



# Github

**{desafío}**  
latam\_



**Activen las cámaras los que puedan y  
pasemos asistencia**





# Inicio

{desafío}  
latam\_



/\* Realizar operaciones de navegación de directorios, usando los comandos básicos del terminal, para crear y manipular archivos y directorios.\*/

/\* Aplicar las etapas del versionamiento de GIT, para mantener un repositorio de versiones.\*/

/\* Aplicar el procedimiento de subida del código versionado mediante una conexión SSH, para la mantención de un repositorio remoto.\*/

## Objetivo



# Desarrollo

{desafío}  
latam\_



# Github

## ¿Qué es Github?

Es una red social de código.

En github podemos descargar millones de proyectos, compartir nuestro código y trabajar en equipo.



# Trabajando con Github

*¿Qué necesitamos?*



*Terminal*

*para ejecutar comandos*



*Git*

*Para controlar versiones*



*Github*

*Para subir y descargar proyectos*

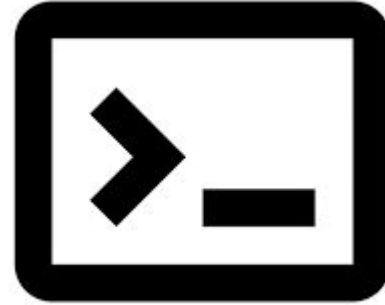


**/\* Introducción a terminal \*/**

# Terminal

*¿Qué es el terminal?*

El terminal, es una poderosa herramienta donde podemos utilizar líneas de comandos para navegar por archivos y directorios. Al mismo tiempo, se utiliza para interactuar con programas que no tienen interfaz gráfica.



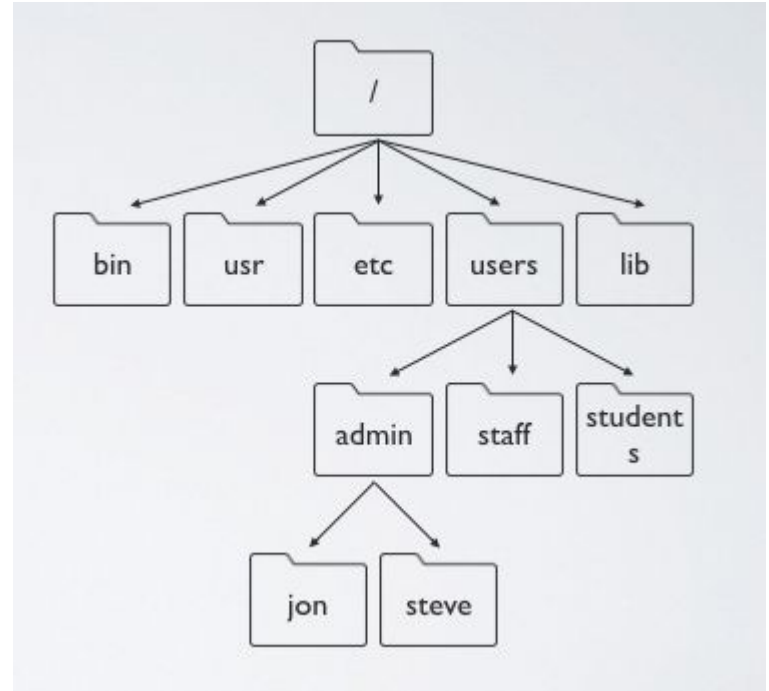
# Inicialización de terminal

- En Linux: Presiona ctrl + alt + t.
- En Mac: Presiona ⌘ + espacio, busca por spotlight terminal.
- En Windows: Presiona inicio (tecla de windows) + r, escribe “cmd” en la caja de texto y presiona aceptar.

# Utilizando el terminal

## *Explicación de las estructuras de directorio*

El árbol de directorios comienza en la raíz y contiene ramas o directorios, al mismo tiempo que al interior de estos directorios pueden existir archivos u otros directorios.



