

A diagram illustrating a distributed version control system. At the top center is a green square icon containing a white diamond with a branching symbol. Dotted lines radiate from this central icon to several other green square icons with the same branching symbol, arranged in a circular pattern around the center. Below each of these icons is a grey silhouette of a person, representing a developer. The entire diagram is set against a light green background.

Despliegue de aplicaciones Web

GIT

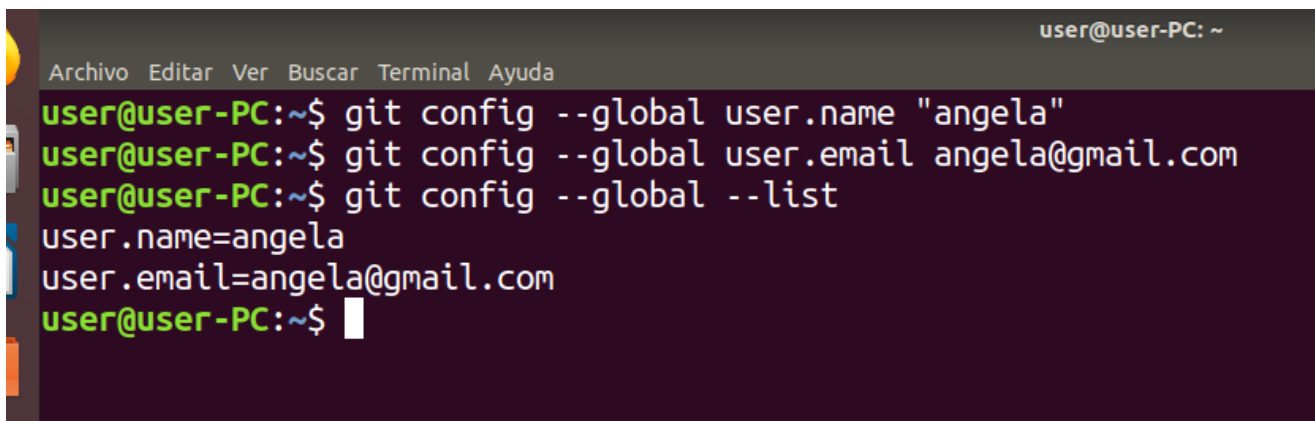
Ángela Pérez Álvarez 2ºD.A.W.

Apartado 1

Para la instalación de **GIT** lo primero que hacemos es ir a la [página oficial](#) y, en el apartado de descargas, seleccionamos la adecuada según nuestro sistema operativo y versión de este. En esta tarea utilizaremos el SO Ubuntu, por tanto, abrimos la terminal, actualizamos los repositorios y escribimos la siguiente línea de comando.

➤ `sudo apt-get install git`

Una vez instalado lo primero que debemos hacer es establecer un nombre y email, para ello escribimos lo siguiente:

A screenshot of a terminal window with a dark background. The title bar at the top right says 'user@user-PC: ~'. Below the title bar is a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal shows the following commands and output:

```
user@user-PC:~$ git config --global user.name "angela"
user@user-PC:~$ git config --global user.email angela@gmail.com
user@user-PC:~$ git config --global --list
user.name=angela
user.email=angela@gmail.com
user@user-PC:~$
```

Apartado 2

Puedes ver el documento usuarios.xml en el apartado 3

Antes de iniciar GIT, creamos la carpeta y el documento xml. Estando situado en la carpeta recién creada iniciamos GIT con la siguiente línea.

