

PRÁCTICA 1

PROGRAMACIÓN APLICACIONES TELEMÁTICAS

Eva Manrique Sanz
201910294 3ºA GITT

- OBJETIVO DE LA PRÁCTICA

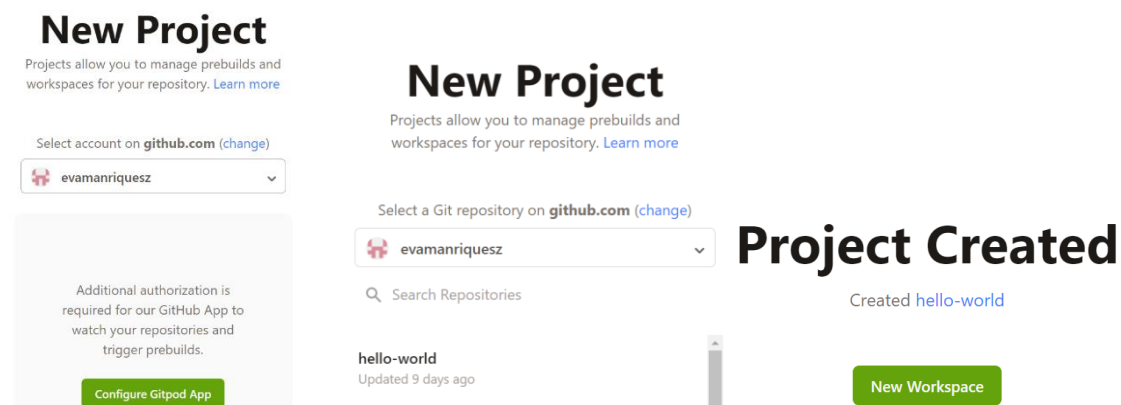
El objetivo de esta práctica es adquirir nociones básicas de GitHub para su posterior uso en otras prácticas, así como comprobar que se ha instalado todo el software necesario para la asignatura, verificando su correcto funcionamiento.

- REPOSITORIO DE GITHUB

<https://github.com/evamanriquesz/hello-world>

- PASOS PARA CLONAR EL REPOSITORIO

En primer lugar, creo un nuevo proyecto, seleccionando la opción de que se copien a GitPod todos los proyectos tanto pasados como futuros. Elijo que el proyecto sea hello-world.



- DESARROLLO DE LA PRÁCTICA (COMANDOS)

git clone → nos situamos en la carpeta en la que se va a ubicar el proyecto, y clonamos el repositorio utilizando su link.

```
C:\Users\evama\IdeaProjects\PAT>git clone https://github.com/evamanriquesz/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.
Receiving objects: 50% (19/38) 31 (delta 0), pack-reused 0
Receiving objects: 100% (38/38), 58.97 KiB | 838.00 KiB/s, done.
Resolving deltas: 100% (1/1), done.

C:\Users\evama\IdeaProjects\PAT>
```

git status → nos situamos en la carpeta del proyecto, y nos muestra el estado de este directorio.

En este primer caso, el directorio está al día.

```
C:\Users\evama\IdeaProjects\PAT\hello-world>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

En este segundo caso, hemos realizado y guardado modificaciones, pero no hemos hecho ni commit ni push.

```
C:\Users\evama\IdeaProjects\PAT\hello-world>git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   src/main/java/com/mycompany/app/Prueba4.java

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/Prueba4.java
```

Pruebo el comando en gitpod:

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

nothing to commit, working tree clean
```

git add → lo utilizamos para añadir ficheros a nuestro GitHub. Si sólo queremos añadir un fichero, podemos usar “git add .”, pero si, por el contrario, queremos añadir más de uno, podemos hacer “git add --all” para que añada todos los ficheros de manera recursiva.

En el primer caso, sólo añadimos un fichero.

```
C:\Users\evama\IdeaProjects\PAT\hello-world>git add .
```

En el segundo caso, añadiré dos.

```
<command> [xargs]
C:\Users\evama\IdeaProjects\PAT\hello-world>git add --all
warning: LF will be replaced by CRLF in src/main/java/com/mycompany/app/App.java.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/main/java/com/mycompany/app/Prueba.java.
The file will have its original line endings in your working directory
C:\Users\evama\IdeaProjects\PAT\hello-world>
```

Pruebo en gitpod:

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

git commit → GitHub está dividido en una especie de tres niveles: nivel local, un nivel intermedio de repositorio, y, por último, un nivel público en el que subes tu proyecto a GitHub, desde donde otras personas pueden descargarlo. Git commit sirve para actualizar tu nivel intermedio, confirmar un cambio a nivel de repositorio. Utilizamos “-am” para crear una confirmación de todos los cambios preparados, junto con un mensaje que informe sobre ellos.

```
C:\Users\evama\IdeaProjects\PAT\hello-world>git commit -am "Segundo commit"
[main 356ecf6] Segundo commit
 3 files changed, 9 insertions(+)
 create mode 100644 src/main/java/com/mycompany/app/Prueba2.java
 create mode 100644 src/main/java/com/mycompany/app/Prueba3.java
C:\Users\evama\IdeaProjects\PAT\hello-world>_
```

```
gitpod /workspace/hello-world $ git commit -am "doce"
[main c8d9912] doce
 1 file changed, 2 insertions(+)
```

