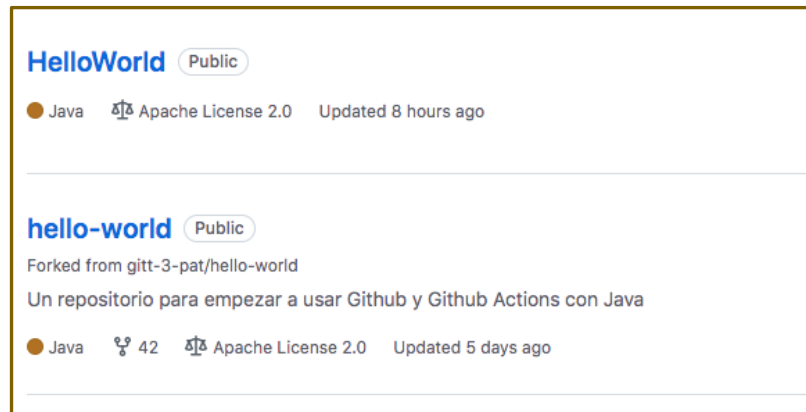


PRÁCTICA 1: PLATAFORMA GITHUB & GIT

Durante esta primera práctica se procede a mostrar el correcto funcionamiento de unos comandos sobre nuestro GitHub, el cual ha sido previamente creado.

Para comenzar la práctica se realiza un *fork* sobre el repositorio del Github proporcionado por el profesor. De esta manera, se puede visualizar los diferentes archivos que contiene el link en nuestro Github y se puede comenzar a trabajar con ello. En la siguiente figura se muestran los repositorios que contiene nuestro Github. Comentar que el primero se creo ya que la propia aplicación te daba la opción de incluir el repositorio introduciendo el link proporcionado para hacer el fork. Se ha comprobado por lo tanto que de esa manera también se incluyen todos los archivos. El segundo *hello-world* como se puede observar se ha realizado correctamente el fork y será con el que trabajaremos.



El entorno de desarrollo requerido para la asignatura consta de Java 17, Maven e IntelliJ. Las siguientes imágenes muestran que en efecto se ha instalado correctamente el software.

Primeramente, una vez instalado Java 17 y haciendo uso del comando *java -version* se observa que la versión de java instalada es la 17.0.2:

```
(base) MacBook-Air-de-alfredo:~ Cristina$ 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)
```

En cuanto a la instalación de maven ha sido un poco más enrevesada ya que una vez descargado se ha tenido que trasladar a Aplicaciones.

```
(base) MacBook-Air-de-alfredo:~ Cristina$ cd /Applications/apache-maven-3.8.4
(base) MacBook-Air-de-alfredo:apache-maven-3.8.4 Cristina$ pwd
/Applications/apache-maven-3.8.4
(base) MacBook-Air-de-alfredo:apache-maven-3.8.4 Cristina$ cd /Users/Cristina/
```

Una vez trasladado se ha introducido el comando `ls -a` para buscar si se nos había creado el fichero `bash_profile`, al cual efectivamente aparecía y se había creado correctamente:

```
(base) MacBook-Air-de-alfredo:~ Cristina$ ls -a
.                  .oracle_jre_usage
..                 .p2
.CFUserTextEncoding .packettracer
.DS_Store          .r
.RData             .rstudio-desktop
.Rapp.history       .subversion
.Rhistory           .tcshrc
.Trash              .tooling
.anaconda           .xonshrc
.android            .zshrc
.astah              Applications
.bash_history       Desktop
.bash_profile      Documents
.bash_sessions      Downloads
.bashrc             Library
.conda              MPLABXProjects
.condarc            Movies
.config             Music
.continuum           PacketTracer7
```