➤ `git init`

Ahora subimos el archivo a la rama master, para ello, utilizamos la línea:

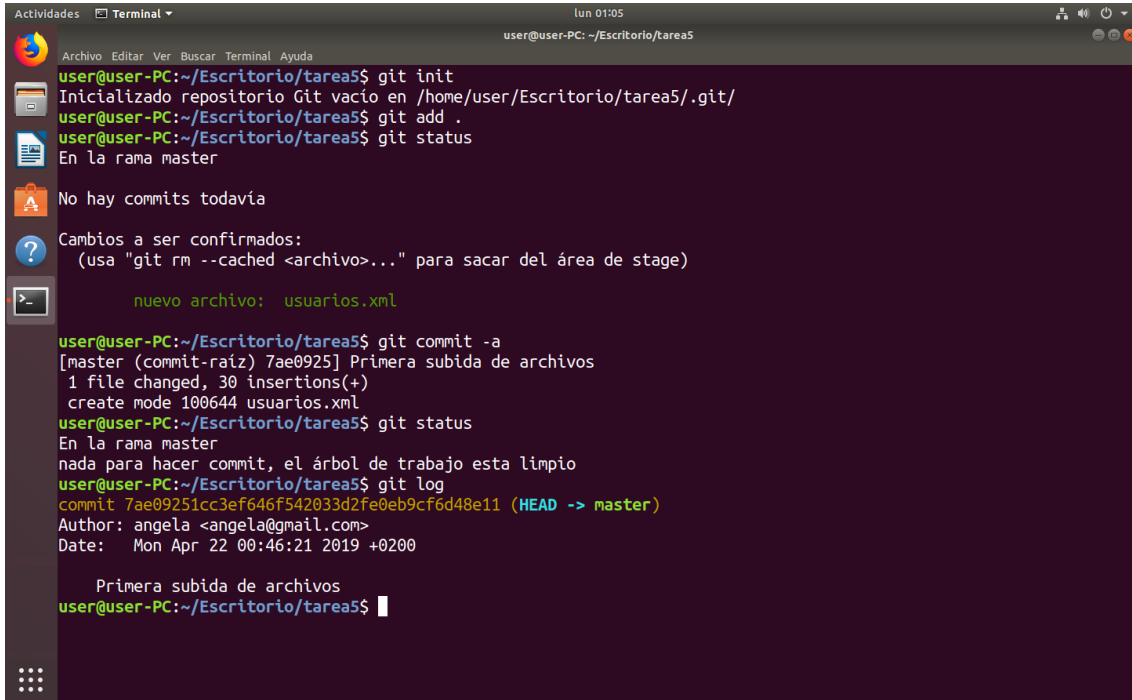
➤ `git add .`

Con la línea de arriba le hemos indicado que agregue a la rama master todo, pero aun no está agregado, ante de agregarlo finalmente, debemos



utilizar el siguiente comando en el que escribiremos una pequeña descripción de la acción o cambio que hemos realizado. Existen 2 formas:

- `git commit -a` (Aparecerá un editor)
- `git commit -m "Nuestro mensaje"`

A screenshot of a Linux terminal window. The title bar shows 'Actividades' and 'Terminal'. The terminal content shows a user at a PC in the directory ~/Escritorio/tarea5. They run 'git init', which initializes a repository. Then they run 'git add .' to add the current directory's contents. Next, they run 'git status', which shows that there are no commits yet and that changes are to be confirmed. They then run 'git commit -a', which creates a commit for the first time. The commit message is 'Primera subida de archivos'. The terminal shows the commit hash 7ae0925 and the file usuarios.xml. Finally, they run 'git log', which shows the commit history, including the first commit.

```
user@user-PC:~/Escritorio/tarea5$ git init
Iniciado repositorio Git vacío en /home/user/Escritorio/tarea5/.git/
user@user-PC:~/Escritorio/tarea5$ git add .
user@user-PC:~/Escritorio/tarea5$ git status
En la rama master

No hay commits todavía

Cambios a ser confirmados:
(usa "git rm --cached <archivo>..." para sacar del área de stage)

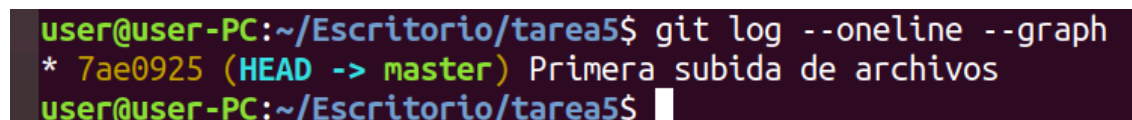
nuevo archivo: usuarios.xml

user@user-PC:~/Escritorio/tarea5$ git commit -a
[master (commit-raíz) 7ae0925] Primera subida de archivos
1 file changed, 30 insertions(+)
create mode 100644 usuarios.xml
user@user-PC:~/Escritorio/tarea5$ git status
En la rama master
nada para hacer commit, el árbol de trabajo está limpio
user@user-PC:~/Escritorio/tarea5$ git log
commit 7ae09251cc3ef646f542033d2fe0eb9cf6d48e11 (HEAD -> master)
Author: angela <angela@gmail.com>
Date: Mon Apr 22 00:46:21 2019 +0200

Primera subida de archivos
user@user-PC:~/Escritorio/tarea5$
```

