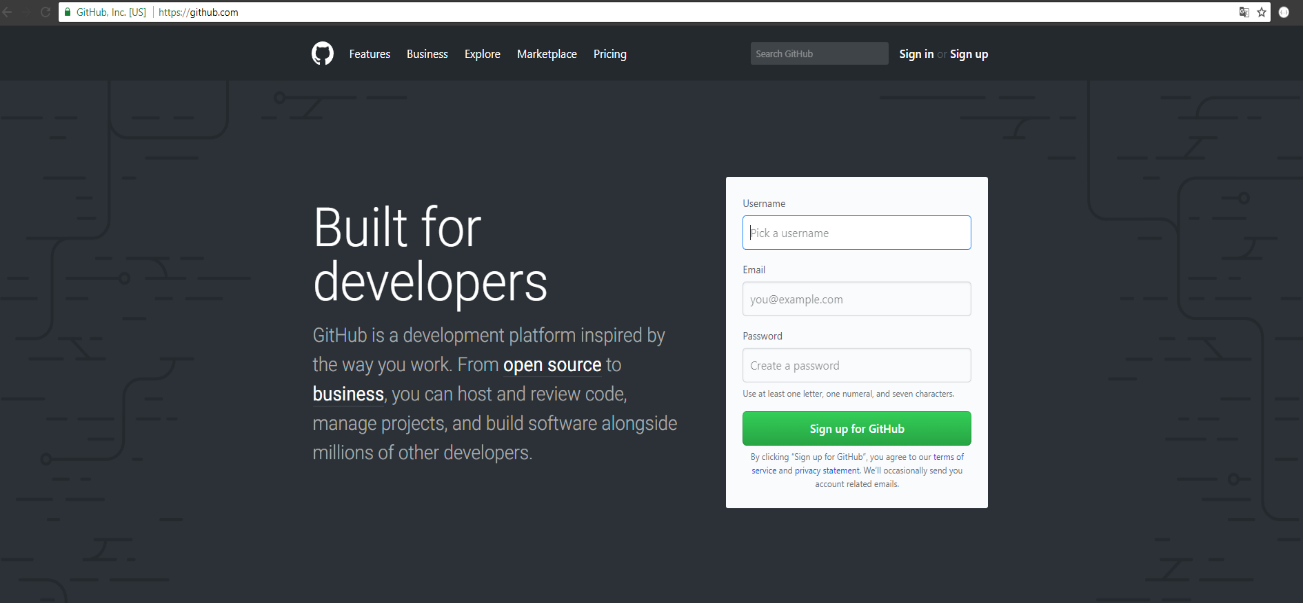
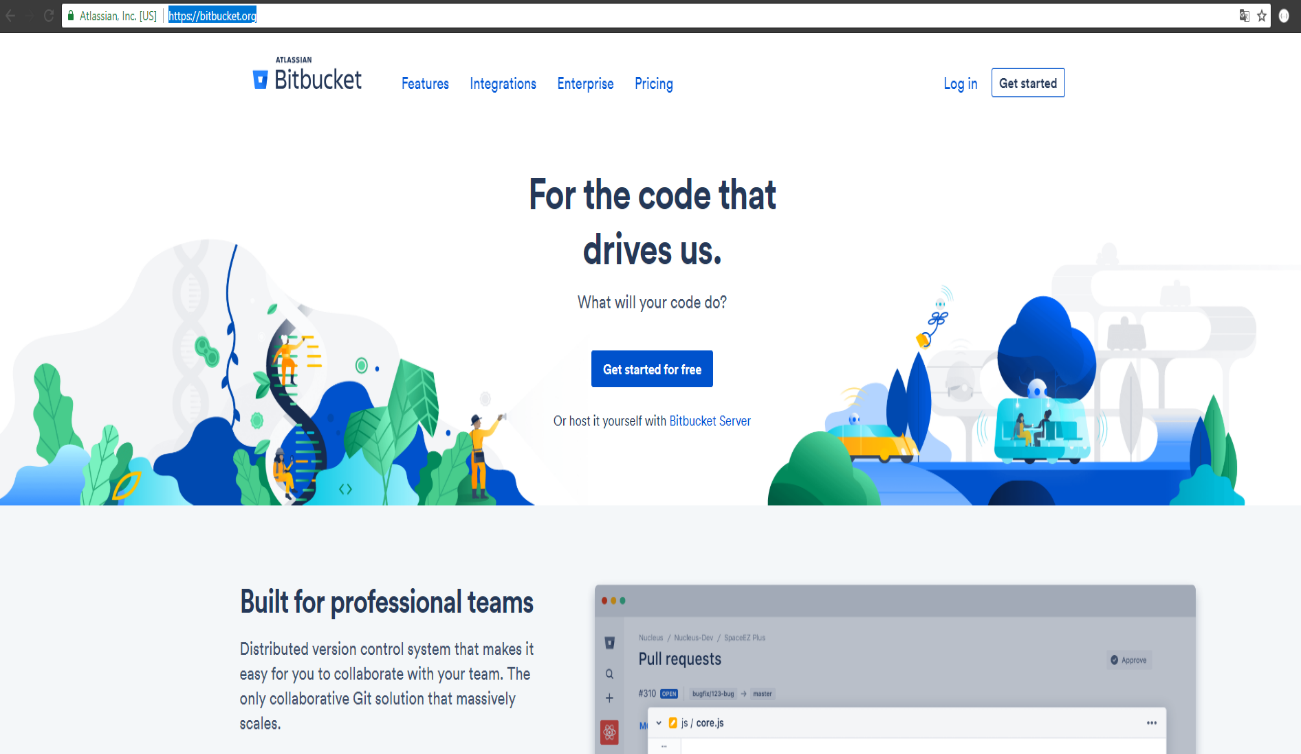
****

