

# Introducción a Git & Github

## Conceptos básicos e instalación

Daniel Jiménez M.

Universidad Nacional de Colombia

30 -09 -2020

# Agenda

- ¿Qué es Git?
- ¿Para qué sirve Git?
- ¿Cómo se instala Git?
- Editores de texto para Git
- Primeros comandos en Git
- Ejemplo de uso de Git

# ¿Qué es Git?

Según el portal de [git](#) se define como :

*“Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.”*

# ¿Qué es Git?

Una forma sencilla de entenderlo es definirlo como

*“Un sistema que guarda solo los cambios de los archivos, basado en el trabajo colaborativo”*

# ¿Qué es Git?

Por lo tanto:

- Es un sistema que permite que multiples personas puedan trabajar en un mismo proyecto;
- El sistema permite retroceder en el nivel de verionamiento tantas veces como se quiera;
- Es posible y sencillo identificar los errores de producción.

# ¿Qué es Git?

Git se usa para trabajar de manera local, cuando se quiere trabajar con varias personas se debe usar github (red social del código).



## ¿Para qué sirve Git?

Imaginé el siguiente ejemplo:

[illegible]

# ¿Para qué sirve Git?

Ahora debe hacer lo siguiente :

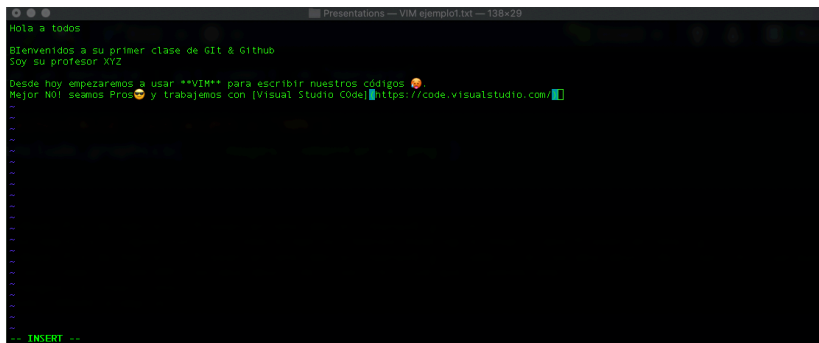
- Agregue el archivo al repositorio : `git init`
- `git add ejemplo1.txt`
- Agregue un mensaje : `git commit -m "mensaje"`

```
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git init
Initialized empty Git repository in /Users/danieljimenez/Desktop/Github-Class/Presentations/.git/
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git commit -m "estaba aburrido y escribi cualquier pendejada"
[master (root-commit) fadcd95] estaba aburrido y escribi cualquier pendejada
1 file changed, 6 insertions(+)
create mode 100644 ejemplo1.txt
```



# ¿Para qué sirve Git?

Ahora suponga que quiere hacer unos cambios que son importantes.



```
Presentations — VIM ejemplo1.txt — 138x29
Hola a todos
Bienvenidos a su primer clase de Git & Github
Soy su profesor XYZ

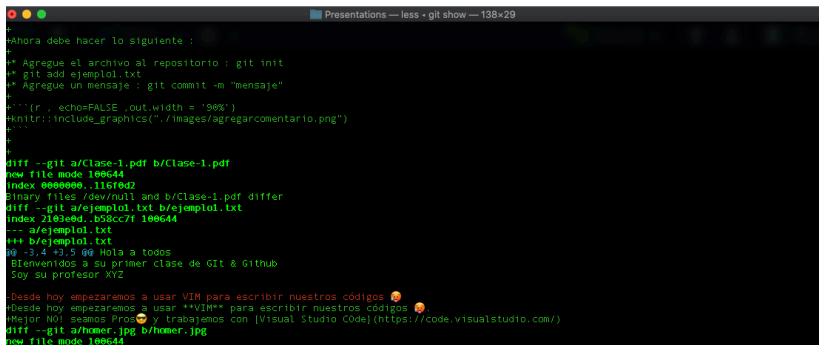
Desde hoy empezaremos a usar **VIM** para escribir nuestros códigos 🤖
Mejor NO! seamos Pros 😊 y trabajemos con [Visual Studio Code] https://code.visualstudio.com/

-- INSERT --
```

# ¿Para qué sirve Git?

Ahora :

