código en el archivo App.java, **git status** nos muestra si hay cambios locales. Si no hay cambios:

```
gitpod /workspace/hello-world $ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Una vez cambiamos algo, usar este comando nos avisa de los cambios locales que no se han guardado en el repo:

```
src > main > java > com > mycompany > app > App.java > App > main(String[])
1 package com.mycompany.app;
2
3 /**
4  * Hello world!
5  *
6  */
7 public class App
8 {
9     public static void main( String[] args )
10    {
11        System.out.println( "Hello World!" );
12        System.out.println( "Bye World!" );
13    }
14 }
15
```

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL

```
nothing to commit, working tree clean
gitpod /workspace/hello-world $ 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:   src/main/java/com/mycompany/app/App.java

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

Para guardar estos cambios podemos usar **git add** para especificar el elemento que queremos actualizar, y **git commit** para efectuar la actualización en el repositorio en el que estamos trabajando. La extensión **-m** nos permite ponerle nombre para identificar la actualización que hemos realizado:

```
gitpod /workspace/hello-world $ git add /workspace/hello-world/src/main/java/com/mycompany/app/App.java
gitpod /workspace/hello-world $
gitpod /workspace/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)
        modified:   src/main/java/com/mycompany/app/App.java
gitpod /workspace/hello-world $ git commit -m "P1: Primer commit"
[main ceb5bf2] P1: Primer commit
1 file changed, 1 insertion(+) _
```

Usando **git status** otra vez nos notifica de que nuestra rama local tiene un commit que no está confirmado en el repositorio real. Para confirmar esta actualización, que está a nivel local, usamos el comando **git push**, que actualizará finalmente el repositorio de GitHub.

```
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)

nothing to commit, working tree clean
gitpod /workspace/hello-world $ git push
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (9/9), 579 bytes | 579.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/alexalzaga/hello-world.git
48fe276..ceb5bf2  main -> main
```

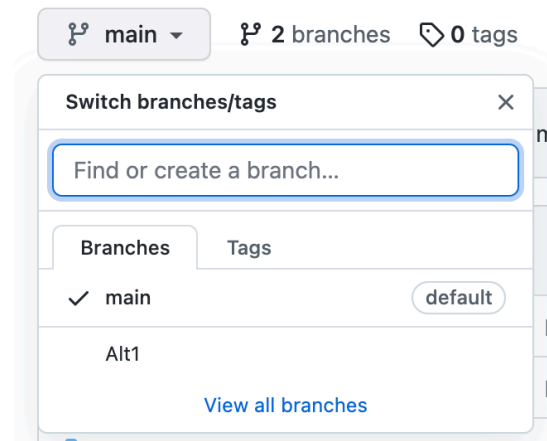
Para crear nuevos branches, podemos usar **git checkout -b nombredelbranch**. Con **git checkout nombredelbranch** podemos cambiar el branch activo, y con **git branch** podemos consultar cuántos branches hay en el repositorio. Como siempre, para actualizar el repositorio real usamos push, en este caso **git push origin nombredelbranch** para añadir el nuevo branch.

```
gitpod /workspace/hello-world $ git checkout -b Alt1
Switched to a new branch 'Alt1'

gitpod /workspace/hello-world $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

gitpod /workspace/hello-world $ git branch
Alt1
* main
gitpod /workspace/hello-world $ git push origin Alt1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'Alt1' on GitHub by visiting:
remote:   https://github.com/alexalzaga/hello-world/pull/new/Alt1
remote:
To https://github.com/alexalzaga/hello-world.git
* [new branch]      Alt1 -> Alt1
```

Aquí vemos que, tras el push, nuestra rama nueva es visible en el repositorio desde GitHub.



