

Introducción a Git & Github

Conceptos básicos

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
- Flujo de trabajo
- Introducción a Github.

¿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.

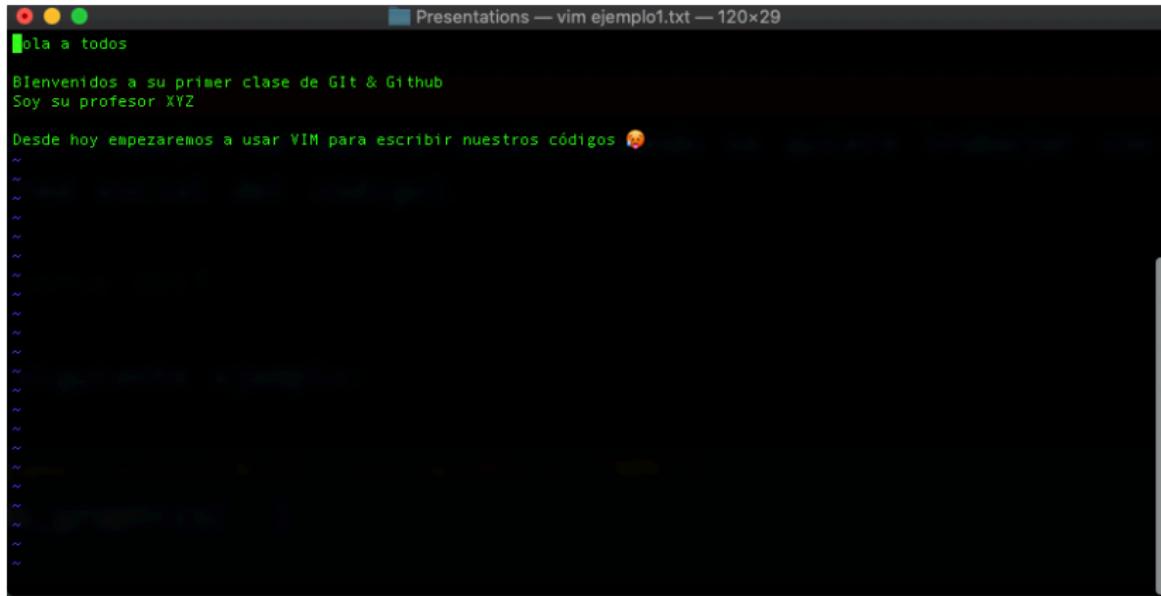
¿Qué es Git?

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?

Imaginesé el siguiente ejemplo:



The screenshot shows a terminal window titled "Presentations — vim ejemplo1.txt — 120x29". The window contains the following text:

```
ola a todos
Bienvenidos a su primer clase de GIT & Github
Soy su profesor XYZ

Desde hoy empezaremos a usar VIM para escribir nuestros códigos 🎉
```

The terminal has a dark background with light-colored text. The title bar is at the top, and the main content area shows the text in a monospaced font. There are several blank lines below the text.

¿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
```

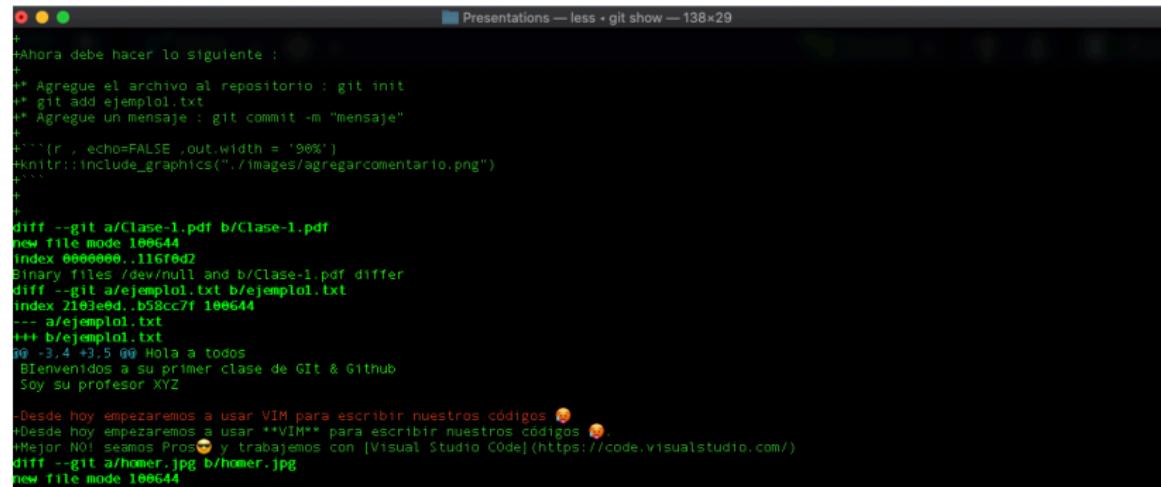
¿Para qué sirve Git?

Ahora suponga que quiere hacer unos cambios que son importantes.

¿Para qué sirve Git?

Ahora :

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



```
+ 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 000000..116f6d2
Binary files /dev/null and b/Clase-1.pdf differ
diff --git a/ejemplo1.txt b/ejemplo1.txt
index 2193e8d..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 Prosesores y trabajemos con [Visual Studio Code](https://code.visualstudio.com/)
diff --git a/homer.jpg b/homer.jpg
new file mode 100644
```

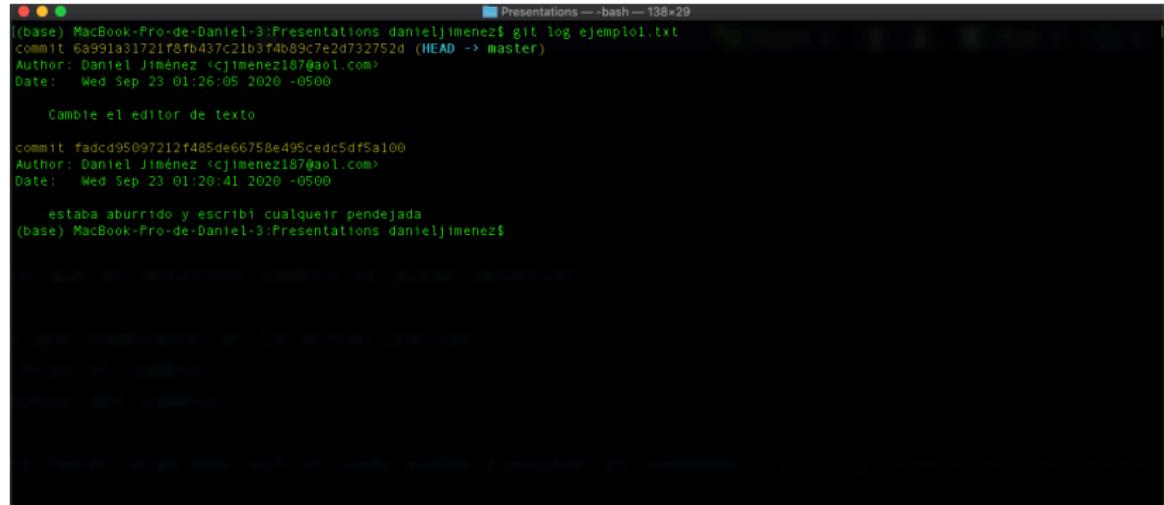
¿Para qué sirve Git?

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?



A screenshot of a terminal window titled "Presentations — bash — 138x29". The window shows the command "git log ejemplo1.txt" being run in a directory "(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez\$". The output displays two commits:

```
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git log ejemplo1.txt
commit 6a991a31721t870437c21b3Td089c7e2d732752d (HEAD -> master)
Author: Daniel Jiménez <cjmenez187@aol.com>
Date:   Wed Sep 23 01:26:05 2020 -0500

    Cambie el editor de texto

commit fadcd95097212f485de66758e495cedc5df5a100
Author: Daniel Jiménez <cjmenez187@aol.com>
Date:   Wed Sep 23 01:20:41 2020 -0500

    estaba aburrido y escribi 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.”¹

¹tomado 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.

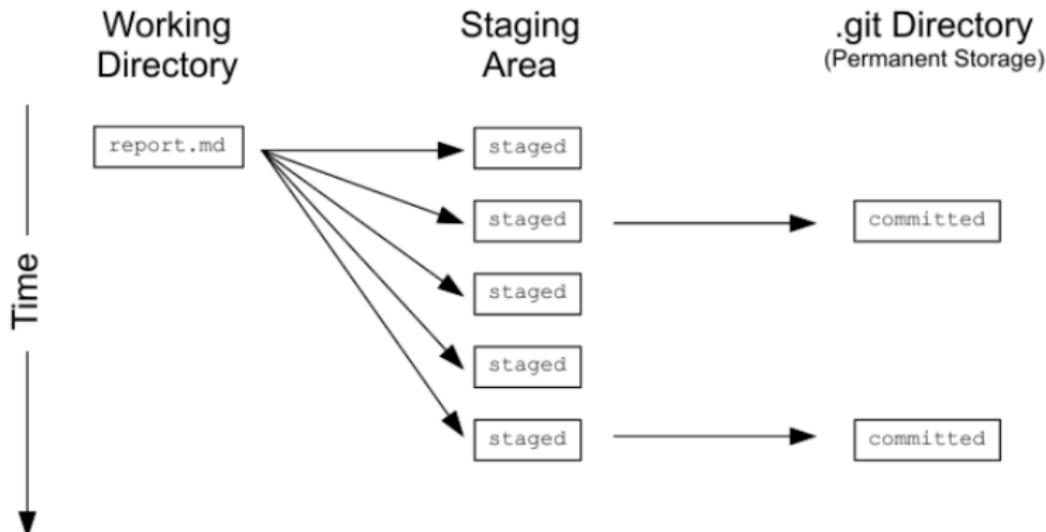
¿Para qué sirve Git?