# Utilizando el terminal

*Conocer en qué directorio estamos (pwd)*

Escribe en tu terminal las letras **pwd** y presiona enter:

```
ale@ale-Lenovo-G40-80:~$ pwd  
/home/ale  
ale@ale-Lenovo-G40-80:~$
```

# Utilizando el terminal

## *Listar archivos (ls)*

El comando **ls** (listar) muestra una lista de los archivos y directorios contenidos en el directorio en el que se está ejecutando el comando.

```
ale@ale-Lenovo-G40-80:~$ ls
Descargas  Documentos  Escritorio  Imágenes  Música
Plantillas  Público    snap       Steam     Vídeos
ale@ale-Lenovo-G40-80:~$
```

# Utilizando el terminal

## *Anatomía de un comando*

Todos los comandos tienen un nombre que los distingue, por ejemplo **ls** y **pwd**, serían el nombre del comando. Hay comandos que pueden recibir opciones.

Las opciones las especificamos anteponiendo - o -- al igual que con el comando ls que ya habíamos realizado, es decir, ls -a. En este caso el -a da la opción de ver archivos ocultos.

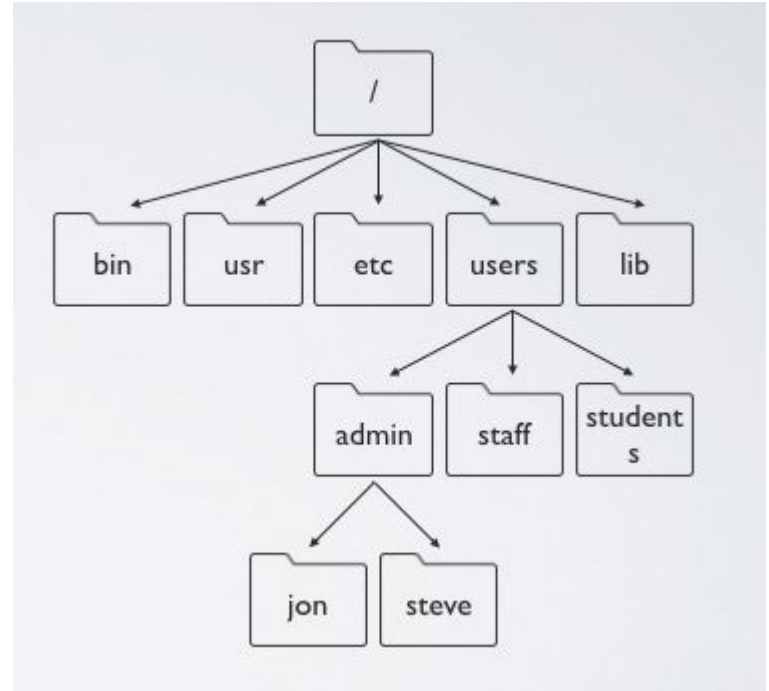


**Nota:** En linux y osx los archivos que empiezan con puntos (.) quedan ocultos, por ejemplo: ejemplo\_ .secreto

# Utilizando el terminal

## *Comandos de navegación entre directorios (cd)*

- El comando llamado `cd` (change directory) nos permite movernos entre los directorios.
- Podemos hacerlo a un directorio padre con `cd`.
- Y para un directorio hijo utilizamos `cd nombre_directorio`





