

git

documento_final

documento_final2

documento_final2_ahora_si

documento_final2_ahora_si_final

documento_final2_ahora_si_final_de_verdad

git

Sistema de control de versiones que permite trabajar simultáneamente a grupos de personas, permite separar el trabajo por ramas para implementar de manera paralela distintos sectores del código.

<https://git-scm.com/downloads>

github

- creen usuario en [GitHub.com](https://github.com)

- configuren en la cmd/terminal el usuario

`git config --global user.email "su_correo_registrado_en_github"`

github

- creen un nuevo repositorio desde [GitHub.com](https://github.com)

- clonen su repositorio

```
git clone https://github.com/su_usuario/su_repo.git
```

git

- haga modificaciones en el la carpeta local de su repositorio
- agregue esos cambios y cree un commit

`git add -A`

`git commit -m "mensaje que representa mi commit"`

git

- mande su commit a el repositorio en GitHub

git push

- revise que los cambios hayan quedado en [GitHub.com](https://github.com)

git

- para traer la información de si hay cambios en el repositorio en GitHub

`git fetch`

- Para traerse los cambios del repositorio en GitHub

`git pull`

git

- Crearemos una rama a partir de la rama en la que nos encontramos

```
git branch mi_rama
```

- Nos cambiaremos a la rama recién creada

```
git checkout mi_rama
```

- Le diremos al repositorio en GitHub que la cree

```
git push --set-upstream origin mi_rama
```


git

- haremos cambios en la rama y los mandaremos con push
- desde GitHub crearemos un pull-request para ingresar los cambios de mi_rama a la rama de la que salimos, en este caso master
- Resolveremos conflictos si es que tenemos y hacemos un merge para que los cambios de mi_rama queden en master

git

- finalmente eliminaremos la rama del servidor de GitHub y de nuestro repositorio local

```
git checkout master
```

```
git push origin -d mi_rama
```

```
git branch -d mi_rama
```