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.

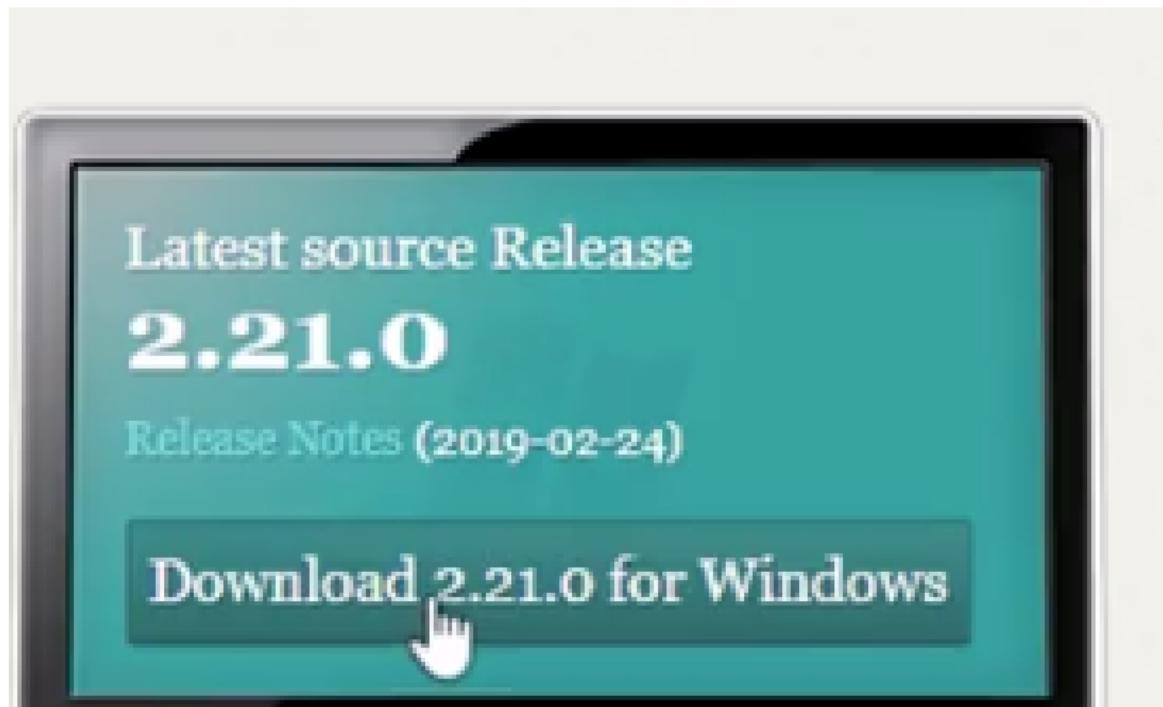
¿Para qué sirve Git?

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.

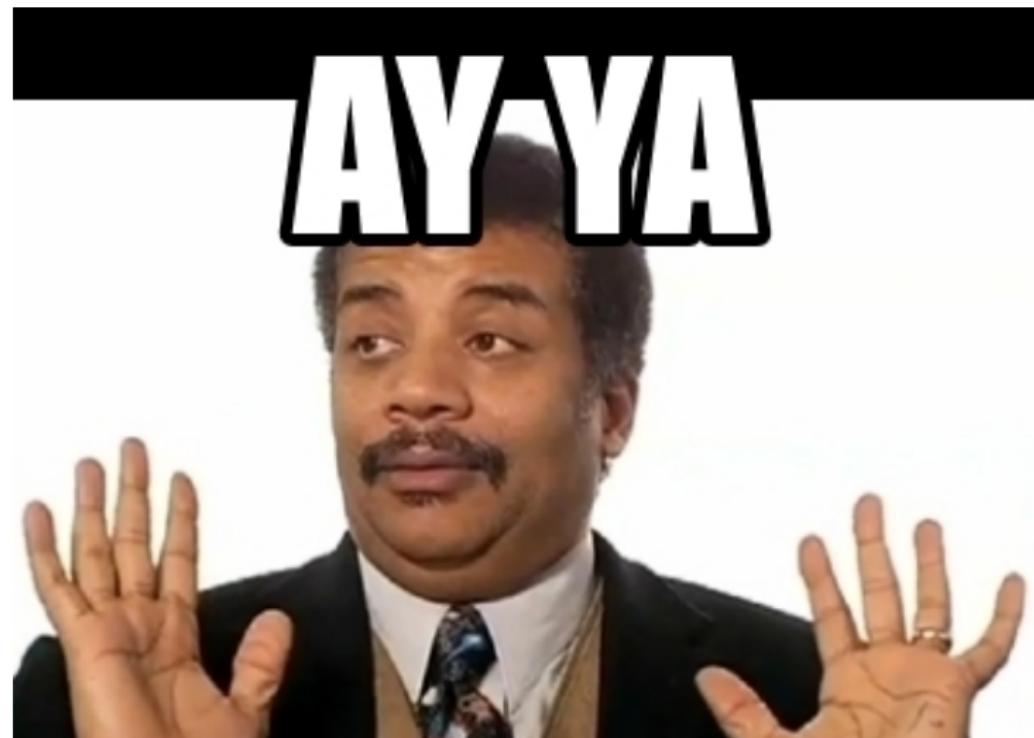
Editores de texto para Git

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.



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 Code](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 --
```

Editores de texto para Git

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'.

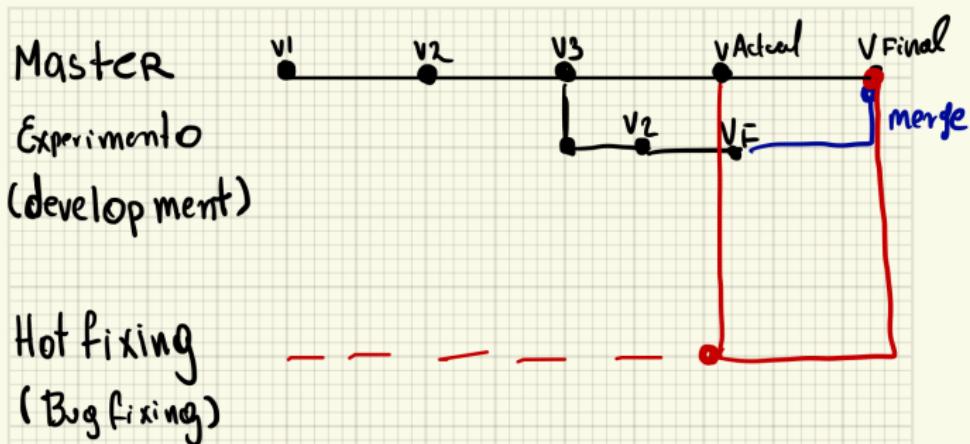
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified: Clase 1.Rmd
    new file: images/ciclo.jpeg
    new file: images/gitfirst.png
    new file: ruta_navegacion.txt

(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 recomendaré algunos comandos y buenas prácticas, pero les recomiendo que la estudien por su cuenta, se darán cuenta que no cambiaron 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-Clas
ss/.git/
[(base) MacBook-Pro-de-Daniel-3:Github-Class danieljimenez$ ls - al
ls: .: No such file or directory
ls: ..: 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”

```
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 --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.

```
# 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

    Agrege 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

```
Presentations — less • git show Clase 1.Rmd — 97×24
commit e68946eaa5e07bff86bb214cd251b11dc56e0af6 (HEAD -> master, origin/master, origin/HEAD)
Author: Daniel Jiménez <cjmenez187@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 @@ Notese que hice una modificación al documento y recuerde estos pasos :
## Editores de texto para Git

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

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

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

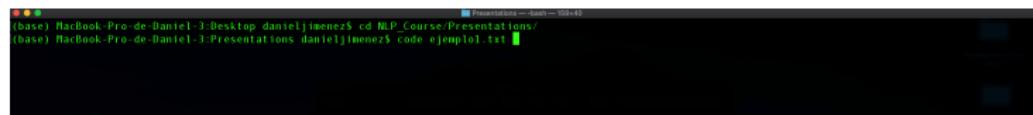
# Primeros comandos en Git

Ahora haremos una modificación al documento pero con Visual Studio Code, aplicando los siguientes pasos :