- Agregar todos los cambios : `git add` . {ojo que los espacios son importantes}
- Comentar los cambios : `git commit -m "Cambie el editor de texto"`
- Verifica el estatus de cambio : `git status`



```
Presentations — less + git show — 138x29
+
+Ahora debe hacer lo siguiente :
+
+* Agregue el archivo al repositorio : git init
+* git add ejemplo1.txt
+* Agregue un mensaje : git commit -m "mensaje"
+
+'''(r , echo=FALSE ,out.width = '90%')
+knitr::include_graphics("../images/agregarcomentario.png")
+'''
+
+
+diff --git a/Clase-1.pdf b/Clase-1.pdf
+new file mode 100644
+index 0000000..11670d2
+Binary files /dev/null and b/Clase-1.pdf differ
+diff --git a/ejemplo1.txt b/ejemplo1.txt
+index 2103e0d..b58cc7f 100644
+--- a/ejemplo1.txt
++++ b/ejemplo1.txt
+@@ -3,4 +3,5 @@ Hola a todos
+ Bienvenidos a su primer clase de Git & Github
+ Soy su profesor XYZ
+
+Desde hoy empezaremos a usar VIM para escribir nuestros códigos 🐼
+Desde hoy empezaremos a usar **VIM** para escribir nuestros códigos 🐼
+Mejor NO! seamos Pros 🐼 y trabajemos con [Visual Studio Code](https://code.visualstudio.com/)
+diff --git a/homer.jpg b/homer.jpg
+new file mode 100644
```

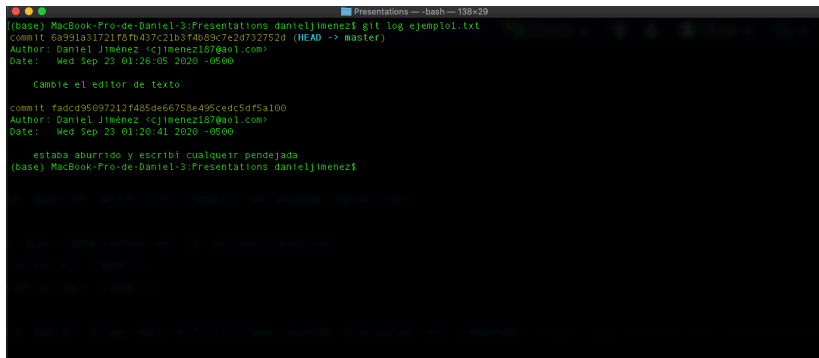
# ¿Para qué sirve Git?

Observese que en anterior cambio se puede observar:

- Lineas que cambiaron en la actualización.
- Quien hizo el cambio.
- el estatus del cambio.

Si quiere hacer algo más sofisticado puede ejecutar el comando **git log nombre del documento**

# ¿Para qué sirve Git?



```
Presentations - bash - 138x29
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git log ejemplo1.txt
commit 6a991a31721f8fb437c21b3f4b89c7e2d732752d (HEAD -> master)
Author: Daniel Jiménez <cjimenez187@aol.com>
Date:   Wed Sep 23 01:26:05 2020 -0500

    Cambie el editor de texto

commit fadcd95097212f485de66758e495cedc5df5a100
Author: Daniel Jiménez <cjimenez187@aol.com>
Date:   Wed Sep 23 01:20:41 2020 -0500

    estaba aburrido y escribí cualquier pendejada
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$
```

# ¿Para qué sirve Git?

Como tal Git es un sistema de control distribuido ¿Qué quiere decir eso?

*“La mayoría de los sistemas de control de versiones utilizan un servidor centralizado para guardar todos los cambios y actualizaciones. Esta característica es maravillosa porque permite que personas en diferentes partes del mundo puedan trabajar sobre un mismo proyecto, sin embargo está fuertemente atada a la conexión a Internet para funcionar, de otra manera los cambios nunca podrán reflejarse en el servidor central.”*

# ¿Para qué sirve Git?

“Git se presenta como un sistema distribuido, en el que todos los nodos manejan la información en su totalidad y por lo tanto pueden actuar de cliente o servidor en cualquier momento, es decir, se elimina el concepto de “centralizado”. Esto se lo logra gracias a que cada vez que sincronizas los cambios con el repositorio remoto Git, te guarda una copia entera de los datos con toda la estructura y los archivos necesarios. Así ya no es necesario salir a Internet para consultar los cambios históricos sobre un archivo o para ver quién fue la última persona que lo editó, todo se hace directamente sobre tu copia local y luego, cuando lo consideres oportuno, puedes enviar esos cambios hacia el repositorio remoto.” <sup>1</sup>