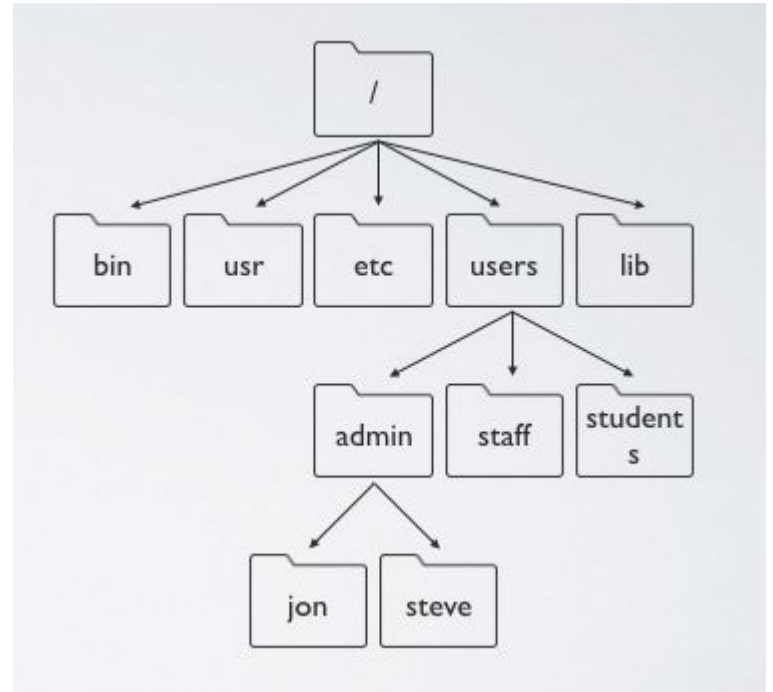
# Utilizando el terminal

## Explicación de las estructuras de directorio

Desde **users** podemos movernos directamente a **jon** indicando:

**cd admin/jon**

Luego, podríamos movernos al directorio admin con **cd**.



# Utilizando el terminal

## Creando una carpeta

Para poder crear una carpeta utilizando la terminal debemos utilizar el comando **mkdir**. Este comando va acompañado del nombre que le daremos a la carpeta.

En este ejemplo, accedimos al escritorio utilizando el comando **cd** y dentro del escritorio creamos la carpeta llamada *terminal*.

```
ale@ale-Lenovo-G40-80:~$ cd Escritorio/  
ale@ale-Lenovo-G40-80:~/Escritorio$ mkdir terminal
```

Una vez creada la carpeta pueden utilizar el comando **ls** para visualizarla en la terminal, o accediendo al escritorio.

# Utilizando el terminal

## *Creando un archivo*

Una vez creada la carpeta crearemos un archivo html usando el comando **touch**, acompañado del nombre del archivo y su extensión.

Primero debemos entrar a la carpeta recién creada utilizando el comando **cd** y una vez dentro, escribimos en la terminal **touch index.html**

```
ale@ale-Lenovo-G40-80:~/Escritorio$ cd terminal
ale@ale-Lenovo-G40-80:~/Escritorio/terminal$ touch index.html
ale@ale-Lenovo-G40-80:~/Escritorio/terminal$ ls
index.html
```

/\* Realizar operaciones de navegación de directorios, usando los comandos básicos del terminal, para crear y manipular archivos y directorios.\*/ ✓

/\* Aplicar las etapas del versionamiento de GIT, para mantener un repositorio de versiones.\*/

/\* Aplicar el procedimiento de subida del código versionado mediante una conexión SSH, para la mantención de un repositorio remoto.\*/

## Objetivo

**/\* Introducción a Git \*/**

# Introducción a Git

## *Ventajas*

Git es un sistema de control de versiones ampliamente utilizado en el desarrollo.



# Introducción a Git

*¿Cuándo debemos usar git?*

La recomendación es usarlo siempre que trabajemos desarrollando código ya que nos evitará realizar trabajo extra si ocurre algún problema.



# Introducción a Git

## *Formas de uso de git*

- Existen distintas formas de trabajar con git. Se puede trabajar directamente desde el terminal y algunos editores de texto como VSCode traen incorporado herramientas para utilizarlo.
- Desde el terminal tenemos mucha flexibilidad, así que lo aprenderemos a utilizar de esta forma.



# Configurando Git

## *Configuración del user y el email*

Configurar nuestro usuario en git:

```
git config --global user.name "Tu  
Nombre"
```

```
git config --global user.email  
tucorreo@mail.com
```

# Configurando Git

## *Verificando la configuración*

```
git config --list
```

```
user.name=Nombre Apellido
```

Si ves este mensaje, es porque lo lograste.

```
user.email=micorreo@mail.com
```

# Uso básico de git

## *Inicializando git*

Nuestro primer paso será escribir en la carpeta de nuestro último desafío lo siguiente:

```
git init
```