- Ir a la terminal o el CMD;
- Cambiar de directorio hasta llegar a donde se encuentre el documento;
- Escribir en la terminal o cmd code + nombre del documento

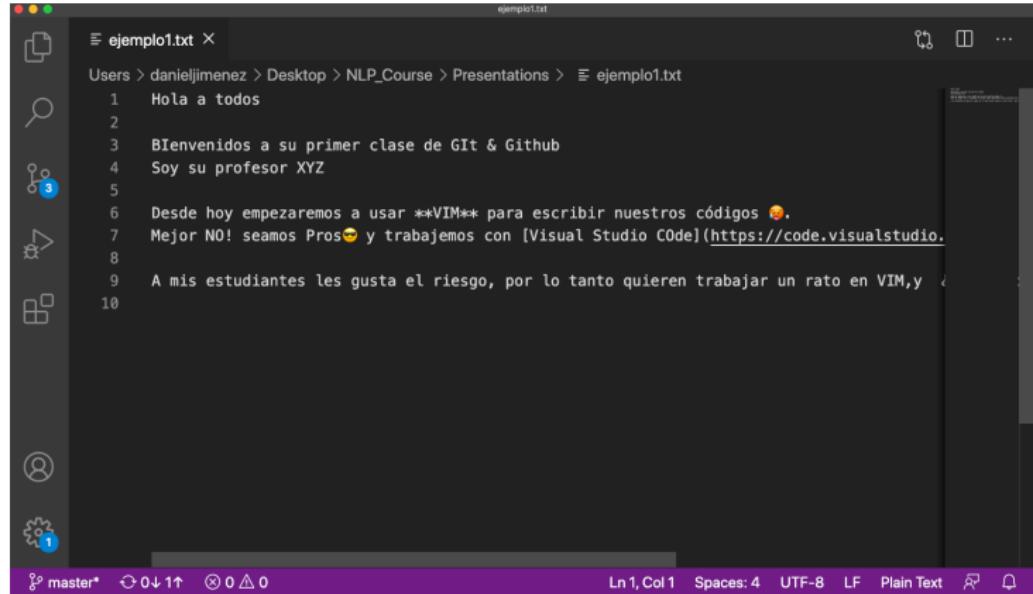
Lo que aparecerá sera lo siguiente

# Primeros comandos en Git



```
(base) MacBook-Pro-de-Daniel1:3 Desktop danieljimenez$ cd NLP.Course/Presentations/
(base) MacBook-Pro-de-Daniel1:3 Presentations danieljimenez$ code ejemplo.txt
```

# Primeros comandos en Git

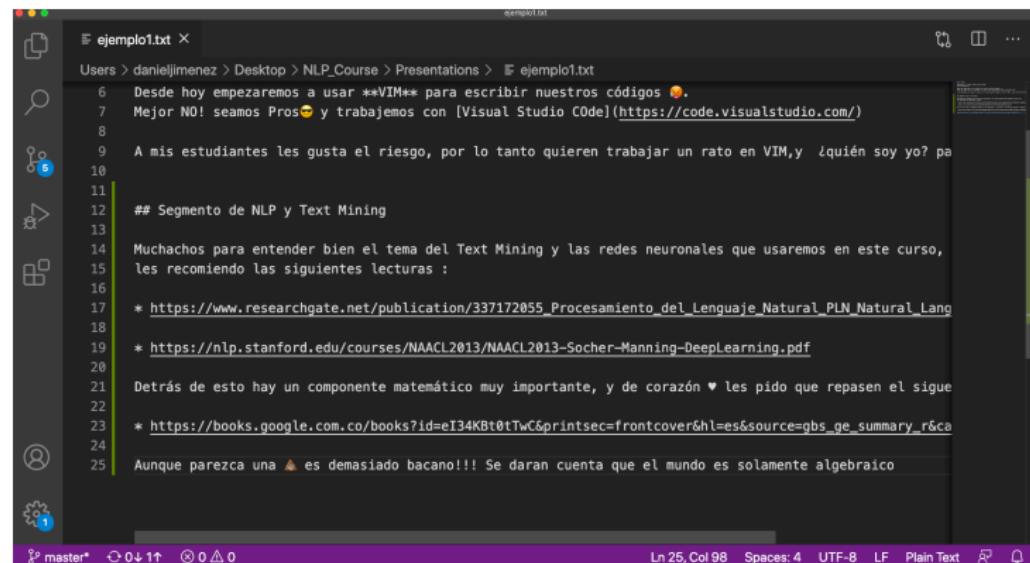


```
ejemplo1.txt
Users > danieljimenez > Desktop > NLP_Course > Presentations > ejemplo1.txt
1 Hola a todos
2
3 BIenvenidos a su primer clase de GIIt & Github
4 Soy su profesor XYZ
5
6 Desde hoy empezaremos a usar **VIM** para escribir nuestros códigos 😊.
7 Mejor NO! seamos Pros😎 y trabajemos con [Visual Studio C0de](https://code.visualstudio.com).
8
9 A mis estudiantes les gusta el riesgo, por lo tanto quieren trabajar un rato en VIM,y
10
```

# Primeros comandos en Git

Se agregan unos cambios y se analizan

(Recuerde agregar y hacer el commit)



```
Users > danieljimenez > Desktop > NLP_Course > Presentations > ejemplo1.txt
6 Desde hoy empezaremos a usar **VIM** para escribir nuestros códigos 😊.
7 Mejor NO! seamos Pros 🚀 y trabajemos con [Visual Studio Code](https://code.visualstudio.com/)
8
9 A mis estudiantes les gusta el riesgo, por lo tanto quieren trabajar un rato en VIM, y ¿quién soy yo? pa
10
11
12 ## Segmento de NLP y Text Mining
13
14 Muchachos para entender bien el tema del Text Mining y las redes neuronales que usaremos en este curso,
15 les recomiendo las siguientes lecturas :
16
17 * https://www.researchgate.net/publication/337172055_Procesamiento_del_Lenguaje_Natural_PLN_Natural_Lang
18
19 * https://nlp.stanford.edu/courses/NAACL2013/NAACL2013-Socher-Manning-DeepLearning.pdf
20
21 Detrás de esto hay un componente matemático muy importante, y de corazón ❤ les pido que repasen el sigue
22
23 * https://books.google.com.co/books?id=eI34KBt0tTwC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&ca
24
25 Aunque parezca una 🔥 es demasiado bacano!!! Se darán cuenta que el mundo es solamente algebraico
```

master\* 0 0 ▲ 1 0 0 △ 0 Ln 25, Col 98 Spaces: 4 UTF-8 LF Plain Text ⌂ ⌂

# Primeros comandos en Git

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

```
Segmento de NLP y Text Mining
```

```
Muchachos para entender bien el tema del Text Mining y las redes neuronales que usaremos en este curso,
les recomiendo las siguientes lecturas :
```

```
* https://www.researchgate.net/publication/337172855_Procesamiento_del_Lenguaje_Natural_PLN_Natural-Language_Processing_NLP
```

```
* https://nlp.stanford.edu/courses/NAACL2013/NAACL2013-Socher-Manning-DeepLearning.pdf
```

```
Detrás de esto hay un componente matemático muy importante, y de corazón ❤ les pido que repasen el siguiente documento:
```

```
* https://books.google.com.co/books?id=eJ3kBt0TuC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0&q=onepage&qf=false
```

```
Aunque parezca una ▲ es demasiado bacanoff! Se darán cuenta que el mundo es solamente algebraico ▶
```

```
No newline at end of file
```

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

# Primeros comandos en Git

Para ver la historia de los documentos se emplea el comando git log + nombre documento

```
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git log ejemplol.txt
commit cf6b7862921c0287c78a5981c8ab2dee17356281 (HEAD -> master)
Author: Daniel Jiménez <cjmenez187@aol.com>
Date: Wed Sep 30 10:39:09 2020 -0500

 Documentos para la clase

commit b0f684960b3c672dde891b19055b708adf8114c7
Author: Daniel Jiménez <cjmenez187@aol.com>
Date: Wed Sep 23 15:20:36 2020 -0500

 Agregue una linea

commit bdd297991bc72047f91e9554d2add01262458875
Author: Daniel Jiménez <cjmenez187@aol.com>
Date: Wed Sep 23 01:50:38 2020 -0500

 Cambios en la presentación 1
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ █
```