---

<sup>1</sup>tomado de la página de Git

# ¿Para qué sirve Git?

Dicho lo anterior Git sirve para :

- Cada uno de los miembros del equipo tenga una versión local de un proyecto;
- Crear ramificaciones del proyecto;
- Integrar las partes del proyecto (pull request, no lo haga de otra manera por higiene del repositorio )

# ¿Para qué sirve Git?

## Algunas definiciones necesarias

- Repositorio: Lugar donde se almacena los datos actualizados y la historia de los mismos (versiones) con respecto a sus cambios.
- Revisión: Versión determinada de la información que se gestiona.
- Tag: Son etiquetas que permiten identificar de forma fácil las partes y versiones de un proyecto.
- Change: Modificación específica a un documento o código.



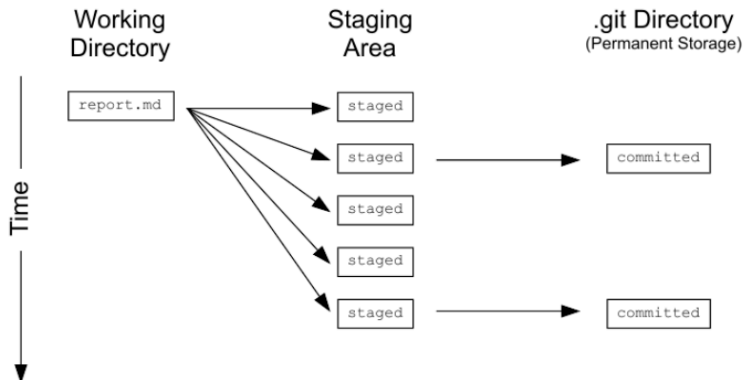
# ¿Para qué sirve Git?

## Algunas definiciones necesarias

- Branch: Conjunto de archivos que pueden ser ramificados.
- Checkout: Crear una copia local desde el repositorio.
- Commit : Leyenda de los cambios
- Conflict : Conflicto cuando varias partes hacen cambios en la misma parte del código o repositorio.
- Merge : Integrar.

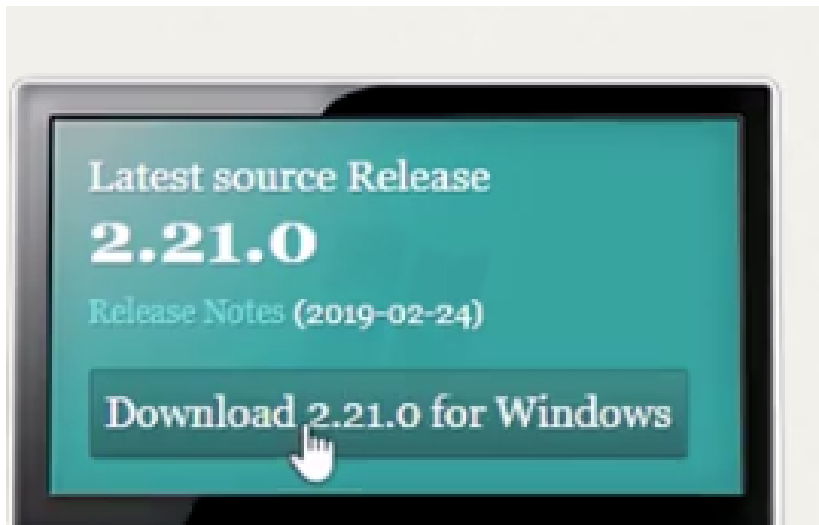
# ¿Para qué sirve Git?

Lo anterior sirve para validar el siguiente flujo



## ¿Cómo se instala Git?

- Para el caso de Windows, diríjase a la página de git haciendo click aquí



## ¿Cómo se instala Git?

- En el caso de Mac, ya viene instalado, pero por si algo haga click aquí



# ¿Cómo se instala Git?

## Observación:

- Para sacarle el provecho a git y a su máquina trabaje desde la terminal o CMD
- Existen varios editores de texto, pero tenga presente que cada uno tiene una función
- VIM es un editor difícil de manejar, pero optimizado para el trabajo de máquina.

Acá hay un tema importante, existen editores para todo tipo de proyecto, en esta diapositiva presento algunas apreciaciones y experiencias sobre los editores :

- Si eres desarrollador (bases de datos, ciencia de datos, analítica, devops) usa Visual Studio Code
- Si trabajas con gestión documental trabaja con Sublime Text
- Otra alternativa para gestión documental, o actualización de códigos por devops es Atom

## Editores de texto para Git

Si no le temes a nada, eres el alfa independiente a cualquier aspecto y en especial eres osado(a) y te gusta el riesgo, trabaja con VIM.

