giGit (pronunciado "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 éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago; te brinda herramientas muy útiles para el trabajo en equipo dentro de un proyecto. Además de eso, puedes contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls.

Realizar un fork es simplemente clonar un repositorio ajeno (genera una copia en tu cuenta), para eliminar algún bug o modificar cosas de él. Una vez realizadas tus modificaciones puedes enviar un pull al dueño del proyecto. Éste podrá analizar los cambios que has realizado fácilmente, y si considera interesante tu contribución, adjuntarlo con el repositorio original.

**Crear un repositorio GIT**

Debemos instalara git desde la consola de MSDOS

**C/Usuario>** install git

Para iniciar un repositorio GIT

**C/Usuario>** git init

Para ver el estado del git utilizamos

**C/Usuario>** git status

Descargar o clonar una copia de un directorio creado en git

**C/Usuario>** git clone htpps:// dirección

Para cambiarse de rama o repositorio

**C/Usuario>** git checkout draf ò master

Para subir el archivo al repositorio, (primero se debe subir el archivo a draf y luego a master

1. **C/Usuario>** git add “nombre del archivo”

Para incluir un comentario al archivo que se sube o actualiza

1. **C/Usuario>** git commit -m "comentario"

Una vez que se sube el archivo, primero al draft, entonces se publica a las demás ramas

1. **C/Usuario>** git push –u origin “master”

Para actualizar una rama o repositorio

**C/Usuario>** git pull

Para mostrar la rama actual

**C/Usuario>** git branch

Para fusionar ramas, situarnos en la rama que va a absorber los cambios

**C/Usuario>** git checkout master

Fusionar las ramas con el comando '# git merge <rama\_a\_fusionar>'

**C/Usuario>** git merge draft

**C/Usuario>**