# Primeros comandos en Git

Para ver las diferencias entre un archivo y otro se usa el comando git diff + nombre del commit final + nombre del commit a comparar

```
diff --git a/Presentations/Clase 1.Rmd b/Presentations/Clase 1.Rmd
index d6ce5a1..fc7d582 100644
--- a/Presentations/Clase 1.Rmd
+++ b/Presentations/Clase 1.Rmd
@@ -27,7 +27,7 @@ options$knitr.duplicate.label = "allow")
¿Cómo se instala GIT?
Editores de texto para GIT

Primeros comandos en GIT
Ejemplo de uso de Git
-88.7 +88.7 @@ Por lo tanto:
Git se usa para trabajar de manera local, cuando se quiere trabajar con varias personas se debe usar github (red social del código).
```{r , echo=FALSE , out.width = '98%'}
knitr::include_graphics("./images/homer.jpg")
knitr::include_graphics("homer.jpg")
```
-88.7 +88.7 @@ knitr::include_graphics("./images/homer.jpg")
Imaginése el siguiente ejemplo:
```{r , echo=FALSE , out.width = '98%'}
knitr::include_graphics("./images/vim1.png")
knitr::include_graphics("vim1.png")
```
-284,172 +284,5 @@ En el ejercicio en Vivo mostraré más cosas de VIM, pero dejo la introducción
knitr::include_graphics("./images/VIM2.png")
```
## -88.7 +88.7 @@ Editores de texto para GIT
```{r , echo=FALSE , out.width = '98%'}
knitr::include_graphics("./images/vim3.png")
```

```

Primeros comandos en Git

Para volver a una versión anterior de los documentos se trabaja de la siguiente manera:

- git reset + commit
 - Se puede usar la extensión – hard : Todo se convierte en su estado natural anterior.
 - Se puede usar la extensión – soft : Y se vuelve a una versión anterior sin tener presente los cambios en el staging

Primeros comandos en Git

Un ejemplo :

```
git --git a/Presentations/Clase 1.Rmd b/Presentations/Clase 1.Rmd
index d6ce5a1..fc7d5b2 100644
--- a/Presentations/Clase 1.Rmd
+++ b/Presentations/Clase 1.Rmd
@@ -27,7 +27,7 @@ options(knitr.duplicate.label = "allow")
 * ¿Cómo se instala GIT?
 
 * Editores de texto para Git
 [REDACTED]
 
 * Primeros comandos en GIT
 [REDACTED]
 
 * Ejemplo de uso de Git
 @@ -68,7 +68,7 @@ Por lo tanto:
 Git se usa para trabajar de manera local, cuando se quiere trabajar con varias personas se debe usar github (red social del código).
 
 {{{r , echo=FALSE, out.width = '98%'}
knitr::include_graphics("./images/homer.jpg")
knitr::include_graphics("homer.jpg")
}}}
 
@@ -88,7 +88,7 @@ knitr::include_graphics("./images/homer.jpg")
Imaginé el siguiente ejemplo:
 
 {{{r , echo=FALSE, out.width = '98%'}
knitr::include_graphics("./images/vim1.png")
knitr::include_graphics("vim1.png")
}}} 
@@ -284,172 +284,5 @@ En el ejercicio en Vivo mostraré más cosas de VIM, pero dejó la introducción
knitr::include_graphics("./images/VIM2.png")
}}}
 
## Editores de texto para Git
[REDACTED]
 
{{{r , echo=FALSE, out.width = '98%'}
knitr::include_graphics("./images/vim3.png")
}}} 
```

Primeros comandos en Git

Se anexan los cambios locales

```
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ code ejemplo1.txt
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git add ejemplo1.txt
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git commit -m "Anexo terror"
[master f7e3d01] Anexo terror
 1 file changed, 3 insertions(+), 1 deletion(-)
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$
```

Primeros comandos en Git

Se verifican los cambios a través de los logs

```
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ code ejemplo1.txt
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git add ejemplo1.txt
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git commit -m "Anexo terror"
[master f7e3db] Anexo terror
 1 file changed, 3 insertions(+), 1 deletion(-)
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git log ejemplo1.txt
commit f7e3db0021646f0cb36a0dbc10fa0be2f19 (HEAD -> master)
Author: Daniel Jiménez <cjmenez107@aol.com>
Date:   Wed Sep 30 10:39:09 2020 -0500

    Anexo terror

commit cf6b7862921c8287c78a5981c8ab2dee17356281
Author: Daniel Jiménez <cjmenez107@aol.com>
Date:   Wed Sep 30 10:39:09 2020 -0500

    Documentos para la clase

commit b8f684960b3c672dde891b19955b788adfb8114c7
Author: Daniel Jiménez <cjmenez107@aol.com>
Date:   Wed Sep 23 15:28:36 2020 -0500

    Agregue una linea

commit bdd297991bc72847f91e9554d2add81262458075
Author: Daniel Jiménez <cjmenez107@aol.com>
Date:   Wed Sep 23 01:50:38 2020 -0500

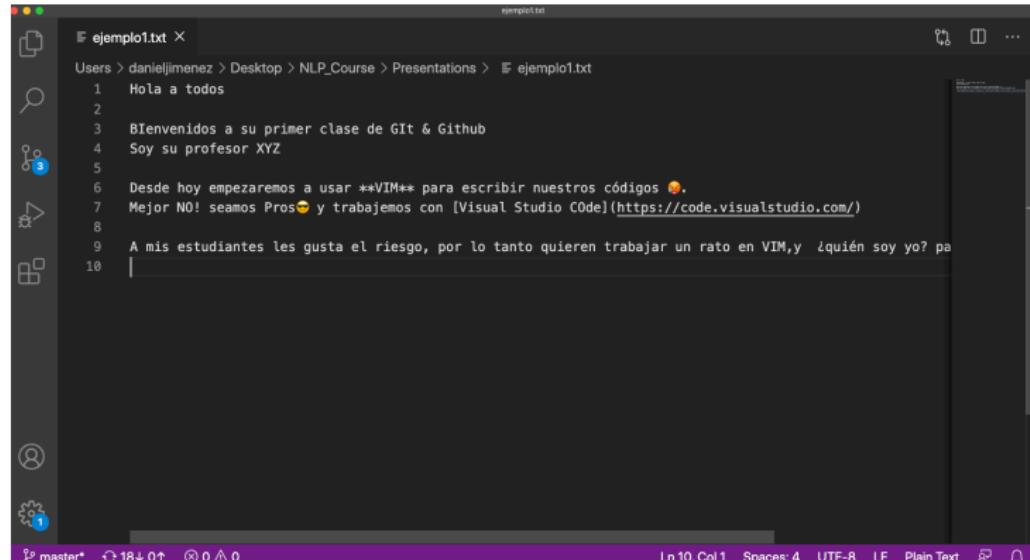
    Cambios en la presentación 1
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$
```

Primeros comandos en Git

Ahora se retrocede de manera hard , con el comando

```
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$ git reset b0f684960b3c672dde891b19055b780adf8114c7 --hard
HEAD is now at b0f6849 Agregue una linea
(base) MacBook-Pro-de-Daniel-3:Presentations danieljimenez$
```

Primeros comandos en Git



The screenshot shows a terminal window with the following content:

```
ejemplo1.txt
Users > danieljimenez > Desktop > NLP_Course > Presentations > ejemplo1.txt
1 Hola a todos
2
3 BIenvenidos a su primer clase de GIT & Github
4 Soy su profesor XYZ
5
6 Desde hoy empezaremos a usar **VIM** para escribir nuestros códigos 😊.
7 Mejor NO! seamos Pros 😎 y trabajemos con [Visual Studio Codel](https://code.visualstudio.com/)
8
9 A mis estudiantes les gusta el riesgo, por lo tanto quieren trabajar un rato en VIM,y ¿quién soy yo? pa
10 |
```

The terminal status bar at the bottom indicates:

- master*
- 18+0t
- 0.0 ▲ 0
- Ln 10, Col 1
- Spaces: 4
- UTF-8
- LF
- Plain Text

Primeros comandos en Git

Ahora hagamos un ejemplo con R.

The screenshot shows the RStudio interface with the following details:

- Left Panel (Code Editor):** Displays the content of `Script_example.R`. The code includes library imports for `widyr`, `tidyTuesdayR`, `lubridate`, `tidylo`, and `scales`. It then loads data sets from GitHub URLs for Beyoncé lyrics, Taylor Swift lyrics, sales data, and charts. Finally, it starts an "Exploratory Data Analysis" section.
- Top Bar:** Shows tabs for `README.md`, `Clase 1.Rmd`, and `Script_example.R`. There are also buttons for `Run` and `Source`.
- Console Tab:** Shows the command `git add -A` followed by a list of files being added: `beyonce_lyrics.RData`, `taylor_swift_lyrics.RData`, `sales.RData`, `charts.RData`, and `Script_example.R`.
- Terminal Tab:** Shows the command `git commit -m "Initial commit"`.
- R Markdown Tab:** Shows the command `git push origin master`.
- GitHub Sidebar:** Shows a GitHub repository named "Github — NLP_Course". The sidebar lists files: `Home`, `Desktop`, `NLP_Course`, and `R.codes`. A file named `Script_example.R` is selected, showing a size of 1.1 KB.
- Bottom Navigation:** Includes tabs for `Environment`, `History`, `Connections`, `Git`, and `Tutorial`.

Primeros comandos en Git

Se le hace seguimiento al archivo

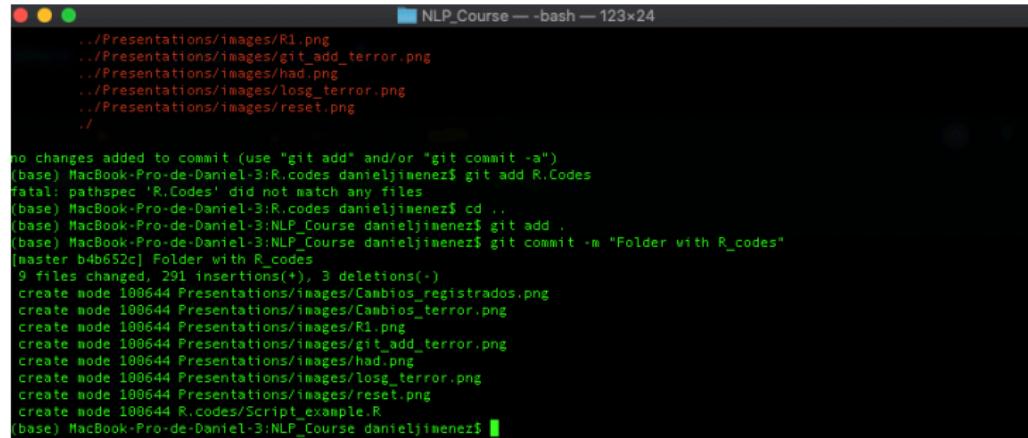
```
(base) MacBook-Pro-de-Daniel-3:R.codes danieljimenez$ touch Script_example.R
(base) MacBook-Pro-de-Daniel-3:R.codes danieljimenez$ code Script_example.R
(base) MacBook-Pro-de-Daniel-3:R.codes danieljimenez$ git status
On branch master
Your branch is behind 'origin/master' by 18 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   ./Presentations/Clase 1.Rmd

