

Tutorial GIT-Sistema de control de versiones

Un Sistema de Control de Versiones (SCV) es una aplicación que permite gestionar los cambios que se realizan sobre los elementos de un proyecto o repositorio

Git es un sistema de control de versiones de código abierto. Git tiene una arquitectura distribuida, lo que significa que en lugar de guardar todos los cambios de un proyecto en un único sitio, cada usuario contiene una copia del repositorio con el historial de cambios completo del proyecto.

En este tutorial se utiliza Bitbucket. Este es un servicio de alojamiento basado en web, para los proyectos que utilizan el sistema de control de versiones Git.

Crear cuenta gratuita Bitbucket

Lo primero que debemos hacer es crear una cuenta gratuita en Bitbucket (u otro servicio como GITHAB O GITLAB). Al ingresar crear un repositorio (“+”) para almacenar su proyecto.

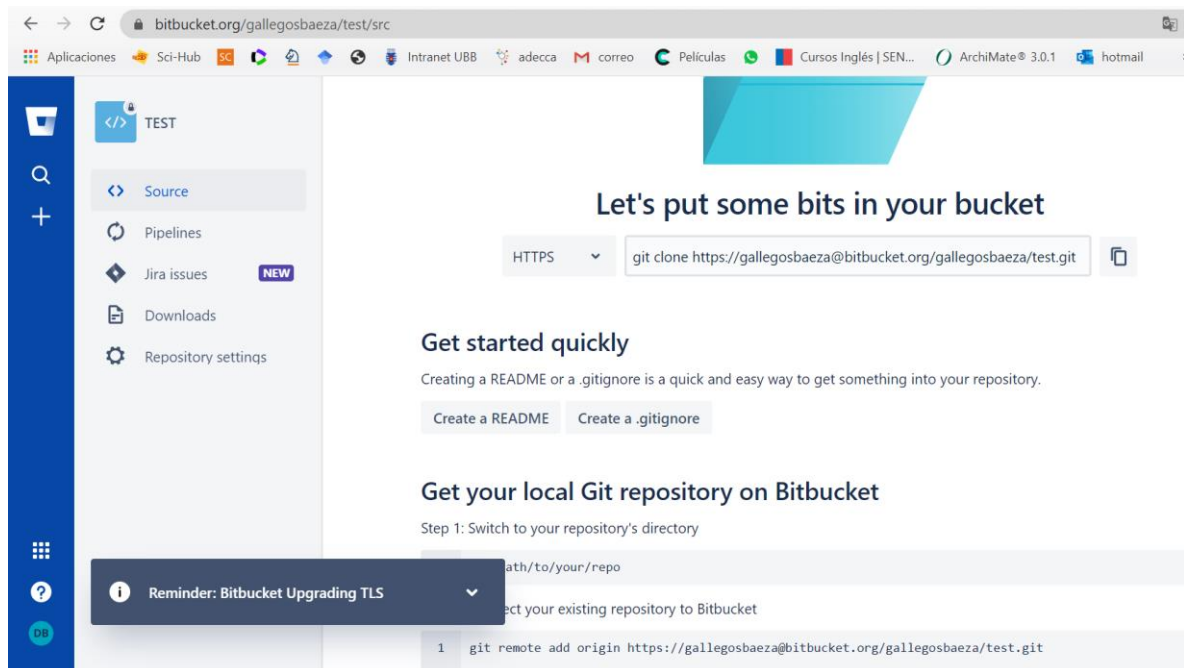


Figura 1: Proyecto en Bitbucket

Creación de un nuevo repositorio

- 1- En la consola vaya al directorio de su repositorio (Figura 2)