Utilizamos el comando '`git log`' para comprobar si ha funcionado correctamente. Para verificar también que están subido a la rama master, utilizamos la siguiente línea de comando:

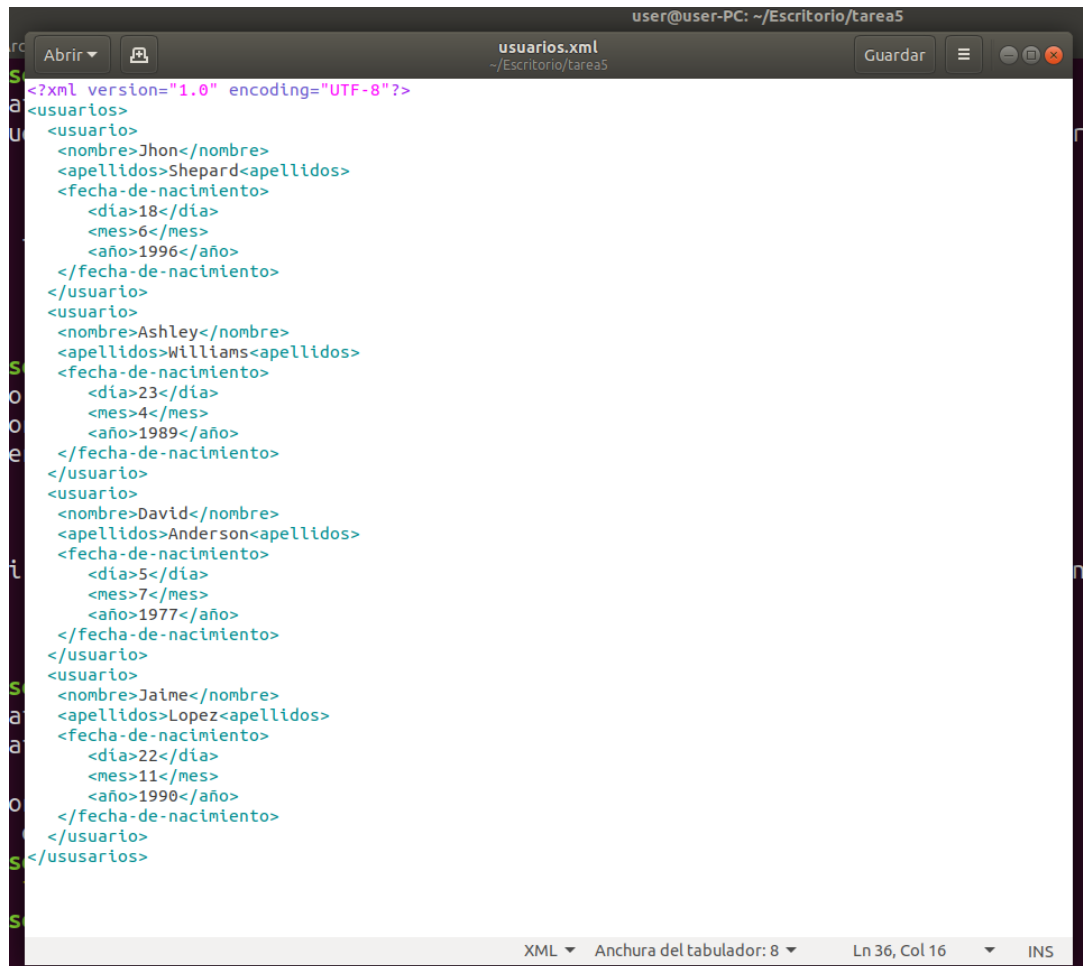
- `git log --oneline --graph`

A screenshot of a terminal window showing the output of the command 'git log --oneline --graph'. The output shows a single commit on the master branch, represented by an asterisk and the commit hash 7ae0925. The commit message is 'Primera subida de archivos'.

```
user@user-PC:~/Escritorio/tarea5$ git log --oneline --graph
* 7ae0925 (HEAD -> master) Primera subida de archivos
user@user-PC:~/Escritorio/tarea5$
```

Apartado 3

Abrimos el documento **usuarios.xml** y agregamos un nuevo usuario.



```
<?xml version="1.0" encoding="UTF-8"?>
<usuarios>
  <usuario>
    <nombre>Jhon</nombre>
    <apellidos>Shepard</apellidos>
    <fecha-de-nacimiento>
      <día>18</día>
      <mes>6</mes>
      <año>1996</año>
    </fecha-de-nacimiento>
  </usuario>
  <usuario>
    <nombre>Ashley</nombre>
    <apellidos>Williams</apellidos>
    <fecha-de-nacimiento>
      <día>23</día>
      <mes>4</mes>
      <año>1989</año>
    </fecha-de-nacimiento>
  </usuario>
  <usuario>
    <nombre>David</nombre>
    <apellidos>Anderson</apellidos>
    <fecha-de-nacimiento>
      <día>5</día>
      <mes>7</mes>
      <año>1977</año>
    </fecha-de-nacimiento>
  </usuario>
  <usuario>
    <nombre>Jaime</nombre>
    <apellidos>Lopez</apellidos>
    <fecha-de-nacimiento>
      <día>22</día>
      <mes>11</mes>
      <año>1990</año>
    </fecha-de-nacimiento>
  </usuario>
</usuarios>
```

Y a continuación subimos el documento actualizado con los comandos que ya hemos utilizado anteriormente.

- git add usuarios.xml