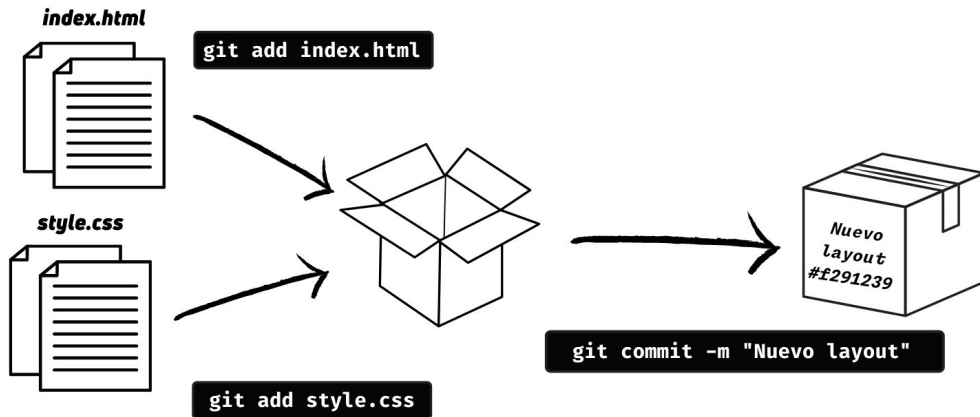
Con ls -a podemos verificar que se creó una carpeta .git que guarda toda la información de las versiones.

# Uso básico de git

## Flujo básico

### Primeros pasos:

- Agregar archivos (o cambios en archivos).
- Confirmar los cambios.



## Practiquemos lo aprendido

- Crea una carpeta nueva en el escritorio.
- Abre el terminal.
- Desde el terminal Ingresa a la carpeta.
- Inicializa git (git init).
- Abre la carpeta en el editor de código.
- Crea el archivo index.html
- Añade el archivo utilizando git add index.html
- Confirma el cambio con git commit -m "primer commit"

## Ejercicio

### ¡Manos al teclado!



# Uso básico de git

*git add*

Agregamos nuestros archivos creados y cambios realizados utilizando un comando llamado git add:

Un archivo

```
git add nombre_archivo
```

Todos los  
archivos

```
git add --all
```

```
git add .
```



Se recomienda añadir los archivos de uno para evitar agregar archivos que no queremos agregar.

## Uso básico de git

### *git commit*

Luego, debemos confirmar estos cambios, que equivale a cerrar la caja y agregarle una etiqueta con una descripción. Esto se logra con:

```
git commit -m "Nombre o descripción del commit"
```

# Uso básico de git

## *Revisando los commits*

Podemos revisar todos los commits de un proyecto con:

```
git log
```

```
commit
5771e50a55e49d1a3897572f6303f69aa05ee1af
(HEAD -> main)
Author: Gonzalo Sánchez
<gonzalo@desafiolatam.com>
Date: Thu Dec 2 10:42:45 2021 -0300
```

```
prueba 1
```



# Uso básico de git

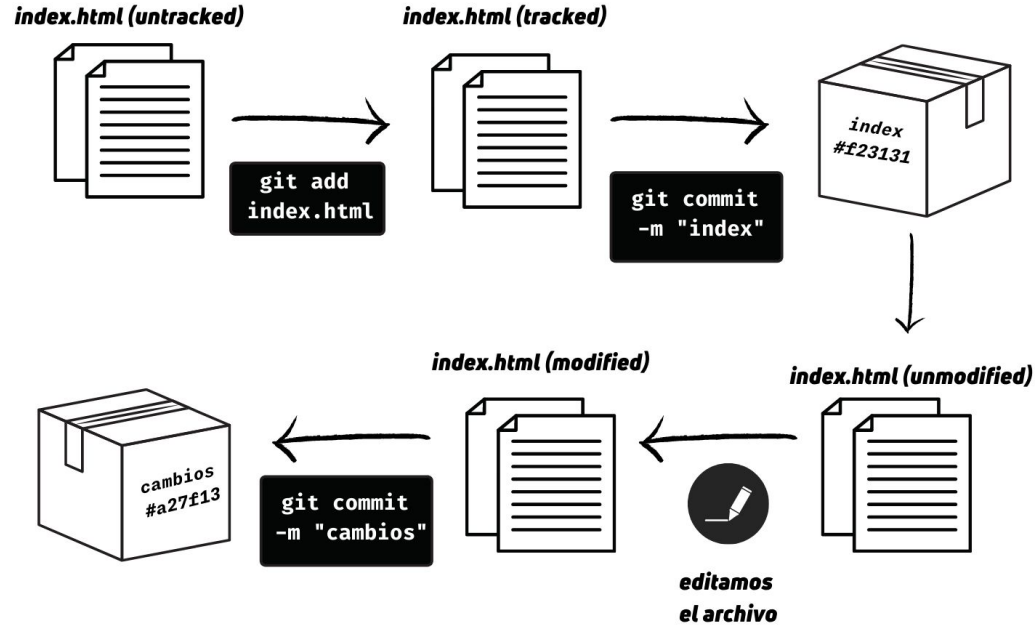
*git add*

En un proyecto nuevo, todos los archivos están en estado untracked (o sea git no revisa si se modifican o no) con git add pasamos un estado de untracked a tracked.