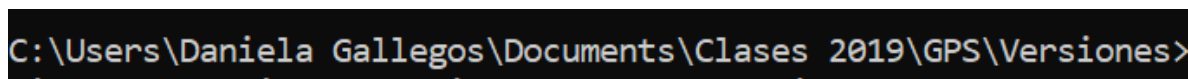


Figura 2: Directorio local

- 2- Use git init para iniciar el repositorio (Figura 3)

git init crea una nueva carpeta con el nombre del repositorio, que a su vez contiene otra carpeta oculta llamada .git que contiene la base de datos donde se registran los cambios en el repositorio.

```
C:\Users\Daniela Gallegos\Documents\Clases 2019\GPS\Versiones>git init
Initialized empty Git repository in C:/Users/Daniela Gallegos/Documents/Clases 2019/GPS/Versiones/.git/

C:\Users\Daniela Gallegos\Documents\Clases 2019\GPS\Versiones>git remote add origin https://gallegosbaeza@bitbucket.org/gallegosbaeza/test.git
```

Figura 3: comando git init

Para conectar tu repositorio a Bitbucket (Figura 2) usar el link que se muestra en la Figura 1.

Subir cambios al repositorio remoto

Todos los cambios que realice en su repositorio permanecen de forma local (solo en su computador). Para subir los cambios al repositorio remoto debemos añadir los cambios (add -A), guardarlos (commit) y luego subirlos (push). Si está trabajando con otros desarrolladores en una misma rama debe descargar los cambios antes de subir los suyos (pull).

Git add -A añade todos los cambios de todos los ficheros no guardados aún en la zona de intercambio temporal.

git commit -m "mensaje" confirma todos los cambios de la zona de intercambio temporal añadiéndolos al repositorio y creando una nueva versión del proyecto.

git push sube al repositorio remoto los cambios de la rama en el repositorio local.

```
C:\Users\Daniela Gallegos\Documents\Clases 2019\GPS\Versiones>git add -A

C:\Users\Daniela Gallegos\Documents\Clases 2019\GPS\Versiones>git commit -m "commit inicial"
[master (root-commit) 3f79b0b] commit inicial
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Prueba.txt

C:\Users\Daniela Gallegos\Documents\Clases 2019\GPS\Versiones>git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 215 bytes | 215.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://bitbucket.org/gallegosbaeza/test.git
 * [new branch]      master -> master
```

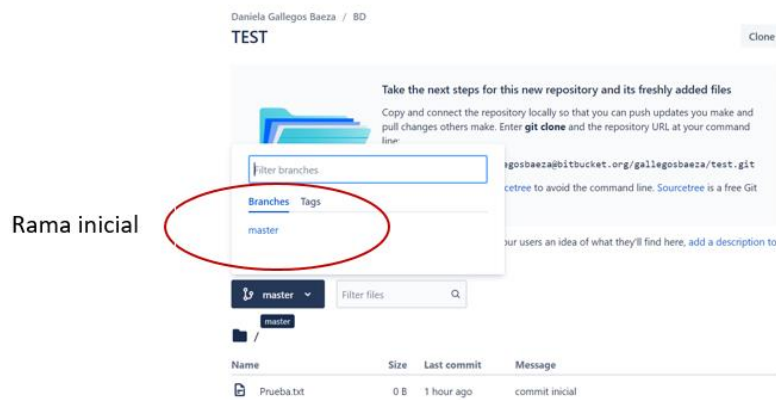


Figura 4: Subir cambios

Ramas

Inicialmente cualquier repositorio tiene una única rama llamada master donde se van sucediendo todos los commits de manera lineal.

Una de las características más útiles de Git es que permite la creación de ramas para trabajar en distintas versiones de un proyecto a la vez.

git checkout -b crea una nueva rama con el nombre y la activa, es decir, que pasa a apuntar al último commit de esta rama. Este comando es equivalente aplicar los comandos git branch y después git checkout.

```
C:\Users\Daniela Gallegos\Documents\Clases 2019\GPS\Versiones>git checkout -b develop
Switched to a new branch 'develop'

C:\Users\Daniela Gallegos\Documents\Clases 2019\GPS\Versiones>git push origin develop
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create pull request for develop:
remote:   https://bitbucket.org/gallegosbaeza/test/pull-requests/new?source=develop&t=1
remote:
To https://bitbucket.org/gallegosbaeza/test.git
 * [new branch]      develop -> develop
```