A meme featuring Redd Foxx from the TV show 'The 747 Squad'. He is shown from the chest up, wearing a dark suit, white shirt, and patterned tie. He has a mustache and is looking slightly to his left with a confident expression. His hands are raised in front of him, palms facing forward. Overlaid on the image is a thick black horizontal bar at the top, and the text 'AY-YA' in large, bold, white capital letters with a black outline is centered over his head.

**AY-YA**

# Editores de texto para Git

Para este curso se trabajará con Visual Studio Code.

Razón: Esta optimizado para deploy.

Pero es necesario ser drásticos y trabajaremos un rato en VIM.



# Editores de texto para Git

En el ejercicio en Vivo mostraré más cosas de VIM, pero dejo la introducción de varios comandos necesarios :

- touch : Para crear un archivo;
- VIM open archivo.ext: para abrir el archivo;
- I: Insertar comentarios ;
- :wq: Para salir de VIM

```
[(base) MacBook-Pro-de-Daniel-3:~ danieljimenez$ cd Desktop/Github-Class/ ]
[(base) MacBook-Pro-de-Daniel-3:Github-Class danieljimenez$ cd Presentations/ ]
[(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ ls ]
Clase 1.Rmd      ejemplo1.txt      images           vim1.png
Clase-1.pdf      homer.jpg         progit.pdf
[(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ VIM ejemplo1.txt ]
```

# Editores de texto para Git

```
Hola a todos

Bienvenidos a su primer clase de Git & Github
Soy su profesor XYZ

Desde hoy empezaremos a usar **VIM** para escribir nuestros códigos 🤖.
Mejor NO! seamos Pros 😎 y trabajemos con [Visual Studio C0de](https://code.visualstudio.com/)

A mis estudiantes les gusta el riesgo, por lo tanto quieren trabajar un rato en
VIM, y ¿quién soy yo? para negarles ese privilegio█
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

Notese que hice una modificación al documento y recuerde estos pasos :

- `git init;`
- `git add . ;`
- `git commit -m "mensaje" ;`
- `git status`
- `git log ejemplo.txt`

**AY-YA**



# Editores de texto para Git

```
Presentations -- -bash -- 80x24
-rw-r--r--  1 danieljimenez  staff   1082 Sep 23 22:29 LICENSE
drwxr-xr-x  9 danieljimenez  staff    288 Sep 23 22:29 Presentations
-rw-r--r--  1 danieljimenez  staff  885514 Sep 23 22:29 README.html
-rw-r--r--  1 danieljimenez  staff   3979 Sep 23 22:29 README.md
(base) MacBook-Pro-de-Daniel-3:Github-Class danieljimenez$ cd Presentations/
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ touch ruta_navegacion.txt
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ vim ruta_navegacion.txt
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git add .
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git add ruta_navegacion.txt
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git status
On branch master
Your branch is up to date with 'origin/master'.

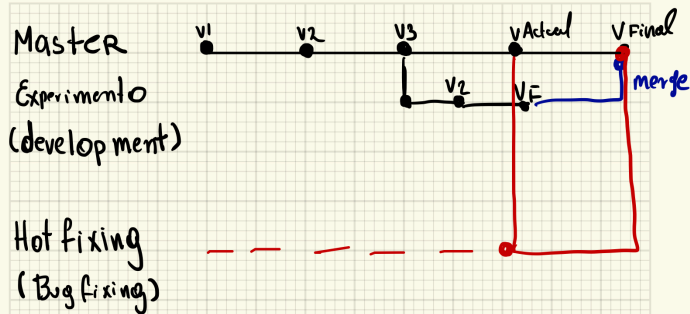
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   Clase 1.Rmd
    new file:   images/ciclo.jpeg
    new file:   images/gitfirst.png
    new file:   ruta_navegacion.txt

(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$
```

# Editores de texto para Git

Para sacarle provecho a Git es necesario conocer y trabajar con la terminal y la línea de comando. Acá recompilaré algunos comandos y buenas prácticas, pero les recomiendo que la estudien por su cuenta, se darán cuenta que no cambiaran el CMD o la terminal por ningún interfaz gráfico.

# Primeros comandos en Git



# Primeros comandos en Git - crear un repositorio -

- Se debe designar donde está la carpeta central de los archivos

```
[(base) MacBook-Pro-de-Daniel-3:Github-Class danieljimenez$ git init.  
git: 'init.' is not a git command. See 'git --help'.  
  