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ./Presentations/images/Cambios_terror.png
    ./Presentations/images/R1.png
    ./Presentations/images/git_add_terror.png
    ./Presentations/images/had.png
    ./Presentations/images/lost_terror.png
    ./Presentations/images/reset.png
    ./

no changes added to commit (use "git add" and/or "git commit -a")
(base) MacBook-Pro-de-Daniel-3:R.codes danieljimenez$
```

Primeros comandos en Git



```
.../Presentations/images/R1.png
.../Presentations/images/git_add_terror.png
.../Presentations/images/had.png
.../Presentations/images/loss_terror.png
.../Presentations/images/reset.png
./

no changes added to commit (use "git add" and/or "git commit -a")
(base) MacBook-Pro-de-Daniel-3:R.codes danieljimenez$ git add R.Codes
fatal: pathspec 'R.Codes' did not match any files
(base) MacBook-Pro-de-Daniel-3:R.codes danieljimenez$ cd ..
(base) MacBook-Pro-de-Daniel-3:NLP_Course danieljimenez$ git add .
(base) MacBook-Pro-de-Daniel-3:NLP_Course danieljimenez$ git commit -m "Folder with R_codes"
[master b4b652c] Folder with R_codes
 9 files changed, 291 insertions(+), 3 deletions(-)
 create mode 100644 Presentations/images/Cambios_registrados.png
 create mode 100644 Presentations/images/Cambios_terror.png
 create mode 100644 Presentations/images/R1.png
 create mode 100644 Presentations/images/git_add_terror.png
 create mode 100644 Presentations/images/had.png
 create mode 100644 Presentations/images/loss_terror.png
 create mode 100644 Presentations/images/reset_
 create mode 100644 R.codes/Script_example.R
(base) MacBook-Pro-de-Daniel-3:NLP_Course danieljimenez$
```

Primeros comandos en Git

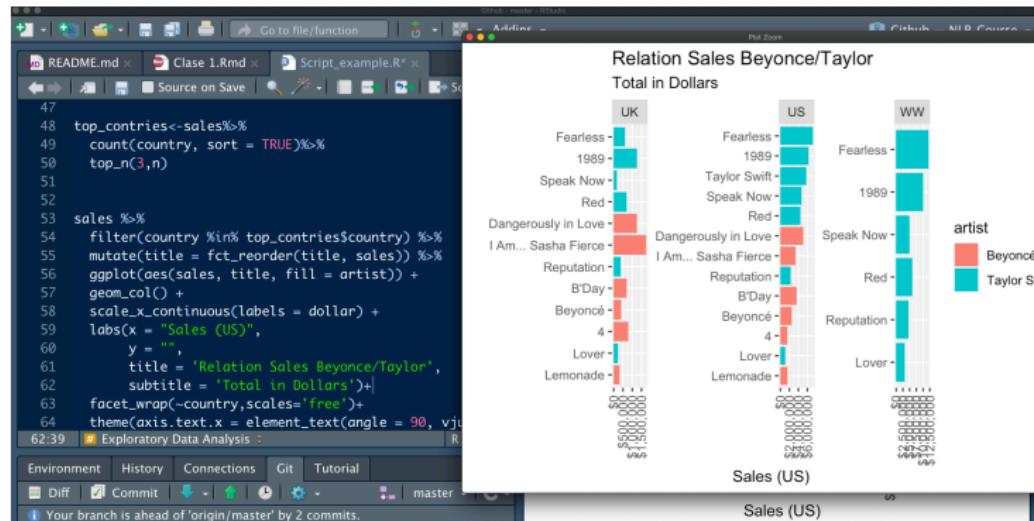
Ahora desarrollamos un poco de código y lo agregamos al repositorio

The screenshot shows an RStudio interface with the following details:

- Script Pane:** Displays an R script titled "Exploratory Data Analysis". The code filters sales data for the US, reorders it by title, and creates a horizontal bar chart comparing sales between Beyoncé and Taylor Swift across various albums.
- Console Pane:** Shows the command used to generate the chart: `ggplot(aes(sales, title, fill = artist)) + geom_col() + scale_x_continuous(labels = dollar) + labs(x = "Sales (US)", y = "", title = 'Relation Sales Beyonce/Taylor', subtitle = 'Total in Dollars')`.
- Viewer Pane:** Displays the resulting horizontal bar chart titled "Relation Sales Beyonce/Taylor Total in Dollars". The x-axis represents "Sales (US)" from \$0 to \$8,000,000. The y-axis lists album titles. The legend indicates two artists: Beyoncé (represented by red bars) and Taylor Swift (represented by teal bars). The chart shows Taylor Swift's sales generally exceeding Beyoncé's for most albums.
- Environment, History, Connections, Git, Tutorial panes:** Standard RStudio navigation and development tools.

Primeros comandos en Git

Ahora probemos con ver estas ventas en varios países, y guardemos los cambios.



Primeros comandos en Git

Se anexan los cambios y se validan con git diff

```
(base) MacBook-Pro-de-Daniel-3:R.codes danieljmenez$ git diff
diff --git a/R.codes/Script_example.R b/R.codes/Script_example.R
index c0ffef0..9bb1f31 100644
--- a/R.codes/Script_example.R
+++ b/R.codes/Script_example.R
@@ -62,3 +62,5 @@ sales %>%
  subtitle = 'Total in Dollars')+
  facet_wrap(~country,scales='free',nrow = 3)+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
(base) MacBook-Pro-de-Daniel-3:R.codes danieljmenez$
```

Primeros comandos en Git

No se les olvide hacer el commit para validar los cambios

```
(base) MacBook-Pro-de-Daniel:3:R.codes danieljimenez$ git commit -m "Alquito de EDA en R"
[master e64517a] Alquito de EDA en R
 2 files changed, 2 insertions(+), 2 deletions(-)
(base) MacBook-Pro-de-Daniel:3:R.codes danieljimenez$
```

Primeros comandos en Git

Ahora suponga que quiere agregar este código

