**Diferencias entre Git y Github.**

Guit es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos del código fuente.

GitHub es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Utiliza el framework Ruby on Rails por GitHub, Inc. (anteriormente conocida como Logical Awesome).

Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc.

El codigo se almacena de forma  pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

**Comandos.**

git help <command>

git clone <uri> namedir # clona usando como nombre de directorio namedir.

git add <dir> # añade recursivamente todos los archivos del dir.

git diff --staged # compares staged changes with last commit

git commit -v # muestra el diff en el editor

git commit -a -m ” # automatically stage tracked files. No hace falta git add

git rm --cached <file or regexp> # Git no realiza un seguimiento del archivo, pero los deja en el directorio de trabajo. Útil cuando se olvida añadir archivos al .gitignore y ya hemos agregado dichos archivos al repositorio.

git rm <file> # borrarlos con git siempre.

git rm -f <file> # si ya está modificado y en el index.

git mv <file> <renamed\_file>

gitk # tcl/tk. Herramienta gráfica para git

git commit --amend # Modificar el mensaje del último commit

git reset HEAD <file> # to unstage

git checkout -- <file> # Descartar cambios en el directorio de trabajo.

**Añadir archivos:**

## AÑADIR ARCHIVOS

git add -i # interactive staggin

git add -p # crea patch

**Branch.**

## BRANCH

git branch <nombre-rama> # crea rama. Puntero al commit actual

git checkout <nombre-rama> # cambiar a la rama especificada.

git checkout -b <nombre-rama> # crea y cambia de rama

git merge <rama> # Mezcla la rama actual con <rama>

git branch -d <rama> # elimina la rama

git push origin --delete <branchName> # Elimina una rama del servidor

git mergetool # Herramienta gráfica para resolver conflictos

git branch # lista ramas

git branch -v # lista ramas mostrando último commit

git branch --merged # lista ramas que han sido mezcladas con la actual. Si no tienen un \*, pueden borrarse, ya que significa que se han incorporado los cambios en la rama actual.

git branch --no-merged # lista ramas que no han sido incorporadas a la actual.