Si queremos borrar un branch, podemos usar **git branch -d nombredelbranch**, y actualizarlo en el repositorio real con otro push origin como anteriormente.

```
gitpod /workspace/hello-world $ git branch -d Alt1
Deleted branch Alt1 (was ceb5bf2).
gitpod /workspace/hello-world $ git branch
* main
gitpod /workspace/hello-world $ git push origin :Alt1
To https://github.com/alexalzaga/hello-world.git
- [deleted]      Alt1
```

Otra herramienta útil (git diff) nos permite ver las diferencias que hay entre el repositorio local en el que estamos trabajando y la versión real. El texto verde en este caso es el que hemos añadido, y el rojo lo hemos quitado:




```
gitpod /workspace/hello-world $ git diff
diff --git a/src/main/java/com/mycompany/app/App.java b/src/main/java/com/mycompany/app/App.java
index 203536e..a3a0be9 100644
--- a/src/main/java/com/mycompany/app/App.java
+++ b/src/main/java/com/mycompany/app/App.java
@@ -9,6 +9,6 @@ public class App
     public static void main( String[] args )
     {
         System.out.println( "Hello World!" );
-        System.out.println( "Bye World!" );
+        System.out.println( "World?" );
     }
 }
```

Si queremos actualizar un branch en particular, podemos hacer el commit local, y al hacer push determinamos en que branch queremos hacerlo. En este caso, actualizamos sólo la branch "alt", y si vemos el GitHub después vemos cómo en la branch "main" el commit más reciente es el primero que hemos hecho, y en la branch "alt" el más reciente es este segundo:


```




gitpod /workspace/hello-world $ git add /workspace/hello-world/src/main/java/com/mycompany/app/App.java
gitpod /workspace/hello-world $ git commit -m "P1: Segundo commit"
[alt b64c417] P1: Segundo commit
1 file changed, 1 insertion(+), 1 deletion(-)
gitpod /workspace/hello-world $ git push origin alt
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (9/9), 587 bytes | 587.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/alexalzaga/hello-world.git
   ceb5bf2..b64c417  alt -> alt

```


 main ▾
  2 branches
  0 tags
 Go to file
Add file ▾
Code ▾

This branch is 1 commit ahead of gitt-3-pat:main.
 Contribute ▾
Fetch upstream ▾

 alexalzaga P1: Primer commit
 ceb5bf2 20 minutes ago 4 commits

 alt ▾
  2 branches
  0 tags
 Go to file
Add file ▾
Code ▾

This branch is 2 commits ahead of gitt-3-pat:main.
 Contribute ▾
Fetch upstream ▾

 alexalzaga P1: Segundo commit
 b64c417 4 minutes ago 5 commits

Por último, vamos a ver la diferencia entre dos branches con **git diff *branchorigen branchdestino***, y después vamos a usar **git merge *nombredelbranch*** para fusionar alt con main, que es la branch desde la que hacemos la acción. De esta manera, ahora nuestra branch "main" va a actualizarse con el segundo commit, que previamente habíamos hecho solo en la branch "alt":

```

gitpod /workspace/hello-world $ git diff alt main
diff --git a/src/main/java/com/mycompany/app/App.java b/src/main/java/com/mycompany/app/App.java
index a3a0be9..203536e 100644
--- a/src/main/java/com/mycompany/app/App.java
+++ b/src/main/java/com/mycompany/app/App.java
@@ -9,6 +9,6 @@ public class App
     public static void main( String[] args )
     {
         System.out.println( "Hello World!" );
-        System.out.println( "World?" );
+        System.out.println( "Bye World!" );
     }
 }
gitpod /workspace/hello-world $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
gitpod /workspace/hello-world $ git merge alt
Updating ceb5bf2..b64c417
Fast-forward
 src/main/java/com/mycompany/app/App.java | 2 +--
 1 file changed, 1 insertion(+), 1 deletion(-)

```


 main ▾ 2 branches 0 tags

Go to file

Add file ▾

Code ▾

This branch is 2 commits ahead of gitt-3-pat:main.

 Contribute ▾ Fetch upstream ▾

alexalzaga P1: Segundo commit

b64c417 8 minutes ago ⌚ 5 commits



.github/workflows

Primera iteracion

last month



.mvn/wrapper

Primera iteracion

last month



src

P1: Segundo commit

8 minutes ago