****

[**https://github.com/**](https://github.com/)

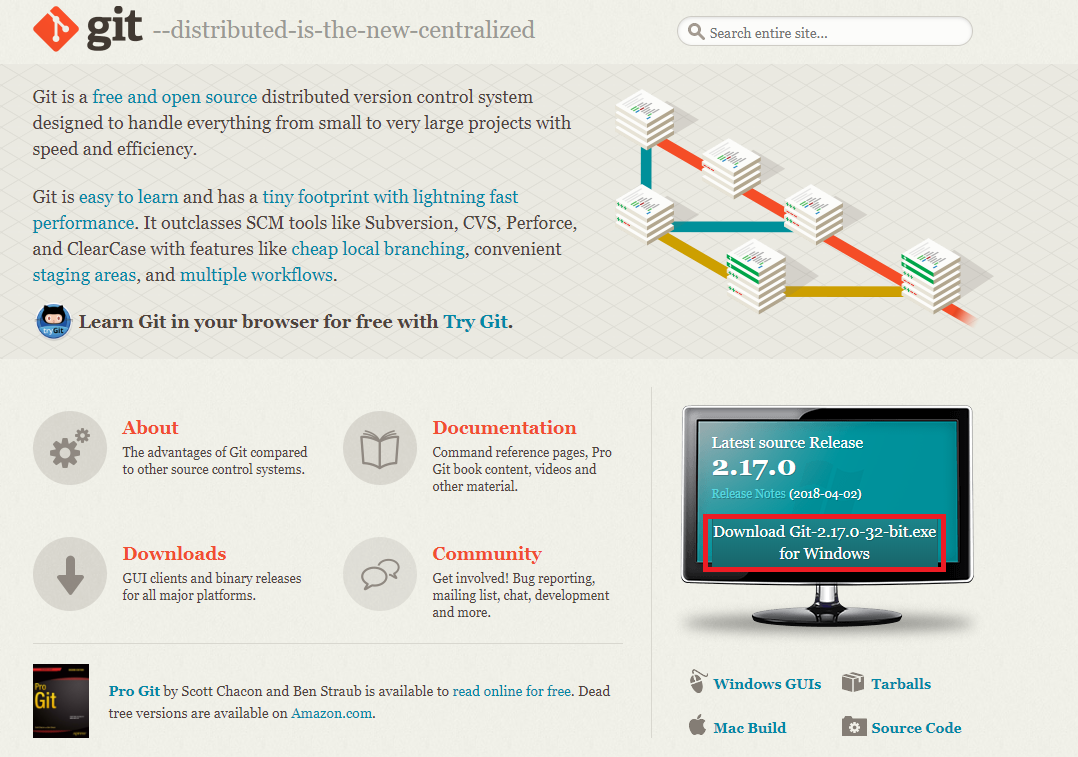
****

[**https://bitbucket.org/**](https://bitbucket.org/)

**¿Qué necesitas?**

* Solamente tener GIT instalado en tu sistema.

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



**Comandos básicos de GIT**

# GIT HELP

* git help
* git help <verbo> x ejemplo add

# GIT CONFIG

Es usado para establecer una configuración específica de usuario, como sería el caso del email, un algoritmo preferido para diff, nombre de usuario y tipo de formato, etc

* git config --global user.name "fname lname"
* git config --global user.email [example@gmail.com](mailto:example@gmail.com)
* git config --global user.password "secret"

Por base de repositorio, también puede editar .git/config manualmente en su lugar.

* git config --list:Ejecute para verificar el nombre de usuario y el correo electrónico actuales en su repositorio local.

**Nota:** Si la consola queda en : al final, salir con “q”

# GIT INIT

Este comando se usa para crear un nuevo repertorio vacío de GIT

* git init

GIT ADD

Este comando puede ser usado para agregar archivos al index

* git add .

# GIT CLONE

Permite clonar un repositorio local desde el servidor de GIT

* git clone <http://10.90.1.27:3000/GENDARMERIANACIONAL/GLOSARIO.git>

# GIT COMMIT

El comando commit es usado para cambiar a la cabecera. Ten en cuenta que cualquier cambio comprometido no afectara al repertorio remoto ES TODO LOCAL.

* git commit –m “Mensaje del commit”

GIT STATUS

Este comando muestra la lista de los archivos que se han cambiado junto con los archivos que están por ser añadidos o comprometidos.