Dicho documento contenía lo siguiente que se muestra a continuación mediante el comando `open -e bash_profile`. No obstante, se ha añadido la última línea donde se indicaba donde exportarlo:

```
.bash_profile
|
export PATH="$PATH":"/Applications/microchip/xc32/v2.41/bin"

# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup='${/opt/anaconda3/bin/conda}' 'shell.bash' 'hook' 2> /dev/null)
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "/opt/anaconda3/etc/profile.d/conda.sh" ]; then
        . "/opt/anaconda3/etc/profile.d/conda.sh"
    else
        export PATH="/opt/anaconda3/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda initialize <<<

export PATH="/Applications/apache-maven-3.8.4/bin:$PATH"
```

Una vez guardados los cambios, se cargan con el comando `source bash.profile` y ya se puede ver que la versión que se ha descargado es la 3.8.4 mediante el comando `mvn --version`

```
(base) MacBook-Air-de-alfredo:~ Cristina$ open -e .bash_profile
(base) MacBook-Air-de-alfredo:~ Cristina$ mvn --version
Apache Maven 3.8.4 (9b656c72d54e5baced989b64718c159fe39b537)
Maven home: /Applications/apache-maven-3.8.4
Java version: 17.0.2, vendor: Oracle Corporation, runtime: /Library/Java/JavaVirtualMachines/jdk-17.0.2.jdk/Contents/Home
Default locale: es_ES, platform encoding: UTF-8
OS name: "mac os x", version: "10.13.6", arch: "x86_64", family: "mac"
(base) MacBook-Air-de-alfredo:~ Cristina$
```

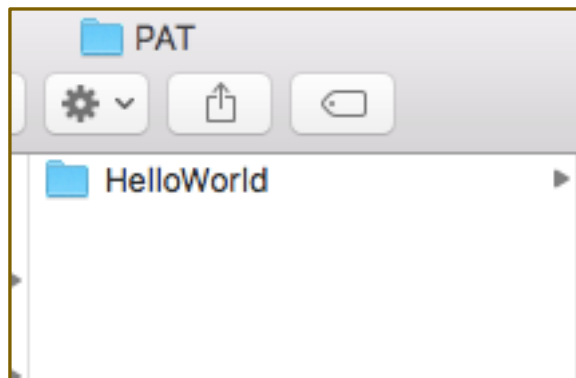
A continuación, se proceden a comprobar los diferentes comandos sobre la consola principalmente, aunque también se hace uso de IntelliJ.

□ Git clone

La idea principal detrás de este comando es copiar o clonar un repositorio ya existente, el cual puede ser tanto local como remoto.

```
(base) MacBook-Air-de-alfredo:Desktop Cristina$ cd /Users/Cristina/Desktop/PAT
(base) MacBook-Air-de-alfredo:PAT Cristina$ git clone https://github.com/CristinaAlvarez/HelloWorld
Cloning into 'HelloWorld'...
remote: Enumerating objects: 38, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 38 (delta 1), reused 38 (delta 1), pack-reused 0
Unpacking objects: 100% (38/38), done.
```

Tras situarnos en nuestra carpeta de PAT y hacer uso del comando *git clone + link del Github a clonar*, se puede observar que se ha copiado correctamente. No obstante, nos vamos a la carpeta de PAT para comprobar que efectivamente se ha creado dicha carpeta HelloWorld.