```
sales %>%
  filter(country %in% c("World", "WW")) %>%
  mutate(title = fct_reorder(title, sales)) %>%
  ggplot(aes(sales, title, fill = artist)) +
  geom_col() +
  scale_x_continuous(labels = dollar) +
  labs(x = "Sales (World)",
       y = "")
```

Primeros comandos en Git

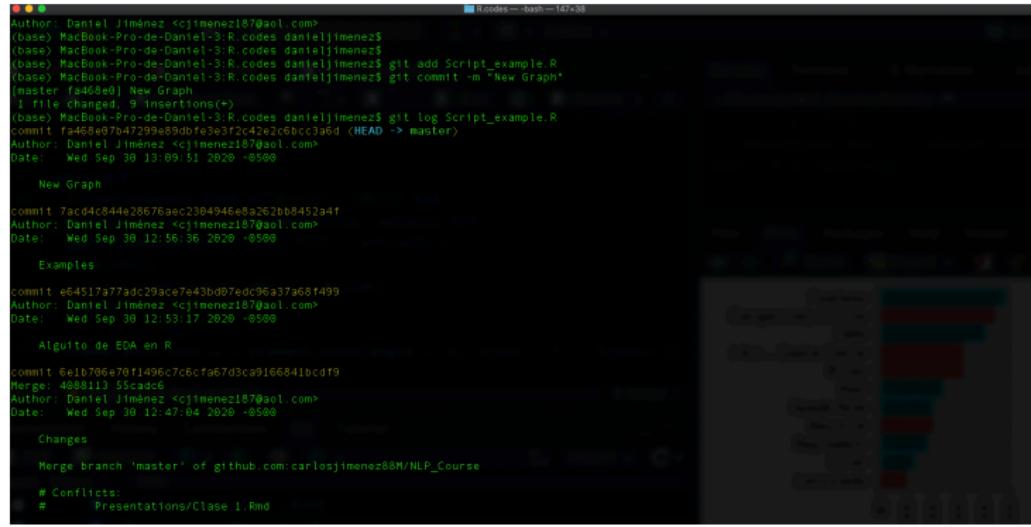
The screenshot shows the RStudio interface. In the top-left pane, there are three open files: README.md, Clase 1.Rmd, and Script_example.R. The Script_example.R file contains R code for creating a bar chart. In the top-right pane, the 'Console' tab shows the command `y = "))+` followed by a partially visible command for theme settings. The bottom-right pane displays a horizontal bar chart titled 'Sales (World)' comparing album sales for Beyoncé and Taylor Swift across various albums. The legend indicates 'artist' with red for Beyoncé and teal for Taylor Swift. The chart shows Beyoncé's sales are generally higher than Taylor Swift's.

| Album | Artist | Sales (World) |
|----------------------|--------------|---------------|
| Fearless | Beyoncé | \$10,000,000 |
| Dangerously in Love | Beyoncé | \$9,000,000 |
| 1989 | Taylor Swift | \$8,500,000 |
| I Am... Sasha Fierce | Beyoncé | \$8,000,000 |
| B'Day | Taylor Swift | \$7,500,000 |
| Red | Taylor Swift | \$7,000,000 |
| Speak Now | Beyoncé | \$6,500,000 |
| Beyoncé | Beyoncé | \$6,000,000 |
| Reputation | Taylor Swift | \$5,500,000 |
| Lover | Beyoncé | \$5,000,000 |
| Lemonade | Beyoncé | \$4,500,000 |

muy bien !!! Ahora anexemos los cambios

Esto se ve

Primeros comandos en Git



```
Author: Daniel Jiménez <cjmenez187@aol.com>
(base) MacBook-Pro-de-Daniel:3 R.codes danieljimenez$ 
(base) MacBook-Pro-de-Daniel:3 R.codes danieljimenez$ 
(base) MacBook-Pro-de-Daniel:3 R.codes danieljimenez$ git add Script_example.R
(base) MacBook-Pro-de-Daniel:3 R.codes danieljimenez$ git commit -m "New Graph"
[master f4d8e0] New Graph
 1 file changed, 9 insertions(+)
(base) MacBook-Pro-de-Daniel:3 R.codes danieljimenez$ git log Script_example.R
commit f4d8e07b47299e89dbfe3e3f2c42ezc6bbc3a6d (HEAD -> master)
Author: Daniel Jiménez <cjmenez187@aol.com>
Date:   Wed Sep 30 13:09:51 2020 +0500

  New Graph

commit 7acd4c844e28676ae2304946e8a262bb8452a4f
Author: Daniel Jiménez <cjmenez187@aol.com>
Date:   Wed Sep 30 12:56:36 2020 +0500

  Examples

commit e64517a77adc29ace7e43bd07edc96a37a68f499
Author: Daniel Jiménez <cjmenez187@aol.com>
Date:   Wed Sep 30 12:53:17 2020 +0500

  Algoritmo de EDA en R

commit 6e1b786a78f1496c7c6cfa67d3ca9166841bcd19
Merge: 4888113 55cad6
Author: Daniel Jiménez <cjmenez187@aol.com>
Date:   Wed Sep 30 12:47:04 2020 +0500

  Changes

  Merge branch 'master' of github.com:carlosjmenez88/M/NLP_Course

  # Conflicts:
  #   Presentations/Clase 1.Rmd
```

Primeros

comandos en Git

Ahora como ustedes son sofisticados, seguro se preguntaran ¿Y Donde vemos los cambios en tamaño, bytes y de más?

A lo cual les dejare un código y una imagen

- git log + – stat

Primeros comandos en Git

Examples

```
Presentations/Clase 1.Rmd          | 13 ++++++++--  
Presentations/images/Anexo de cambios.png | Bin 0 -> 82466 bytes  
Presentations/images/cambiosr3.png    | Bin 0 -> 260570 bytes  
R.codes/Script_example.R            | 2 ++  
4 files changed, 14 insertions(+), 1 deletion(-)
```

ADIVINEN

Tarea

- Como unidad tendran que crear un Git donde trabajaremos un proyecto en conjunto.
- Suba un documento, un texto, el que ustedes quieran, no rollo por ello.
- Ahora creen un Script e intenten leer ese documento.
- Haganle cambios de palabras , investiguen como, yo dejaré esto por acá y me retirare lentamente

```
gsub('palabra','palabra a cambiar',  
      donde_se_encuentra_la_palabra)
```

ADIVINEN

Está tarea se debe hacer antes de la siguiente clase!!!

Flujo de Trabajo

Cuando se trabaja de forma colaborativa, los repositorios no pueden estar de forma local y es por ello que se trabaja con **Github** (hay otras alternativas pero no hablaremos de ellas en esta clase!)

Cuando se trabaja con un servidor remoto se trabaja no con init sino con **git clone** del repositorio original.

Flujo de Trabajo

Para el flujo de trabajo, por lo general se trabaja con ramas (branch), con lo cual se busca asignar partes de proyecto a miembros del equipo, para así después hacerle fusión con **merge**

Para esto es necesario entender lo siguiente:

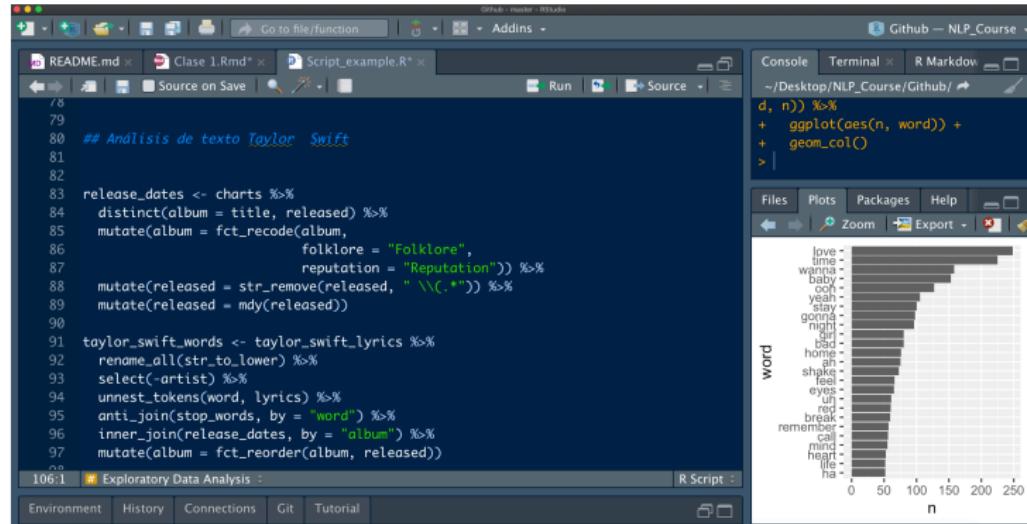
- Cuando se clona el repositorio git clone + ID,
- En el local se crea un nuevo staging el cual se le puede hacer seguimiento ,
- En cada commit, se puede evaluar el bugfix del proceso.

Flujo de Trabajo

Con las ramas, es necesario poder generar versionamiento sobre los cambios y es por ello que ya no se usará (en esta fase) el git add , sino el **git commit -am**

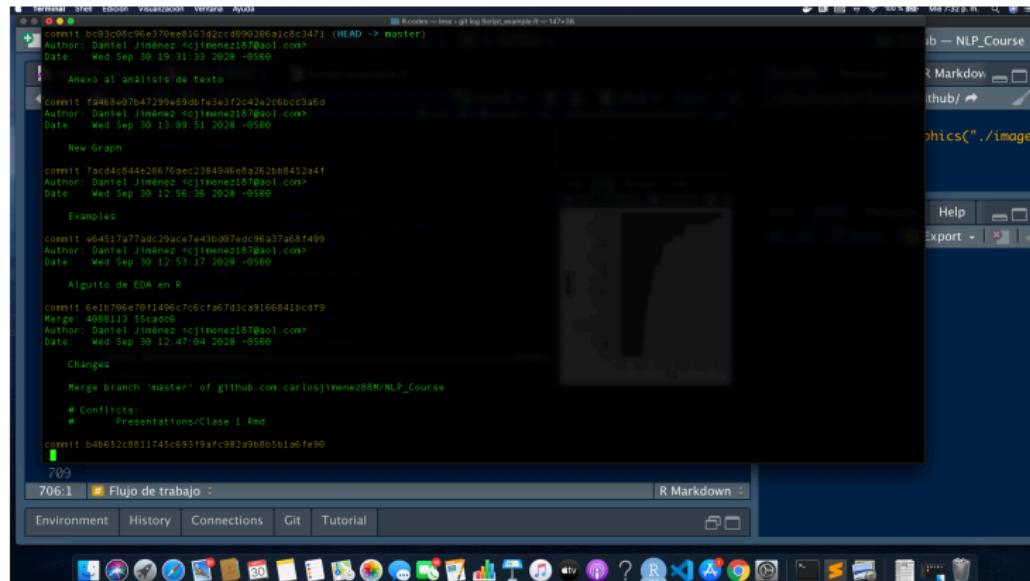
Flujo de trabajo

Suponga que anexo un cambio al script y analizo los textos de Taylor Swift



Flujo de trabajo

Se evalua el cambio