- git commit

```

user@user-PC: ~/Escritorio/tarea5
Archivo Editar Ver Buscar Terminal Ayuda
user@user-PC:~/Escritorio/tarea5$ git status
En la rama master
Cambios no rastreados para el commit:
(usa "git add <archivo>..." para actualizar lo que será confirmado)
(usa "git checkout -- <archivo>..." para descartar los cambios en el directorio de trabajo)

    modificado:    usuarios.xml

sin cambios agregados al commit (usa "git add" y/o "git commit -a")
user@user-PC:~/Escritorio/tarea5$ git add usuarios.xml
user@user-PC:~/Escritorio/tarea5$ git commit -m "nueva actualizacion"
[master 68e0d88] nueva actualizacion
1 file changed, 10 insertions(+), 1 deletion(-)
user@user-PC:~/Escritorio/tarea5$ git status
En la rama master
nada para hacer commit, el árbol de trabajo esta limpio
user@user-PC:~/Escritorio/tarea5$ git log --oneline --graph
* 68e0d88 (HEAD -> master) nueva actualizacion
* 7ae0925 Primera subida de archivos
user@user-PC:~/Escritorio/tarea5$

```

Apartado 4

Volvemos a modificar el archivo **usuarios.xml** y creamos el documento **privado.txt** :

```

user@user-PC: ~/Escritorio/tarea5
Archivo Editar Ver Buscar Terminal Ayuda
user@user-PC:~/Escritorio/tarea5$ nano privado.txt
user@user-PC:~/Escritorio/tarea5$ nano usuarios.xml
user@user-PC:~/Escritorio/tarea5$ git status
En la rama master
Cambios no rastreados para el commit:
(usa "git add <archivo>..." para actualizar lo que será confirmado)
(usa "git checkout -- <archivo>..." para descartar los cambios en el directorio de trabajo)

    modificado:    usuarios.xml

Archivos sin seguimiento:
(usa "git add <archivo>..." para incluirlo a lo que se será confirmado)

    privado.txt

sin cambios agregados al commit (usa "git add" y/o "git commit -a")
user@user-PC:~/Escritorio/tarea5$