Luego, cuando modificamos un archivo pasa de estado no-modificado a modificado, donde para agregar los cambios tenemos que volver a utilizar git add.

# Uso básico de git

*git add*



# Uso básico de git

## *Revisando el estado y commits*

Podemos revisar el estado de los archivos con git status:

```
git status
```

Podemos revisar todos los commits de un proyecto con:

```
git log
```

## Repitamos el ejercicio utilizando status y log

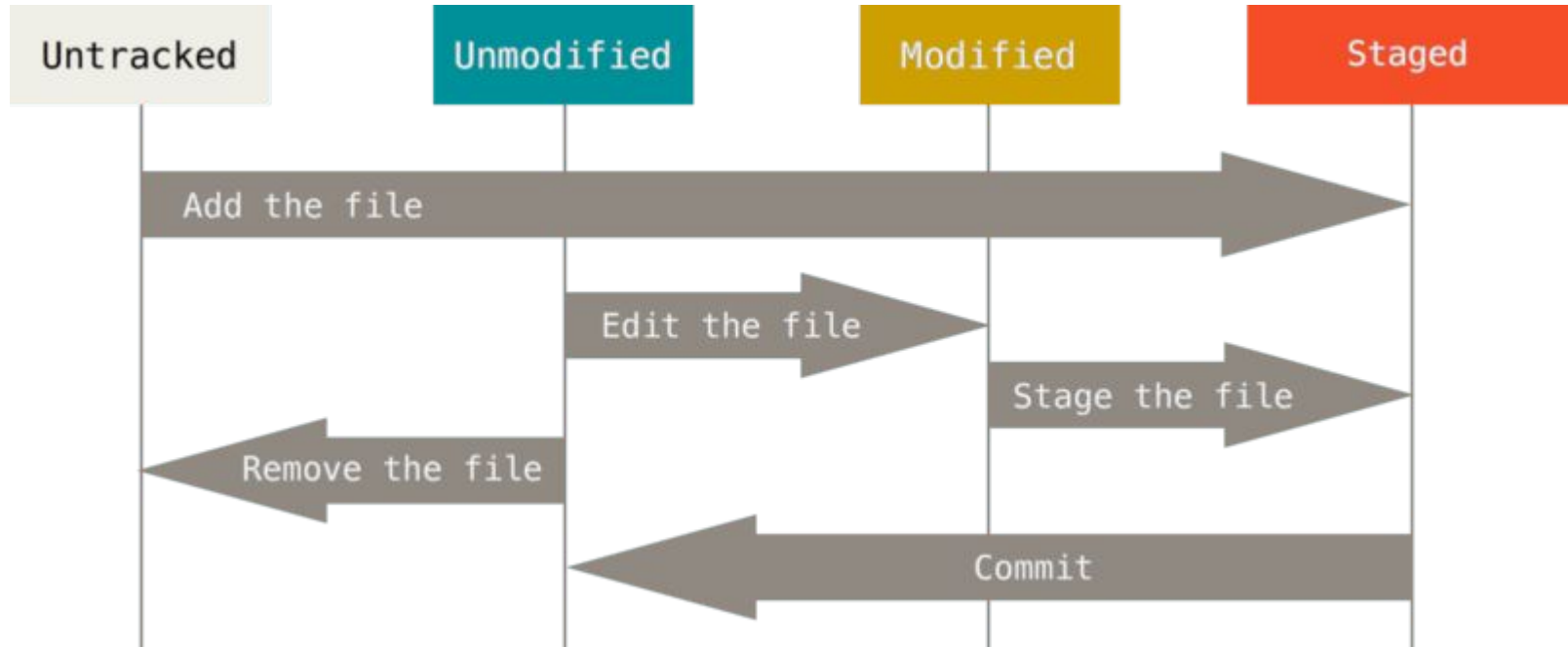
- Crea una carpeta nueva en el escritorio.
- Abre el terminal.
- Desde el terminal Ingresa a la carpeta.
- Inicializa git (git init).
- Abre la carpeta en el editor de código.
- Crea el archivo index.html y ejecuta git status.
- Añade el archivo utilizando git add index.html y ejecuta git status.
- Confirma el cambio con git commit -m "primer commit"
- ejecuta git status.
- ejecuta git log.