The most similar command is  
  init  
[(base) MacBook-Pro-de-Daniel-3:Github-Class danieljimenez$ git init  
Reinitialized existing Git repository in /Users/danieljimenez/Desktop/Github-Class/.git/  
[(base) MacBook-Pro-de-Daniel-3:Github-Class danieljimenez$ ls -al  
ls: -: No such file or directory  
ls: al: No such file or directory  
[(base) MacBook-Pro-de-Daniel-3:Github-Class danieljimenez$ ls -al  
total 1776  
drwxr-xr-x  10 danieljimenez  staff    320 Sep 23 22:30 .  
drwx-----@ 15 danieljimenez  staff    480 Sep 23 22:57 ..  
-rw-r--r--@   1 danieljimenez  staff   6148 Sep 23 22:57 .DS_Store  
drwxr-xr-x  12 danieljimenez  staff    384 Sep 23 23:00 .git  
-rw-r--r--   1 danieljimenez  staff     40 Sep 23 22:29 .gitignore  
drwxr-xr-x   4 danieljimenez  staff    128 Sep 23 22:30 Github  
-rw-r--r--   1 danieljimenez  staff   1082 Sep 23 22:29 LICENSE  
drwxr-xr-x   9 danieljimenez  staff    288 Sep 23 22:29 Presentations  
-rw-r--r--   1 danieljimenez  staff  885514 Sep 23 22:29 README.html  
-rw-r--r--   1 danieljimenez  staff   3979 Sep 23 22:29 README.md  
(base) MacBook-Pro-de-Daniel-3:Github-Class danieljimenez$
```



# Primeros comandos en Git - crear un repositorio -

- El `.git` es una carpeta oculta que reposará los cambios de los archivos.
- Cada uno de los cambios se guarda de manera independiente.

Ahora se creará un nuevo archivo y tendrá algunas particularidades.

Desarrolle el nuevo archivo con VIM o con Visual Studio Code

# Primeros comandos en Git

Al crear el archivo es necesario hacer git add “nombre del archivo”, y después verificar el estatus.

```
Presentations — vim ruta_navegacion.txt — 80x24
#=====#
#   Lecturas e instrumentos   #
#   necesarios para el      #
#   desarrollo de NLP       #
#=====#

Para el desarrollo de NLP es necesario que revise la siguiente documentación:

* "Text Mining with R" -> https://www.tidytextmining.com/
* "Supervised Machine Learning for Text Analysis" https://smltar.com/

Paso seguido vea los siguientes ejemplos

* https://www.youtube.com/watch?v=RggCAXBe6BA
* https://www.youtube.com/watch?v=evTuL-RcRpc&t=171s

Y lo último pero no menos importante:
> "La practica hace al maestro"
~
-- INSERT --
```

# Primeros comandos en Git

```
Presentations -- -bash -- 80x24
-rw-r--r--  1 danieljimenez  staff   1082 Sep 23 22:29 LICENSE
drwxr-xr-x  9 danieljimenez  staff    288 Sep 23 22:29 Presentations
-rw-r--r--  1 danieljimenez  staff  885514 Sep 23 22:29 README.html
-rw-r--r--  1 danieljimenez  staff   3979 Sep 23 22:29 README.md
(base) MacBook-Pro-de-Daniel-3:Github-Class danieljimenez$ cd Presentations/
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ touch ruta_navegacion.txt
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ vim ruta_navegacion.txt
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git add .
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git add ruta_navegacion.txt
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   Clase 1.Rmd
    new file:   images/ciclo.jpeg
    new file:   images/gitfirst.png
    new file:   ruta_navegacion.txt

(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$
```

# Primeros comandos en Git

Se finaliza con un git commit -m “escribe algo”

```
Presentations — -bash — 80x24
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ vim ruta_navegacion.txt
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git add .
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git add ruta_navegacion.txt
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   Clase 1.Rmd
        new file:   images/ciclo.jpeg
        new file:   images/gitfirst.png
        new file:   ruta_navegacion.txt

(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git commit -m "Ruta de navegación"
[master 2fcc2d7] Ruta de navegación
 4 files changed, 39 insertions(+), 3 deletions(-)
 create mode 100644 Presentations/images/ciclo.jpeg
 create mode 100644 Presentations/images/gitfirst.png
 create mode 100644 Presentations/ruta_navegacion.txt
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$
```

# Primeros comandos en Git

Si su git no está configurado haga `git config git config --list`

- Para hacer los cambios globales : `git config --global user.name "su nombre"`
- Para hacer los cambios globales : `git config --global user.mail "su mail"`
- Vea los cambios con `git config --list`

# Primeros comandos en Git

Suponga que quiere hacer una modificación del documento y ver quién la hizo.