```

Ahora, si lo que queremos es que **GIT** no agregue el documento **privado.txt** a la rama master tendremos que crear el archivo **'.gitignore'** y escribimos dentro el nombre de todos los archivos que queremos que ignore.

```

user@user-PC:~/Escritorio/tarea5$ git status
En la rama master
Cambios no rastreados para el commit:
(usa "git add <archivo>..." para actualizar lo que será confirmado)
(usa "git checkout -- <archivo>..." para descartar los cambios en el directorio de trabajo)

    modificado:    usuarios.xml

Archivos sin seguimiento:
(usa "git add <archivo>..." para incluirlo a lo que se será confirmado)

    .gitignore

sin cambios agregados al commit (usa "git add" y/o "git commit -a")
user@user-PC:~/Escritorio/tarea5$ git add .
user@user-PC:~/Escritorio/tarea5$ git commit -m "gitignore"
[master b5027e3] gitignore
2 files changed, 6 insertions(+), 5 deletions(-)
create mode 100644 .gitignore
user@user-PC:~/Escritorio/tarea5$ git log --oneline --graph
* b5027e3 (HEAD -> master) gitignore
* 68e0d88 nueva actualizacion
* 7ae0925 Primera subida de archivos
user@user-PC:~/Escritorio/tarea5$

```

Cuando utilizamos el comando '*git status*' podemos ver que ya no rastrea el documento **privado.txt** sino **.gitignore** que contendrá la lista de los archivos que queramos que ignore. Lo siguiente será agregarlo todo a la rama principal.

Ahora volveremos al primer commit, para ello, utilizamos el siguiente comando:

➤ `git checkout <hash del commit>`

```

commit b5027e366aa8a3114c21ee21f595fed6dce9e2c4 (HEAD -> master)
Author: angela <angela@gmail.com>
Date:   Wed Apr 24 00:51:03 2019 +0200

    gitignore

commit 68e0d886c64531ae9175e5391ff35701ac48f320
Author: angela <angela@gmail.com>
Date:   Mon Apr 22 01:39:51 2019 +0200

    nueva actualizacion

commit 7ae09251cc3ef646f542033d2fe0eb9cf6d48e11
Author: angela <angela@gmail.com>
Date:   Mon Apr 22 00:46:21 2019 +0200

    Primera subida de archivos
user@user-PC:~/Escritorio/tarea5$ git checkout 7ae09251cc3ef646f542033d2fe0eb9cf6d48e11
Nota: actualizando el árbol de trabajo '7ae09251cc3ef646f542033d2fe0eb9cf6d48e11'.

Te encuentras en estado 'detached HEAD'. Puedes revisar por aquí, hacer
cambios experimentales y confirmarlos, y puedes descartar cualquier
commit que hayas hecho en este estado sin impactar a tu rama realizando
otro checkout.

Si quieres crear una nueva rama para mantener los commits que has creado,
puedes hacerlo (ahora o después) usando -b con el comando checkout. Ejemplo:

    git checkout -b <nombre-de-nueva-rama>

HEAD está ahora en 7ae0925 Primera subida de archivos
user@user-PC:~/Escritorio/tarea5$

```

Y creamos una nueva version con la siguiente línea de commando:

➤ `git tag v1.0`

```
user@user-PC:~/Escritorio/tarea5$ git log
commit 7ae09251cc3ef646f542033d2fe0eb9cf6d48e11 (HEAD, tag: v1.0)
Author: angela <angela@gmail.com>
Date: Mon Apr 22 00:46:21 2019 +0200
```