* git status

# GIT PUSH

Este es uno de los comandos más básicos. Un simple push envía los cambios que se han hecho en la rama principal de los repertorios remotos que están asociados con el directorio que está trabajando. Por ejemplo:

* git push origin master

# GIT PULL

Para poder fusionar todos los cambios que estaban en el server en el repositorio local que estamos trabajando trabajando.

* git pull

# GIT REMOTE

El comando git se usa para conectar a un repositorio remoto.

* git remote add [nombre] [url]

Ejemplo:

git remote add origin <http://10.90.1.27:3000/aplicacion-name/repo-name.git>

El siguiente comando muestra los repositorios remotos que están configurados actualmente:

* git remote –v

# GIT BRANCH

Este comando se usa para listar, crear o borrar ramas. Para listar todas las ramas se usa:

* git branch featureA
* git checkout -b feature
* git branch -d <branch-name>

# GIT CHECKOUT

El comando checkout se puede usar para crear ramas o cambiar entre ellas. Por ejemplo, el siguiente comando crea una nueva y se cambia a ella:

* git checkout -b <banch-name>

Para cambiar de una rama a otra solo usa:

* git checkout <branch-name>

# GIT MERGE

Este comando se usa para fusionar una rama con otra rama activa:

Nos situamos en la rama que queremos actualizar por ejemplo la rama master

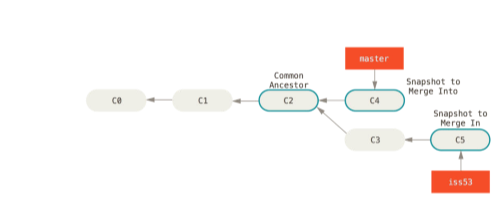
1. git checkout master

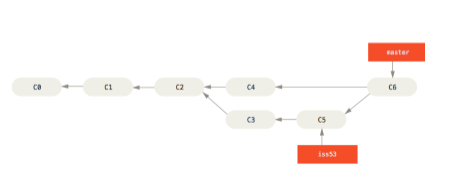
Desde master traemos todos los cambios de la rama hotfix

1. git merge hotfix

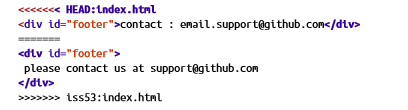


Antes del Merge (master en C4) y iss53(hotfix) (C5)



Luego del merge junta ambas ramas en (C6)

Pueden surgir conflictos en las fusiones y se verán de la siguiente manera

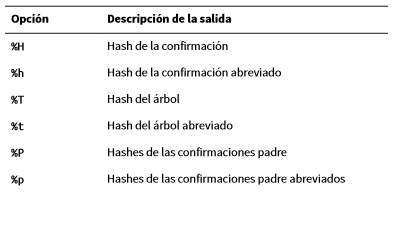


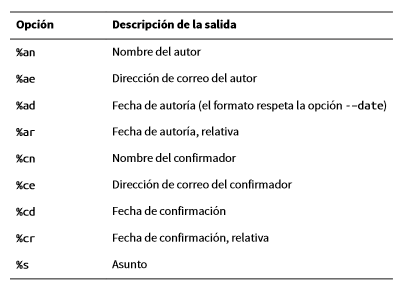
Como estamos parados en master nuestro HEAD es esa rama y la otra linea es lo que esta en la rama iss53 y nosotros deberemos decider que version debe quedar en la rama C6.

# GIT LOG

Ejecutar este comando muestra una lista de commits en una rama junto con todos los detalles. Por ejemplo:

* git log
* git log --stat
* git log --pretty=oneline
* git log --pretty=format:"%h, %ar : %s"





# Deshacer archivo agregado (GIT ADD . )



git reset HEAD CONTRIBUTING.md

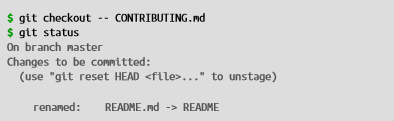


# Deshacer un Archivo Modificado

Queremos volver atras esos cambios de un archive modificado y no preparado.



Si ponemos git status vemos README



# GIT RESET

Para resetear el index y el directorio que está trabajando al último estado comprometido se usa este comando:

* git reset - -hard HEAD

Ejemplo

git reset --hard d545833fa849d41a4586cd14579d58a28822e182

git reset --soft HEAD@{1}

git commit -m “Revertir el commit d545833fa849d41a4586cd14579d58a28822e182”

**--soft => Deja los archivos modificados con el rollback.**

# FORZAR GIT PULL

* git fetch -–all

Si es la master

* git reset --hard origin/master

Si es alguna rama

* git reset --hard origin/<branch\_name>

### **Explicación:**

git fetch descarga lo último desde el control remoto sin intentar fusionar o volver a generar nada.

A continuación, git reset restablece la rama maestra a lo que acaba de recuperar. La –-hard opción cambia todos los archivos en su árbol de trabajo para que coincida con los archivos en origin/master

# FORZAR GIT PUSH

* git push --force origin B:C