## Ejercicio ¡Manos al teclado!



# Uso básico de git

*Estado de un archivos*



/\* Realizar operaciones de navegación de directorios, usando los comandos básicos del terminal, para crear y manipular archivos y directorios.\*/ ✓

/\* Aplicar las etapas del versionamiento de GIT, para mantener un repositorio de versiones.\*/ ✓

/\* Aplicar el procedimiento de subida del código versionado mediante una conexión SSH, para la mantención de un repositorio remoto.\*/

## Objetivo

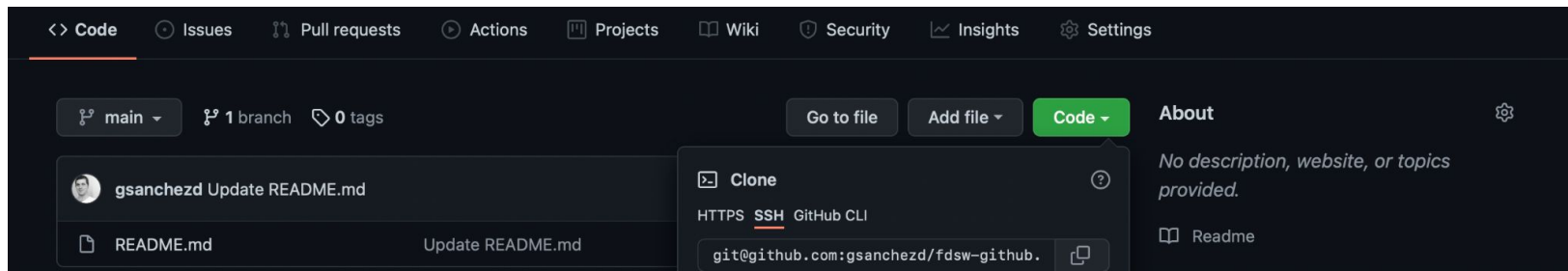
**/\* Introducción a Github \*/**

Para utilizar github necesitamos algunas configuraciones que todavía no tenemos, pero encontraremos en la lectura y cubriremos en la tutoría.





## Descargando un proyecto con git clone



Desde el terminal podemos descargar un proyecto con git clone:

```
git clone ssh(git) [nombre_proyecto]
```



nombre\_proyecto es opcional y es para darle un nombre distinto a la carpeta de como se llama en Github.

Si queremos subir cambios a un proyecto primero tenemos que hacer **fork** desde Github.

Un fork copia el código a nuestro espacio de trabajo dentro de github, desde ahí podemos hacer modificaciones.

Luego, haremos clon de nuevo (hay otra forma de hacerlo sin volver a descargar pero lo estudiaremos en la guía).

Podemos subir todos los cambios **confirmados** escribiendo `git push origin main`.



/\* Realizar operaciones de navegación de directorios, usando los comandos básicos del terminal, para crear y manipular archivos y directorios.\*/ ✓

/\* Aplicar las etapas del versionamiento de GIT, para mantener un repositorio de versiones.\*/ ✓

/\* Aplicar el procedimiento de subida del código versionado mediante una conexión SSH, para la mantención de un repositorio remoto.\*/ ✓

## Objetivo



Cierre

{desafío}  
latam\_



¿Existe algún concepto que no  
hayas comprendido?

Reflexionemos

- Revisar la guía que trabajarán de forma autónoma.
- Indicaciones tutoría.
- Revisar en conjunto el desafío.

¿Qué sigue?



*Academia de  
talentos digitales*

[www.desafiolatam.com](http://www.desafiolatam.com)



/DesafioLatam



/DesafioLatam



/DesafioLatam



/DesafioLatam