```
Presentations — vim ruta_navegacion.txt — 80x24
#      necesarios para el      #
#      desarrollo de NLP      #
#=====#

Para el desarrollo de NLP es necesario que revise la siguiente documentación:

* "Text Mining with R" -> https://www.tidytextmining.com/
* "Supervised Machine Learning for Text Analysis" https://smltar.com/

Paso seguido vea los siguientes ejemplos

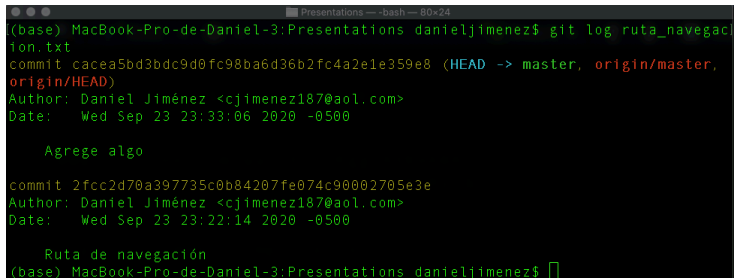
* https://www.youtube.com/watch?v=RggCAXBe6BA
* https://www.youtube.com/watch?v=evTuL-RcRpc&t=171s

Y lo último pero no menos importante:
> "La practica hace al maestro"

Si tiene alguna duda comuníquese con Mary <- Esta es la modificación
```

# Primeros comandos en Git

Con `git log 'nombre del documento'` se puede ver la historia del archivo.



```
Presentations — -bash — 80x24
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git log ruta_navegacion.txt
commit cacea5bd3bdc9d0fc98ba6d36b2fc4a2e1e359e8 (HEAD -> master, origin/master, origin/HEAD)
Author: Daniel Jiménez <cjimenez187@aol.com>
Date:   Wed Sep 23 23:33:06 2020 -0500

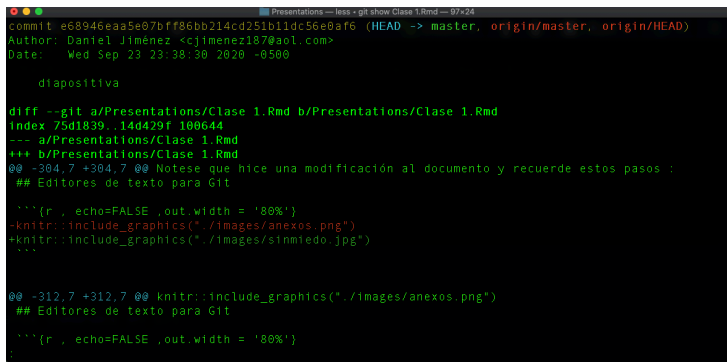
    Agrega algo

commit 2fcc2d70a397735c0b84207fe074c90002705e3e
Author: Daniel Jiménez <cjimenez187@aol.com>
Date:   Wed Sep 23 23:22:14 2020 -0500

    Ruta de navegación
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$
```

# Primeros comandos en Git

Escriba `git show` + nombre del documento y vea los cambios en el archivo



```
commit e68946eaa5e07bfff86bb214cd251b11dc56e0af6 (HEAD -> master, origin/master, origin/HEAD)
Author: Daniel Jiménez <cjimenez187@aol.com>
Date:   Wed Sep 23 23:38:30 2020 -0500

    diapositiva

diff --git a/Presentations/Clase 1.Rmd b/Presentations/Clase 1.Rmd
index 75d1839..14d429f 100644
--- a/Presentations/Clase 1.Rmd
+++ b/Presentations/Clase 1.Rmd
@@ -304,7 +304,7 @@
 ## Editores de texto para Git

 ```{r , echo=FALSE ,out.width = '80%'}
-knitr::include_graphics("../images/anexos.png")
+knitr::include_graphics("../images/sinmiedo.jpg")
 ```

@@ -312,7 +312,7 @@ knitr::include_graphics("../images/anexos.png")
 ## Editores de texto para Git

 ```{r , echo=FALSE ,out.width = '80%'}
:
```



# Primeros comandos en Git