```
git clone https://github.com/carlosjimenez88M/NLP_Course.git
cd NLP_Course
git log --oneline --graph --all --date=short
commit b091c0596e370ee8f63d2cc8992866a1c3471 (HEAD -> master)
Author: Daniel Jiménez <jimenez187@aol.com>
Date:   Wed Sep 30 19:31:33 2020 -0500

    Anexo al análisis de texto

commit f4d48e07047299e890fbfa3f2c42a2c6b0cc3a6d
Author: Daniel Jiménez <jimenez187@aol.com>
Date:   Wed Sep 30 13:09:51 2020 -0500

    New Graph

commit 7a7d4c844c8676aec238494fe5a262bb45724f
Author: Daniel Jiménez <jimenez187@aol.com>
Date:   Wed Sep 30 12:56:36 2020 -0500

    Examples

commit ee4517a77adc29ac7e43bd97edc96a37a687499
Author: Daniel Jiménez <jimenez187@aol.com>
Date:   Wed Sep 30 12:53:17 2020 -0500

    Algoritmo de EDU en R

commit 6e1b706e70f14967c6cf9a67d3ca9166841bcdf9
Merge: 4088113 5cad65
Author: Daniel Jiménez <jimenez187@aol.com>
Date:   Wed Sep 30 12:47:04 2020 -0500

    Changes

    Merge branch 'master' of https://github.com/carlosjimenez88M/NLP_Course
    # Conflicts:
    #   Presentations/Clase 1.Rmd

commit b4b652c9811745c693f9afc982a9b8b5b1a67e90
709
706:1 Flujo de trabajo : R Markdown
```

Flujo de trabajo

Ahora se crea la rama, para eso debemos ir a donde vive el archivo

- Verifique el Status git status;
- Se crea la rama con git branch + nombre de la rama

```
(base) MacBook-Pro-de-Daniel:3 R codes danieljimenez$ git branch beyonce
(base) MacBook-Pro-de-Daniel:3 R codes danieljimenez$ git show Script_example.R
(base) MacBook-Pro-de-Daniel:3 R codes danieljimenez$ git show
commit f6145ecd73401afdf7af11e2c6e41ef2661b9c76a (HEAD -> master, origin/master, origin/HEAD, beyonce)
Author: Daniel Jiménez <cjimenez187@aol.com>
Date:   Wed Sep 30 19:39:36 2020 -0500

    some changes
```

Flujo de trabajo

Ahora con git checkout + nombre de la rama, nos movemos a la rama creada

```
(base) MacBook-Pro-de-Daniel-3:R.codes danieljimenez$ git checkout beyonce
M      Presentations/Clase 1.Rmd
Switched to branch 'beyonce'
(base) MacBook-Pro-de-Daniel-3:R.codes danieljimenez$ git status
On branch beyonce
  Changes to be committed:
    (use "git add <file>" to include in what will be committed)

        modified:   Presentations/Clase 1.Rmd
```

Flujo de trabajo

Ahora suponga que se hicieron unos cambios en la rama beyonce

The screenshot shows the RStudio interface with two tabs open: 'Script_example.R' and 'Console'. The 'Script_example.R' tab contains R code for data analysis and visualization. The 'Console' tab shows the output of the R code, which includes a bar chart comparing word usage between Beyoncé and Taylor Swift. The chart has two legends: 'Beyoncé' (red) and 'Taylor Swift' (teal). The x-axis represents the relative use of words, and the y-axis lists various words.

```
136 filter(artist == "Beyonce") %>%
137 slice_max(num_words_total, n = 100, with_ties = FALSE) %>%
138 slice_max(abs(log_odds_weighted), n = 25, with_ties = FALSE) %>%
139 mutate(word = fct_reorder(word, log_odds_weighted)) %>%
140 mutate(direction = ifelse(log_odds_weighted > 0, "Beyoncé", "Taylor Swift"))
141 word_differences %>%
142 ggplot(aes(log_odds_weighted, word, fill = direction)) +
143 geom_col() +
144 scale_x_continuous(breaks = log2 ^ seq(-6, 9, 3)),
145           labels = paste0(2 ^ abs(seq(-6, 9, 3)), "X")) +
146 labs(x = "Relative use in Beyoncé vs Taylor Swift (weighted)",
147       y = "",
148       title = "Which words most distinguish Beyoncé and Taylor Swift songs?",
149       subtitle = "Among the 100 words most used by the artists (combined)",
150       fill = "")
```

Exploratory Data Analysis : R Script

Environment History Connections Git Tutorial

Diff Commit Pull Push | beyonce | C |

Staged Status Path

Presentations/Clase 1.Rmd

Github — beyonce - RStudio

Console Terminal R Markdown

~/Desktop/NLP_Course/Github/

words most used by the artists (combined),
+ fill = "")
> |

Files Plots Packages Help

Zoom Export

Which words most distinguish Beyoncé and Taylor Swift songs?
Among the 100 words most used by the artists (combined)

The chart displays the relative use of words in Beyoncé's and Taylor Swift's songs. The x-axis represents the relative use, ranging from approximately -6 to 9. The y-axis lists words. Red bars represent Beyoncé, and teal bars represent Taylor Swift. The chart shows that Beyoncé uses words like 'body', 'babe', 'slay', 'twerk', 'bitch', 'drip', 'forever', 'home', 'holla', 'broke', 'remember', 'warning', 'toss', 'sochi', 'stun', 'stay', and 'time' more frequently than Taylor Swift. Taylor Swift uses words like 'bey', 'babe', 'slay', 'twerk', 'bitch', 'drip', 'forever', 'home', 'holla', 'broke', 'remember', 'warning', 'toss', 'sochi', 'stun', 'stay', and 'time' more frequently than Beyoncé.

Relative use in Beyoncé vs Taylor Swift (weight)

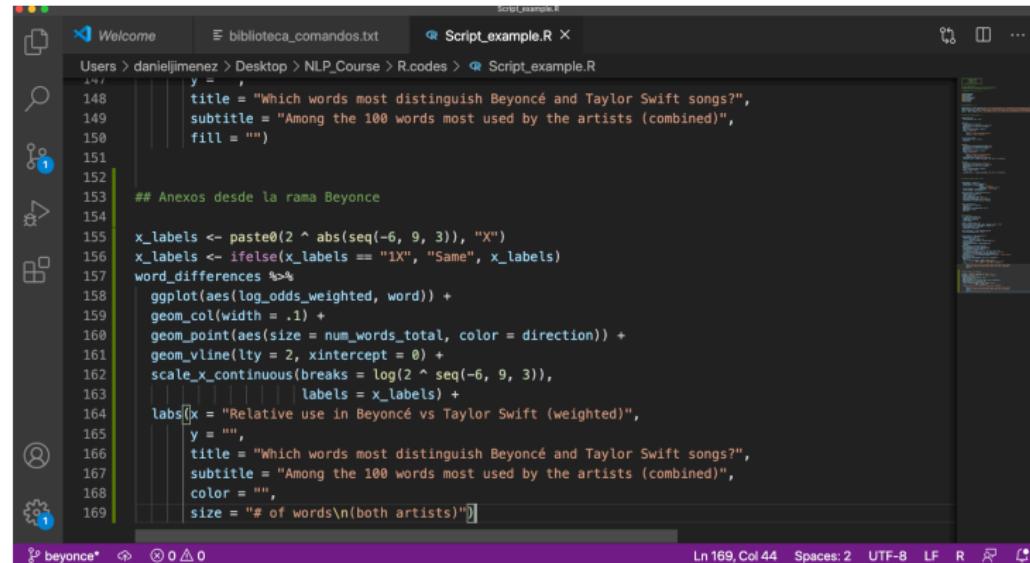
Flujo de trabajo

Se anexan los cambios

```
base) MacBook-Pro-de-Daniel-3:R.codes danieljimenez$ git add Script_example.R
base) MacBook-Pro-de-Daniel-3:R.codes danieljimenez$ git commit -m "Comparativa entre las cantantes
"
beyonce f385d0f] Comparativa entre las cantantes
2 files changed, 48 insertions(+)
base) MacBook-Pro-de-Daniel-3:R.codes danieljimenez$
```

Flujo de trabajo

Ahora toca hacer un merge, en donde la rama se une a la ruta final. Pero suponga lo siguiente : Agregue un segmento de código



The screenshot shows the RStudio interface with the following details:

- Title Bar:** Shows "Welcome", "biblioteca_comandos.txt", and "Script_example.R".
- Code Editor:** Displays an R script named "Script_example.R". The code includes sections for setting plot parameters and creating a ggplot object. A specific line of code is highlighted with a red box:

```
147     y = '',
148     title = "Which words most distinguish Beyoncé and Taylor Swift songs?",
149     subtitle = "Among the 100 words most used by the artists (combined)",
150     fill = "")
```



```
151
152 ## Anexos desde la rama Beyonce
153
154 x_labels <- paste0(2 ^ abs(seq(-6, 9, 3)), "X")
155 x_labels <- ifelse(x_labels == "1X", "Same", x_labels)
156 word_differences %>%
157   ggplot(aes(log_odds_weighted, word)) +
158   geom_col(width = .1) +
159   geom_point(aes(size = num_words_total, color = direction)) +
160   geom_vline(lty = 2, xintercept = 0) +
161   scale_x_continuous(breaks = log(2 ^ seq(-6, 9, 3)),
162                      labels = x_labels) +
163   labs(x = "Relative use in Beyoncé vs Taylor Swift (weighted)",
164        y = "",
165        title = "Which words most distinguish Beyoncé and Taylor Swift songs?",
166        subtitle = "Among the 100 words most used by the artists (combined)",
167        color = "",
168        size = "# of words\n:both artists")
```
- Status Bar:** Shows "Ln 169, Col 44" and other file statistics.

Flujo de trabajo

Vea que los cambios no se ven en el código general, por lo cual hay que hacer un merge con git merge + nombre de la rama

The screenshot shows an RStudio interface. On the left, the code editor displays `Script_example.R` with the following R code:

```
155 x_labels <- paste0(2 ^ abs(seq(-6, 9, 3)), "X")
156 x_labels <- ifelse(x_labels == "1X", "Same", x_labels)
157 word_differences %>%
158   ggplot(aes(log_odds_weighted, word)) +
159   geom_col(width = .1) +
160   geom_point(aes(size = num_words_total, color = direction)) +
161   geom_vline(lty = 2, xintercept = 0) +
162   scale_x_continuous(breaks = log(2 ^ seq(-6, 9, 3)),
163                      labels = x_labels) +
164   labs(x = "Relative use in Beyoncé vs Taylor Swift (weighted)",
165        y = "",
166        title = "Which words most distinguish Beyoncé and Taylor Swift songs?",
167        subtitle = "Among the 100 words most used by the artists (combined)",
168        color = "",
169        size = "# of words\n(both artists)")
170
```

The R console window shows the command `ggplot(aes(log_odds_weighted, word))`. To the right, a bubble chart titled "Which words most distinguish Beyoncé and Taylor Swift" displays the relative use of words between the two artists. The x-axis represents the relative use (logarithmic scale from -6 to 9), and the y-axis lists words. The size of the bubbles indicates the number of words used by both artists, with a legend showing sizes for 400, 800, 1200, and 1600. Red bubbles represent Beyoncé, and teal bubbles represent Taylor Swift.

tenemos el Merge

) Ahora

Flujo de trabajo

Ahora notesé que se pueden ver los logs que se hicieron entre las ramas!!

Queda pendiente para su investigación como resolver un problema de conflictos de versiones.

Un consejo, los conflictos solucionelos desde Visual Studio Code.

Introducción a Github

Github: Es un servidor que permite colaborar en el desarrollo de códigos.
En breve palabras es la red social de los programadores.

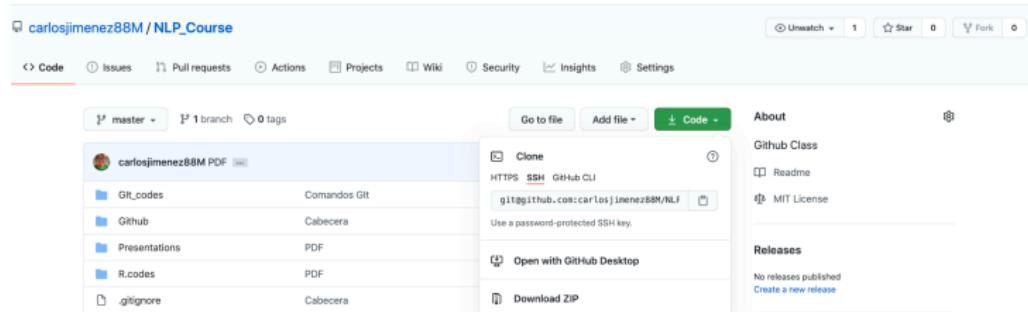
Por favor acceda al siguiente link y registrese <https://github.com/>

Introducción a Github

- Cree un repositorio;
- Integre sus cambios;
- Subalo (git push)

Introducción a Github

El repositorio se clona desde el parámetro code



Introducción a Github

Para hacer el trabajo colaborativo es necesario crear llaves SSH, estas se dividen en :

- Llave pública, es la que se comparte para que puedas acceder al repositorio;
- Llave privada: Se usa para cifrar el mensaje enviado!

Introducción a Github

- La llave ssh se crea de la siguiente manera : `ssh-keygen -t rsa -b 4096 -C "tu@email.com"`
- Para encender el servidor eval `$(ssh-agent -s)`
- Para añadir la llave `ssh-add ruta-donde-guardaste-tu-llave-privada`
 - Si es MAC use el siguiente comando `Host * AddKeysToAgent yes UseKeychain yes IdentityFile ruta-donde-guardaste-tu-llave-privada`
 - `ssh-add -K ruta-donde-guardaste-tu-llave-privada`

Introducción a Github

Después de crear las llaves:

- En github ve a settings;
- Ve a SSH;
- Agregue la llave
- Ahora escriba el siguiente código: `git remote set-url origin + ssh git`

Introducción a Github

Para fusionar bien los repositorios :

- Descargue el repositorio con git pull
- Use git pull origin master
- Verifique el Status.
- Haga cambios y envielos con git push origin master

Introducción a Github

Ahora verifique la historia del repositorio con el siguiente código : git log --all --decorate --oneline

```
(base) MacBook-Pro-de-Daniel-3:R.codes danieljimenez$ git push origin master
Everything up-to-date
(base) MacBook-Pro-de-Daniel-3:R.codes danieljimenez$ git log --all --decorate --oneline
716c45d (HEAD -> beyonce) Algunos cambios
7fe265c Cambios para generar conflicto
2de6bc5 (origin/master, origin/HEAD, master) PDF
c2fa8bf Creación de ramas
ae34a49 changes
ff3020a Anexos al código desde la rama Beyonce
09642b9 Cambios
391265f gggg
f385d0f Comparativa entre las cantantes
08edac9 Anexos Branch
7a8d810 Changes
70145ec some changes
0c83c08 Anexo al análisis de texto
3ded661 Comandos GIT
f169d29 clase1
fad68e0 New Graph
7cbf65b Presentation
7acd4c8 Examples
e64517a Algoito de EDA en R
c0b870a Changes 2
5e10706 Changes
0088113 Taylor Changes
b40652c Folder with R_codes
55cad6 Git diff
cf6b786 Documentos para la clase
f34bc04 Cnaches in Presentation
ca72add Anexo Códigos
bae7f8a Cambios
19173b3 Orden en el código
7fc1ef1 Cambios en el repositorio
095ac3d Changes
762727d Changes 3
117fd6 Changes
dadaa178 Update 2
fabf67c Update Format Class
```

Introducción a Github

- El tag es una etiqueta que se le asigna a las versiones de los commits.
- Para crear un tag copie el nombre del commit en el cual va hacer las etiquetas `git tag -a v.0.1 -m "Cambios en las diapositivas" 716c45d`
- Para ver los tags use el comando `git show-ref --tags`
- Para subir los cambios use `git push origin --tags`

Introducción a Github

The screenshot shows a GitHub repository page for 'carlosjimenez88M / NLP_Course'. The 'Code' tab is selected, showing a list of commits. The first commit is by 'carlosjimenez88M Cambios' at 493de87, 5 minutes ago, with 67 commits. Other files listed include 'Git_codes', 'GitHub', 'Presentations', 'R.codes', '.gitignore', 'LICENSE', and 'README.md'. To the right, sections for 'About', 'Releases', and 'Packages' are visible, along with a '##' symbol.

| File | Commit Message | Time Ago |
|---------------|----------------------|---------------|
| Git_codes | Comandos Git | 4 hours ago |
| Github | Cabecera | 12 days ago |
| Presentations | Cambios | 5 minutes ago |
| R.codes | Cambios | 5 minutes ago |
| .gitignore | Cabecera | 12 days ago |
| LICENSE | Initial commit | 13 days ago |
| README.md | Algunito de EDA en R | 10 hours ago |

Introducción a Github

Para eliminar los tags

- Use git tag para saber cuales son los tags creados;
- Use git tag -d nombre del tag para eliminarlo.
- Envie los cambios con el comando git push origin :refs/tags/v.0.1

Introducción a Github

Para manejar las ramas de desarrollo es necesario :

- Use el comando git show-branch
- Use git push origin y el nombre de la rama

Introducción a Github