git push → utilizamos este comando para guardar los cambios y modificaciones al más alto de los tres niveles mencionados previamente. Realizando push, las actualizaciones realizadas al proyecto quedarán publicadas en GitHub para que cualquier persona pueda descargarlas (mediante git pull). También se compartirán todas las modificaciones a los compañeros si estás programando en un repositorio de GitHub compartido. Ellos deberán hacer git pull para poder acceder a las novedades. (Nota: Aquí he tenido que añadir mi token).

```
C:\Users\evama\IdeaProjects\PAT\hello-world>git push
Enumerating objects: 29, done.
Counting objects: 100% (29/29), done.
Delta compression using up to 8 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (21/21), 1.42 KiB | 363.00 KiB/s, done.
Total 21 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 2 local objects.
To https://github.com/evamanriquesz/hello-world
 48fe276..356ecf6  main -> main
```

```
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% (5/5), done.
Writing objects: 100% (9/9), 636 bytes | 636.00 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/evamanriquesz/hello-world.git
 58dae5e..c8d9912  main -> main
```

git checkout → te permite desplazarte por ramas creadas por git branch. Para comprobarlo, he creado una nueva rama, llamada branch1 y he ejecutado el comando. Vemos que ha cambiado de rama.

```
C:\Users\evama\IdeaProjects\PAT\hello-world>git checkout branch1
Switched to branch 'branch1'
```

```
gitpod /workspace/hello-world $ git checkout branch1
Switched to branch 'branch1'
Your branch is up to date with 'origin/branch1'.
```

Para subir la creación de una nueva rama a GitHub, hacemos:

git --set-upstream origin NOMBRE_RAMA

```
C:\Users\evama\IdeaProjects\PAT\hello-world>git push --set-upstream origin branch1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'branch1' on GitHub by visiting:
remote:   https://github.com/evamanriquesz/hello-world/pull/new/branch1
remote:
To https://github.com/evamanriquesz/hello-world
* [new branch]      branch1 -> branch1
Branch 'branch1' set up to track remote branch 'branch1' from 'origin'.
```

- COMANDOS NO INCLUIDOS EN EL ENUNCIADO DE LA PRÁCTICA

git log --oneline → utilizamos este comando para ver los commits realizados anteriormente. Añadimos “--oneline” para que cada commit se muestre en una línea.

```
C:\Users\evama\IdeaProjects\PAT\hello-world>git log --oneline
e464ff8 (HEAD -> main) Sexto commit
e6385b4 (origin/main, origin/HEAD) Quinto commit
e3d4d9e Cuarto commit
5bd1ced Tercer commit
356ecf6 Segundo commit
ca6cc98 Primer commit
48fe276 Merge pull request #1 from gitt-3-pat/feature/1
5b68377 Primera iteracion
5038239 Initial commit

C:\Users\evama\IdeaProjects\PAT\hello-world>git revert e464ff8 --no-edit
[main 3521246] Revert "Sexto commit"
Date: Sat Jan 22 00:34:03 2022 +0100
2 files changed, 10 insertions(+), 39 deletions(-)
rewrite src/main/java/com/mycompany/app/Prueba2.java (87%)
rewrite src/main/java/com/mycompany/app/Prueba3.java (87%)
```

git revert XXXXXX --no-edit → utilizamos este comando para borrar un commit de los que encontramos en la lista anterior. Las X se sustituyen por el número situado a la izquierda de cada commit (en naranja en la imagen anterior) cuando hacemos git log. Esto es así para asegurarnos de que realizamos el comando de manera segura, y no borramos nada por error.

```
C:\Users\evama\IdeaProjects\PAT\hello-world>git log --oneline
e464ff8 (HEAD -> main) Sexto commit
e6385b4 (origin/main, origin/HEAD) Quinto commit
e3d4d9e Cuarto commit
5bd1ced Tercer commit
356ecf6 Segundo commit
ca6cc98 Primer commit
48fe276 Merge pull request #1 from gitt-3-pat/feature/1
5b68377 Primera iteracion
5038239 Initial commit

C:\Users\evama\IdeaProjects\PAT\hello-world>git revert e464ff8 --no-edit
[main 3521246] Revert "Sexto commit"
Date: Sat Jan 22 00:34:03 2022 +0100
2 files changed, 10 insertions(+), 39 deletions(-)
rewrite src/main/java/com/mycompany/app/Prueba2.java (87%)
rewrite src/main/java/com/mycompany/app/Prueba3.java (87%)
```

