

```
//Muestra todos los commits en el proyecto en la rama actual
git log

//Muestra todos los commits en el proyecto en la rama actual pero de manera compacta
git log --oneline

//Muestra todos los commits en el proyecto en la rama actual pero de manera compacta
//Proporciona información adicional acerca del nombre de las ramas y sus commits asociados
git log --oneline --decorate

//Hace lo anterior y
//--graph dibuja los commits con arte ASCII identificando las ramas que salen y se integran
//--all muestra la historia de todas las ramas
// el 30 indica el número de commits que quiero se muestren
git log --oneline --decorate --graph --all -30

//Me permite crear un alias en el que todos los parametros sean accesibles con la
//palabra que yo defina, en este caso es, 'sla'
git config --global alias.sla 'log --oneline --decorate --graph --all'
git sla

//Pretty me permite invocar una customización a la salida de git log
// %h - me devuelve el commit hash abreviado
// %an - me muestra el 'author name'
// %ar - Muestra la fecha de commit relativo (authored relative)
// %s - Commit subject ó los primeros 50 caracteres del commit
git log --pretty=format:'%h - %an [%ar] %s'
git log --pretty=format:'%C(yellow)%h%C(reset) - %an [%C(green)%ar%C(reset)] %s'

//Me ayuda a buscar una palabra en el commit message
//En este caso, 'update'
git log -E -i --grep 'update'

//Me permite crear un alias en el que todos los parametros en este caso, son accesibles
//mediante la palabra que yo defina, en este caso 'glog'
git config --global alias.glog 'log -E -i --grep'
git glog 'update'

//Busca la cadena de texto dentro del código en el proyecto
//En este caso buscara los archivos que contengan la palabra 'version'
git log -S version

//Me muestra la historia de commits de un archivo en particular
git log --oneline -- package.json

//Me muestra los cambios asociados a un archivo en especifico
//con Short Hash, autor y fecha
git blame package.json

//Me muestra el contenido del commit en especifico
//En este ejemplo el short hash es: 4adc9bad
git show 4adc9bad
```

Actividad 1:

Clonar el **proyecto público** de tu preferencia. En la clase, vimos el contenido de la biblioteca ‘conekta-ruby’

Lo cloné de la siguiente manera

```
git clone git@github.com:conekta/conekta-ruby.git
```

Ejecutar, en el proyecto de tu elección los comandos mostrados en la parte de arriba. Capturar pantalla de lo que te muestra la consola/terminal por cada comando y agregarlo a un documento PDF, Word ó Google Docs.

Recuerda añadir tu nombre al documento

Me encuentro disponible en Slack

O puedes mandar un email a: **fernando.ruiz@futureis.tech**

Actividad 2:

Crea un perfil en Github si no tienes uno. Más adelante vamos a personalizar tu perfil público de Github para que puedas crear un portafolio atractivo.

Esto lo puedes hacer en <https://github.com/>