□ Git status

Este segundo comando nos da información sobre el estado del directorio de trabajo. Actualmente no hay nada en proceso de ser enviado al GitHub o para subir, por lo tanto al introducir el comando, se muestra que nos encontramos en el main y que no hay nada en *commit*, es decir que está limpio.

```
(base) MacBook-Air-de-alfredo:PAT Cristina$ cd HelloWorld
(base) MacBook-Air-de-alfredo:HelloWorld Cristina$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Se mostrará posteriormente lo que aparece una vez estamos enviando.

□ Git add

El comando `git add` va a añadir cambios al directorio del entorno de trabajo. Es muy útil para subir nuevos archivos en remoto o nuevas actualizaciones del mismo.

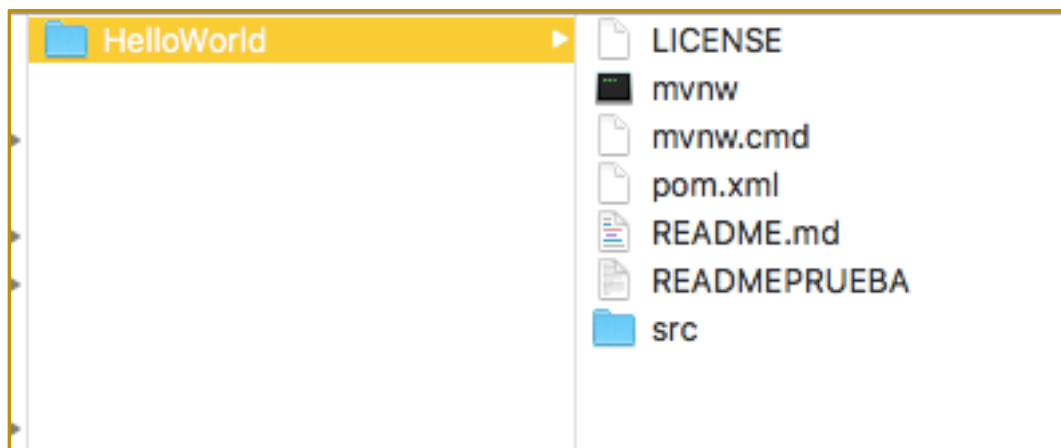
Para demostrar que funciona correctamente se ha creado un txt denominado READMEPRUEBA en la carpeta HelloWorld.

```
(base) MacBook-Air-de-alfredo:HelloWorld Cristina$ git add READMEPRUEBA.txt
(base) MacBook-Air-de-alfredo:HelloWorld Cristina$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   READMEPRUEBA.txt
```

Una vez aplicado el comando se comprueba que efectivamente se ha creado en nuestro directorio de nuestro entorno de trabajo.



Queda comprobado que se ha añadido correctamente el fichero txt a la carpeta de HelloWorld. Además, cabe mencionar que se ha hecho uso nuevamente del comando `git status`. En este caso, se observa en la captura de la consola anterior que los cambios están en proceso de *commit*, lo cual antes estaba vacío.

□ Git commit

El siguiente comando ya se ha mencionado brevemente en el apartado anterior. Va a servir para identificar los cambios realizados en el entorno de trabajo. Se podría decir que es un estado de preparación por el cual se están actualizando los cambios.

```
(base) MacBook-Air-de-alfredo:HelloWorld Cristina$ git commit -m comprobacion1
[main 1373b7c] comprobacion1
Committer: Cristina <Cristina@MacBook-Air-de-alfredo.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 100644 READMEPRUEBA.txt
```

Al final de las líneas de código de la consola que aparecen tras ejecutar el comando `git commit -m comprobación 1`, se observa que se ha cambiado el file y se ha insertado en mi Github.

Cristina and Cristina comprobacion1		1373b7c 2 days ago	🕒 4 commits
📁 .github/workflows	Primera iteracion	last month	
📁 .mvn/wrapper	Primera iteracion	last month	
📁 src	Primera iteracion	last month	
📄 .editorconfig	Primera iteracion	last month	
📄 .gitignore	Primera iteracion	last month	
📄 .gitpod.Dockerfile	Primera iteracion	last month	
📄 .gitpod.yml	Primera iteracion	last month	
📄 .sdkmanrc	Primera iteracion	last month	
📄 LICENSE	Initial commit	last month	
📄 README.md	Primera iteracion	last month	
📄 READMEPRUEBA.txt	comprobacion1	2 days ago	
📄 mvnw	Primera iteracion	last month	
📄 mvnw.cmd	Primera iteracion	last month	
📄 pom.xml	Primera iteracion	last month	

□ Git push

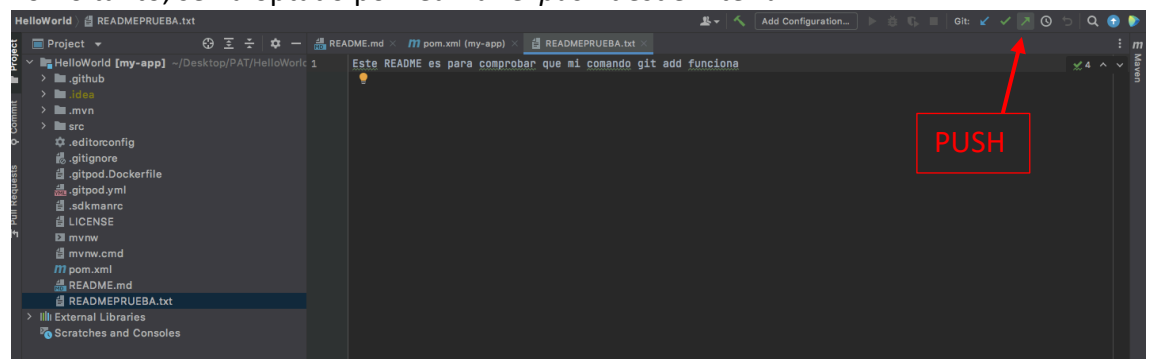
El comando *git push* va a subir los cambios realizados al entorno de trabajo, en este caso debería subirse tanto al GitHub como a IntelliJ.

Es un comando muy importante ya que si bien usando *commit* se guardan los cambios realizados, hasta que no se suban no se van a poder visualizar en el entorno de trabajo.

A la hora de ejecutar el comando *git push* en mi terminal me daba un fallo como se muestra a continuación:

