

git-scm.com  
  
abrir la consola de git-bash  
# git config –global user.name “name”  
# git config –global .user.emailemail  
# git config –global color.ui true  
  
-----------------

**Crear un proyecto**  
# ir desde consola a la carpeta del proyecto  
# git init (solo se usa una vez, al iniciar git en el proyecto)  
# git status (estado del proyecto, los archivos modificados etc)  
# git add archivo o –A (agrega todos)  
# git commit –m “mensaje” para subir los cambios que hicimos con un mensaje para identificar  
# git log (nos indifica todos los comits que se han creado)  
# git checkout (para viajar en el tiempo) se copia el código de git log al que se quiere viajar / o poner master que te mueve al ultimo comit

# git diff (indica que cambio en cada archivo)  
  
# git reset viaja en el tiempo pero mata los commit | - - soft (no toca el código) | -- mixed no toca staging área ni el working área | -- hard borra el código borra todo

**Ramas y fusiones  
  
Todo lo que se haga dentro de una rama no se modifica en las otras**

# git branch | muestras las ramas   
# git branch “name” | crea una rama  
# git checkout name rama | para moverse a una rama  
# git checkout –b name | crea la rama y a la vez te mueva a ella  
# git branch –D name | para borrar una rama  
  
**Fusionar las ramas**  
Debemos movernos a la rama principal para fusionarla con otra  
# git merge rama | el nombre de la rama a fusionar con la principal

Rama master: es la rama principal

gitHub  
  
# git clone url | para clonar un proyecto a nuestro local   
# git remote –v | para ver en que proyecto remoto estoy  
# git remote remove origin | para eliminar la conexión con ese repositorio  
# git pull origin master | bajar cambios del repositorio remoto  
# git push origin master | pasar los archivos, cambios etc del proyecto al repositorio remoto

**Poner nuestro repositorio en github**  
  
1) crear repositorio en gitHub  
2) $ git remote add origin url (https) | vincula nuestro proyecto local con el proyecto remoto   
3) $ git push origin master | pasar los archivos, cambios etc del proyecto al repositorio remoto  
  
Issues| continuar mejorar o solucionar cosas de un proyecto  
 agregar un issues  
 poner nombre de la tarea y descripción a hacer por ejemplo  
 A estos issues se le puede dar estilos  
 Ver link de masteringmarkdown

Milestones: son un grupo de issues que se aplican a proyecto por un tiempo  
  
Labels: diferentes etiquetas para organizar los problemas, se aplican a los issues  
  
  
Workflows

Trabajo en equipo

* Crear una nueva organización
* Poner personas que forman parte del proyecto / organización
* Cambiar privilegios a los usuarios
* Cada uno se debe conectar al repositorio remoto

Si uno sube un cambio al repositorio y otra persona quiere subir también sus cambios, git no lo va a dejar bajar los cambios hasta que actualice su repositorio local, bajando los cambios actuales y ahí recién puede subir sus cambios.

# gitfetchorigin | bajar los cambios que estén en el repositorio remoto a origin master  
# gitmergeorigin/master | baja los cambios de origin master al local  
  
Si hay conflictos de mismos código, git te lo muestra como error en tu editor de desarrollo  
Simplemente se corrige el error en el editor y se vuelven a subir los cambios con un commit nuevo etc.

GitHubpages  
  
Para crear portafolios blogs, todo tipo de código frontend  
  
nombredeusuariogit.github.io – este es el repositorio a crear y dominio  
luego clonar este repositorio y todo el código que tenga se va a ver en el dominio  
leugo crear una rama que se llama $ gitbranchgh-pages

**Deployment con git**

Todos los commits o cambios del repositorio local además de mandarlo a Github se van a mandar al servidor  
  
Llave SSH, a diferencia de HTTPS, no te pide un usuario y contraseña por cada commit  
  
**Configurar llave**  
$ ssh-keygen  
se le da enter, te pide una contraseña pones una o no y se genera la llave.  
# cd .ssh  
# cat id\_rsa.pub / copiar el código e ir a la página de github y pegar el código en sshkeys para que sepa que es nuestra computadora para no tener que poner el usuario y contraseña.

touch.gitignore crea una rchivo en el repositorio local donde se van a poner todos los archivos a ignorar  
  
**Configurar Servidor a través del ssh**En el hosting configurar ssh con el key  
conectarse al servidor  
$ sshroot@dominio / ip  
crear llave en el servidor   
cd .ssh  
cat id\_rsa.pub  
Agregar la nueva llave ssh en github  
  
ir a la carpeta del proyecto dentro del servidor  
$ gitpullorigin rama / pull: (fetch + merge ) / esto es para bajar los cambios