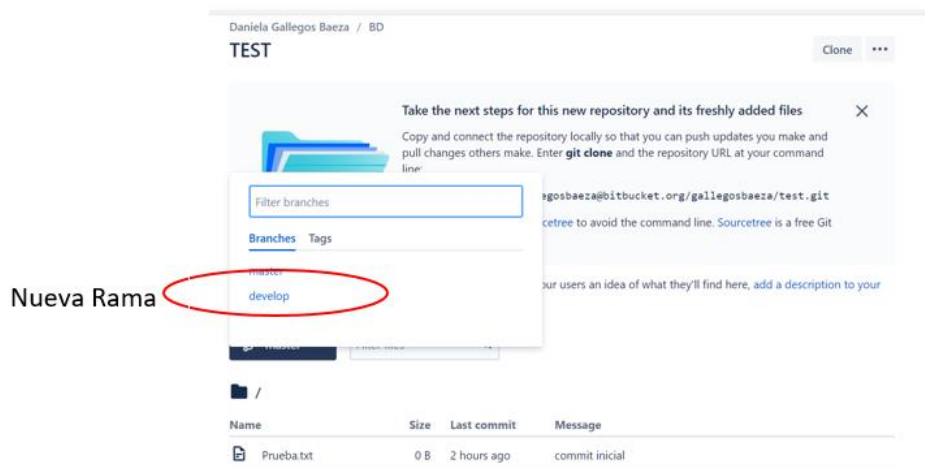


Figura 5: Crear ramas

Fusión de ramas

Una vez que se han realizado cambios en los archivos de una rama y los queremos mezclar con el proyecto principal debemos guardar nuestros cambios (add y commit), descargar los del repositorio principal (pull) y luego hacer la fusión (merge) (Figura 7). Con el comando git status podemos ver el estado actual de nuestro repositorio local (Figura 6).

```

C:\Users\Daniela Gallegos\Documents\Clases 2019\GPS\Versiones>git status
On branch develop
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:   Prueba.txt

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

```

Figura 6: Verificar cambios

```

C:\Users\Daniela Gallegos\Documents\Clases 2019\GPS\Versiones>git status
On branch develop
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:   Prueba.txt

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

C:\Users\Daniela Gallegos\Documents\Clases 2019\GPS\Versiones>git add -A

C:\Users\Daniela Gallegos\Documents\Clases 2019\GPS\Versiones>git commit -m "act repo"
[develop c939892] act repo
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Daniela Gallegos\Documents\Clases 2019\GPS\Versiones>git push origin develop
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 271 bytes | 271.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create pull request for develop:
remote:   https://bitbucket.org/gallegosbaeza/test/pull-requests/new?source=develop&t=1
remote:
To https://bitbucket.org/gallegosbaeza/test.git
   6d6aef8..c939892  develop -> develop

C:\Users\Daniela Gallegos\Documents\Clases 2019\GPS\Versiones>git checkout master
Switched to branch 'master'

C:\Users\Daniela Gallegos\Documents\Clases 2019\GPS\Versiones>git merge develop
Updating 6d6aef8..c939892
Fast-forward
 Prueba.txt | 2 +
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Daniela Gallegos\Documents\Clases 2019\GPS\Versiones>git push origin master
Total 0 (delta 0), reused 0 (delta 0)
To https://bitbucket.org/gallegosbaeza/test.git
   3f79b0b..c939892  master -> master

```

Figura 7: Fusión de ramas

El comando git merge integra los cambios de la rama actual con otra y el comando git checkout <rama> permite cambiarnos a otra rama.

En ocasiones si dos desarrolladores trabajaron en la misma rama pueden producir conflictos. Para fusionar dos ramas es necesario que no haya conflictos entre los cambios realizados a las dos versiones del proyecto. Si en ambas versiones se han hecho cambios sobre la misma, entonces se produce un conflicto y es necesario resolverlo antes de poder fusionar las ramas. La resolución debe hacerse manualmente observando los cambios que interfieren y decidiendo cuales deben prevalecer.