```
(base) MacBook-Air-de-alfredo:HelloWorld Cristina$ git push
Username for 'https://github.com': CristinaAlvarez
Password for 'https://CristinaAlvarez@github.com':
remote: Support for password authentication was removed on August 13, 2021. Please
use a personal access token instead.
remote: Please see https://github.blog/2020-12-15-token-authentication-requirements
-for-git-operations/ for more information.
fatal: Authentication failed for 'https://github.com/CristinaAlvarez/HelloWorld/'
(base) MacBook-Air-de-alfredo:HelloWorld Cristina$
```

Por lo tanto, se ha optado por realizar el *push* desde IntelliJ.



Una vez se ha terminado de hacer push, se comprueba en el entorno de trabajo de Github que efectivamente se ha creado de manera remota:

A screenshot of the GitHub repository page for 'Cristina and Cristina comprobacion1'. The page shows a list of commits. The commit '1373b7c' from 9 hours ago is highlighted, showing 4 commits. The file 'READMEPRUEBA.txt' is highlighted in the commit list.

main	2 branches	0 tags	Go to file	Add file	Code
Cristina and Cristina comprobacion1 1373b7c · 9 hours ago 4 commits					
.github/workflows	Primera iteracion	last month			
.mvn/wrapper	Primera iteracion	last month			
src	Primera iteracion	last month			
.editorconfig	Primera iteracion	last month			
.gitignore	Primera iteracion	last month			
.gitpod.Dockerfile	Primera iteracion	last month			
.gitpod.yml	Primera iteracion	last month			
.sdkmanrc	Primera iteracion	last month			
LICENSE	Initial commit	last month			
README.md	Primera iteracion	last month			
READMEPRUEBA.txt	comprobacion1	9 hours ago			
mvnw	Primera iteracion	last month			
mvnw.cmd	Primera iteracion	last month			
pom.xml	Primera iteracion	last month			

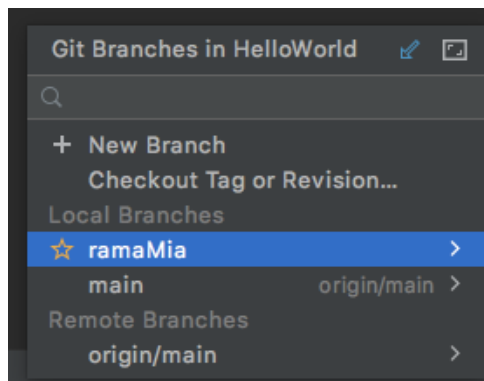
□ Git checkout

Si se hace uso del comando *git checkout* se puede desplazarse entre las diferentes ramas. Si se accede a una rama se actualizan los archivos del directorio de trabajo, de esta manera, por ejemplo, se pueden tener varias versiones del proyecto pudiendo realizar cambios sin perder lo anterior.

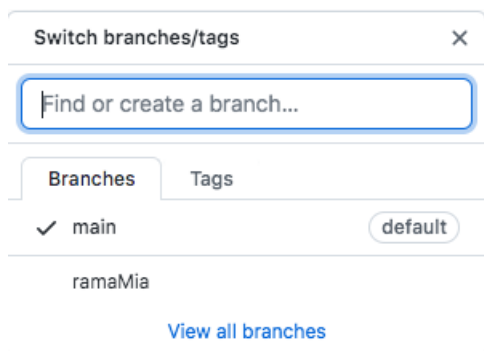
Se ha creado por lo tanto una nueva rama mediante el comando *git checkout -b ramaMia*. Como se muestra en la segunda línea de código se ha creado y se ha cambiado a una nueva rama, que se ha denominado *ramaMia*.