Primera subida de archivos

```
user@user-PC:~/Escritorio/tarea5$
```

Apartado 5

Desde la página de GitHub creamos un nuevo Proyecto, para agregar el contenido de nuestro repositorio local al proyecto utilizamos el siguiente commando:

```
user@user-PC:~/Escritorio/tarea5$ git remote add origin https://github.com/AngelaDAW/Perez_Alvarez_Angela_TareaLM05.git
user@user-PC:~/Escritorio/tarea5$ git push origin master
Username for 'https://github.com': AngelaDAW
Password for 'https://AngelaDAW@github.com':
Contando objetos: 10, listo.
Comprimiendo objetos: 100% (7/7), listo.
Escribiendo objetos: 100% (10/10), 1.01 KiB | 516.00 KiB/s, listo.
Total 10 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/AngelaDAW/Perez_Alvarez_Angela_TareaLM05.git
 * [new branch]      master -> master
user@user-PC:~/Escritorio/tarea5$
```

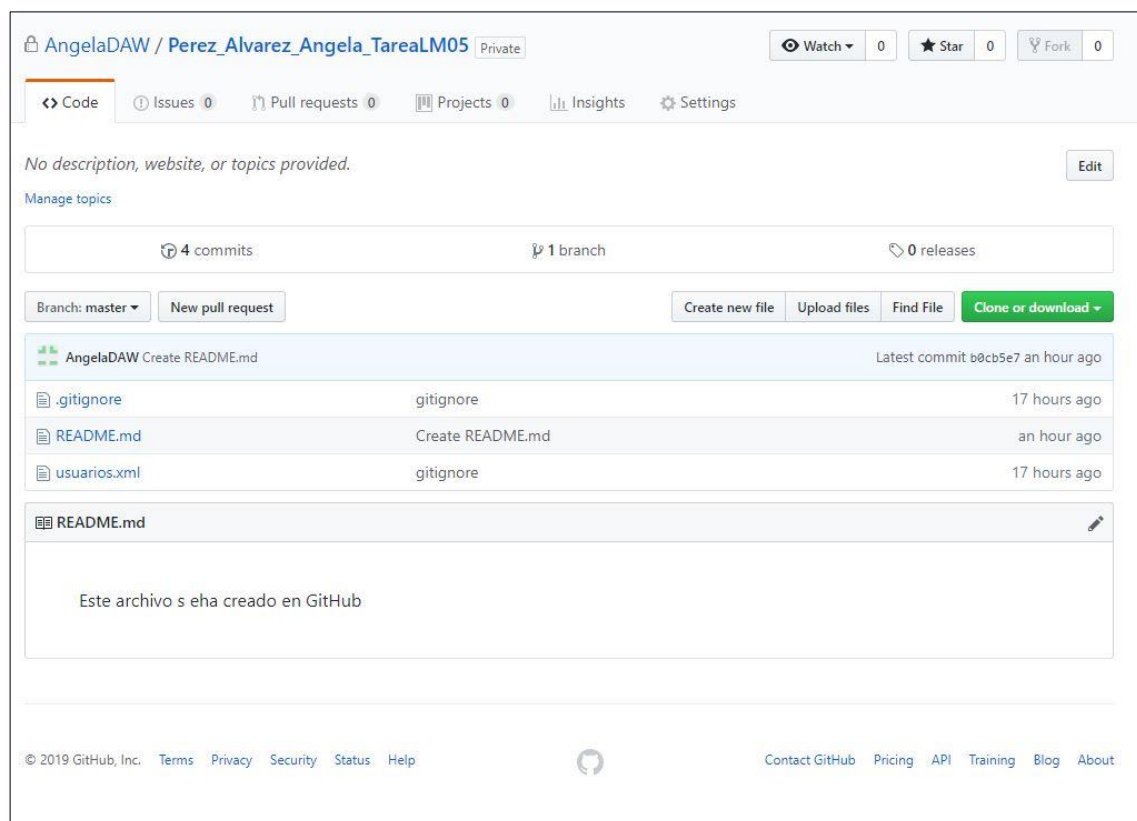
El siguiente paso es crear el archivo **README.md** desde GitHub, nos dirigimos a nuestro proyecto previamente creado y hacemos clic en el botón *Create new file*, luego para guardarlo solo tendremos que pulsar *Commit new file*.