Nota: GitHub considera el revert como un nuevo commit:

```
C:\Users\evama\IdeaProjects\PAT\hello-world>git log --oneline
3521246 (HEAD -> main) Revert "Sexto commit"
e464ff8 Sexto commit
e6385b4 (origin/main, origin/HEAD) Quinto commit
e3d4d9e Cuarto commit
5bd1ced Tercer commit
356ecf6 Segundo commit
ca6cc98 Primer commit
48fe276 Merge pull request #1 from gitt-3-pat/feature/1
5b68377 Primera iteracion
5038239 Initial commit
```

git diff → se usa para ver los cambios que has realizado desde el último commit:

```
C:\Users\evama\IdeaProjects\PAT\hello-world>git diff
warning: LF will be replaced by CRLF in src/main/java/com/mycompany/app/Prueba5.java.
The file will have its original line endings in your working directory
diff --git a/src/main/java/com/mycompany/app/Prueba5.java b/src/main/java/com/mycompany/app/Prueba5.java
index f6efbb9..057571f 100644
--- a/src/main/java/com/mycompany/app/Prueba5.java
+++ b/src/main/java/com/mycompany/app/Prueba5.java
@@ -7,5 +7,10 @@ public class Prueba5 {
    }

+   public Prueba5(String var1, int var2)
+   {
+   }
+ }
}
```

git checkout -b NOMBRE_NUEVA_RAMA → se utiliza para crear una nueva rama, y colocarte en ella:

```
C:\Users\evama\IdeaProjects\PAT\hello-world>git checkout -b branch2
Switched to a new branch 'branch2'
```

git stash → se utiliza para guardar temporalmente un cambio que has realizado, pero todavía no está lo suficientemente perfeccionado como para guardarlo y confirmarlo (porque, por ejemplo, no esté terminado y vaya a dar error si lo ejecutas); en caso de que tengas que hacerlo para ponerte a trabajar en otra parte del código.

```
C:\Users\evama\IdeaProjects\PAT\hello-world>git status
On branch branch2
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/Prueba5.java

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

C:\Users\evama\IdeaProjects\PAT\hello-world>git stash
warning: LF will be replaced by CRLF in src/main/java/com/mycompany/app/Prueba5.java.
The file will have its original line endings in your working directory
Saved working directory and index state WIP on branch2: 58dae5e Commit 9 creo

C:\Users\evama\IdeaProjects\PAT\hello-world>git status
On branch branch2
nothing to commit, working tree clean
```

git stash pop → Se utiliza para volver a aplicar los cambios del stash que se ha realizado previamente:

```
C:\Users\evama\IdeaProjects\PAT\hello-world>git stash pop
On branch branch2
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/Prueba5.java

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (5b0a27734b39e20091f88b53cbf03556f455b9d2)
```

- EVIDENCIA INSTALACIÓN SOFTWARE

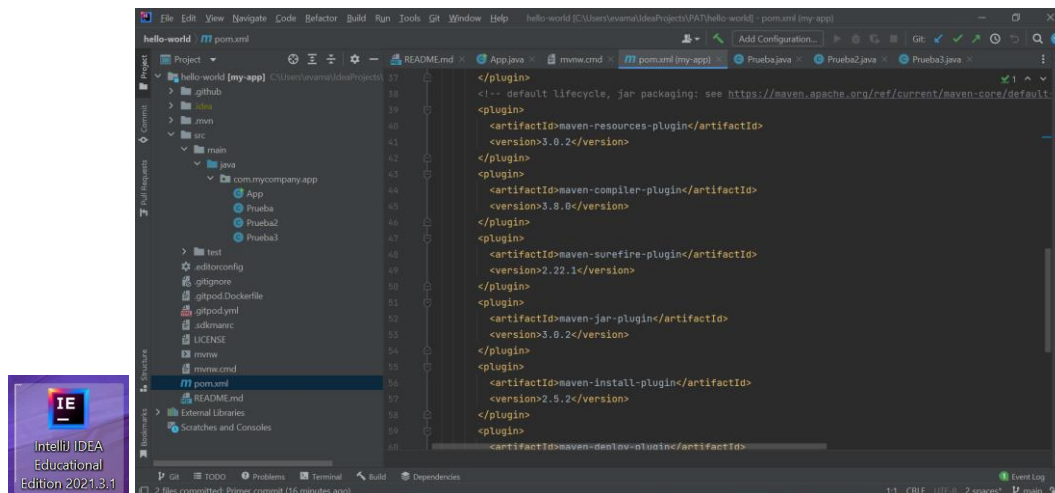
JAVA → Java 17 JDK

```
C:\Users>java -version
java version "17.0.1" 2021-10-19 LTS
Java(TM) SE Runtime Environment (build 17.0.1+12-LTS-39)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.1+12-LTS-39, mixed mode, sharing)
```

MAVEN → Apache Maven 3.8.1

```
C:\Users>mvn -version
Apache Maven 3.8.1 (05c21c65bdfed0f71a2f2ada8b84da59348c4c5d)
Maven home: C:\Program Files\apache-maven-3.8.1\bin\..
```

INTELLIJ → IntelliJ IDEA Educational Edition 2021.3.1



- COMPROBAR CORRECTO FUNCIONAMIENTO DEL SOFTWARE

Para evidenciar el correcto funcionamiento de todo el software de la práctica, se ha creado un pequeño programa en el proyecto hello-world, utilizando las clases App y Prueba, de manera que se cree un objeto Prueba, que tendrá un atributo nombre. Al inicializar este objeto no se especificará dicho, pero el programa te pedirá que introduzcas uno.