```
(base) MacBook-Air-de-alfredo:HelloWorld Cristina$ git checkout -b ramaMia
Switched to a new branch 'ramaMia'
(base) MacBook-Air-de-alfredo:HelloWorld Cristina$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
(base) MacBook-Air-de-alfredo:HelloWorld Cristina$
```

Se comprueba que efectivamente se ha creado la rama tanto en IntelliJ como en el Github:



Captura de IntelliJ

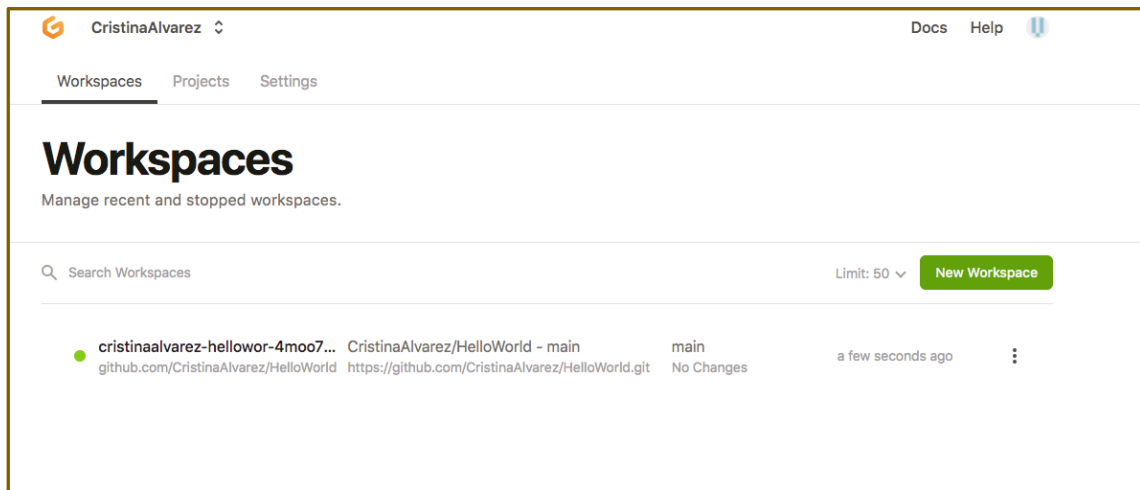


Captura de GitHub

Si la rama a la que se quiere acceder ya existe , como por ejemplo el main, se escribe en la consola *git checkout main*.

No obstante, se puede cambiar rápidamente entre ramas mediante IntelliJ sin tener que usar el comando.

Para finalizar con la práctica se muestra el Gitpod creado cuyo link es el siguiente:
<https://gitpod.io/workspaces>



Además, se puede acceder al Github mediante el siguiente link:

<https://github.com/CristinaAlvarez/HelloWorld.git>