Ahora lo traemos todo al repositorio local, para ello, utilizamos la siguiente línea de comando.

```
user@user-PC:~/Escritorio/tarea5$ git pull https://github.com/AngelaDAW/Perez_Alvarez_Angela_TareaLM05.git master
Username for 'https://github.com': AngelaDAW
Password for 'https://AngelaDAW@github.com':
Desde https://github.com/AngelaDAW/Perez_Alvarez_Angela_TareaLM05
* branch      master      -> FETCH_HEAD
Merge made by the 'recursive' strategy.
 README.md    | 1 +
 usuarios.xml | 21 ++++++++-----
 2 files changed, 16 insertions(+), 6 deletions(-)
 create mode 100644 README.md
user@user-PC:~/Escritorio/tarea5$ ls
privado.txt README.md usuarios.xml
user@user-PC:~/Escritorio/tarea5$
```

Y en el repositorio remoto.



Apartado 6

Lo primero que hacemos es crear la nueva rama y posicionarnos en ella:

➤ `git checkout -b variante`

Desde la nueva rama, editamos el archivo **usuarios.xml** y creamos uno nuevo llamado **grupos.xml**:

```
* 9c42329 (HEAD -> variante) Merge branch 'master' of https://github.com/AngelaDAW/Perez_Alvarez_Angela_TareaLM05 into HEAD
|
| * b0cb5e7 (origin/master) Create README.md
| * b5027e3 (master) gitignore
| * 68e0d88 nueva actualizacion
| * 91bd744 actualizacion
|/
* 7ae0925 (tag: v1.0) Primera subida de archivos
user@user-PC:~/Escritorio/tarea5$ git status
En la rama variante
Cambios a ser confirmados:
  nuevo archivo:  grupos.xml
  modificado:     usuarios.xml

user@user-PC:~/Escritorio/tarea5$ git commit -m "rama variante"
[variante 4fd4f8f] rama variante
 2 files changed, 8 insertions(+), 3 deletions(-)
 create mode 100644 grupos.xml
user@user-PC:~/Escritorio/tarea5$ git log --oneline --graph
* 4fd4f8f (HEAD -> variante) rama variante
* 9c42329 Merge branch 'master' of https://github.com/AngelaDAW/Perez_Alvarez_Angela_TareaLM05 into HEAD
|
| * b0cb5e7 (origin/master) Create README.md
| * b5027e3 (master) gitignore
| * 68e0d88 nueva actualizacion
| * 91bd744 actualizacion
|/
* 7ae0925 (tag: v1.0) Primera subida de archivos
user@user-PC:~/Escritorio/tarea5$
```

Para fusionar, tendremos que volver a la rama master y utilizar el siguiente comando:

➤ `git merge variante`

Y para finalizar solo nos queda subirlo al repositorio remoto.

```
Primera subida de archivos
user@user-PC:~/Escritorio/tarea5$ git merge variante
Actualizando b5027e3..4fd4f8f
Fast-forward
 README.md | 1 +
 grupos.xml | 5 +++++
 usuarios.xml | 6 +++--
 3 files changed, 9 insertions(+), 3 deletions(-)
 create mode 100644 README.md
 create mode 100644 grupos.xml
user@user-PC:~/Escritorio/tarea5$ git push origin master
Username for 'https://github.com': AngelaDAW
Password for 'https://AngelaDAW@github.com':
Contando objetos: 7, listo.
Comprimiendo objetos: 100% (7/7), listo.
Escribiendo objetos: 100% (7/7), 930 bytes | 465.00 KiB/s, listo.
Total 7 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/AngelaDAW/Perez_Alvarez_Angela_TareaLM05.git
 b0cb5e7..4fd4f8f master -> master
user@user-PC:~/Escritorio/tarea5$
```

