

**{desafío}**  
**latam\_**

# Terminal, Git, GitHub y GitHub Pages \_

Sesión Presencial 1



# Itinerario



Activación de conceptos

Desarrollo Desafío

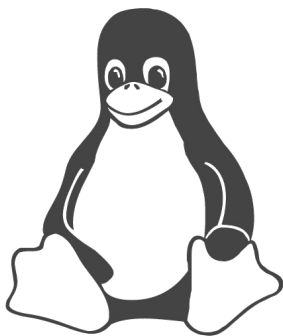
Panel de discusión

# Activación de conceptos

# Terminal

# Iniciar Terminal

## LINUX



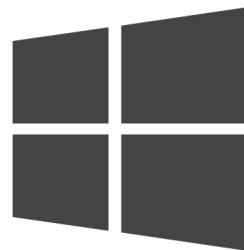
Presiona ctrl + alt + t

## MAC



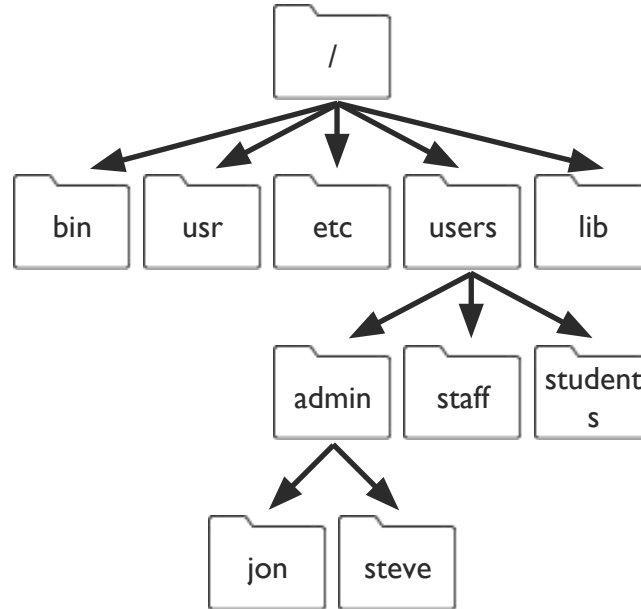
⌘ + espacio, busca por  
spotlight la "terminal"

## WINDOWS



Busca el programa  
"git bash" y ábrelo

# Estructura de Directorios



Comando	Explicación
<code>pwd</code>	Muestra en que directorio te encuentras
<code>cd</code>	Cambia de directorio a uno especificado
<code>cd ..</code>	Permite ir a un directorio anterior al actual
<code>cd /</code>	Lleva a la raíz
<code>cd ~</code>	Lleva al directorio \$home
<code>ls</code>	Muestra todos los archivos y directorios de la carpeta actual
<code>ls -a</code>	Muestra los archivos, los archivos ocultos y los directorios de la carpeta actual
<code>touch</code>	Crea un archivo
<code>mv</code>	Renombra y mueve un archivo
<code>rm</code>	Borra un archivo
<code>mkdir</code>	Crea un directorio
<code>rmdir</code>	Elimina un directorio vacío
<code>rm -r</code>	Elimina todos los archivos de un directorio, incluido el directorio

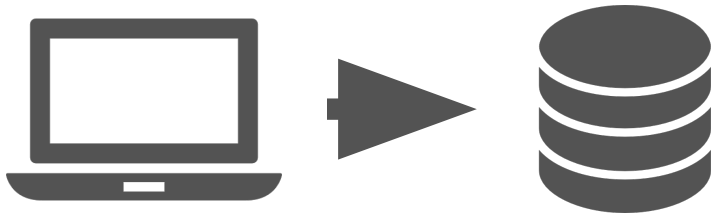
# Git



# Git nos permite:

- Recuperar versiones anteriores de nuestro código
- Recuperar archivos borrados
- Ayudar a gestionar cambios realizados por otras personas
- Administrar un proyecto donde trabajan múltiples desarrolladores

# ¿Qué es “control de versiones”?



Es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante.

# Formas de utilizar Git



**GIT**



Editores de  
texto



Programas de  
versionamiento



Terminal

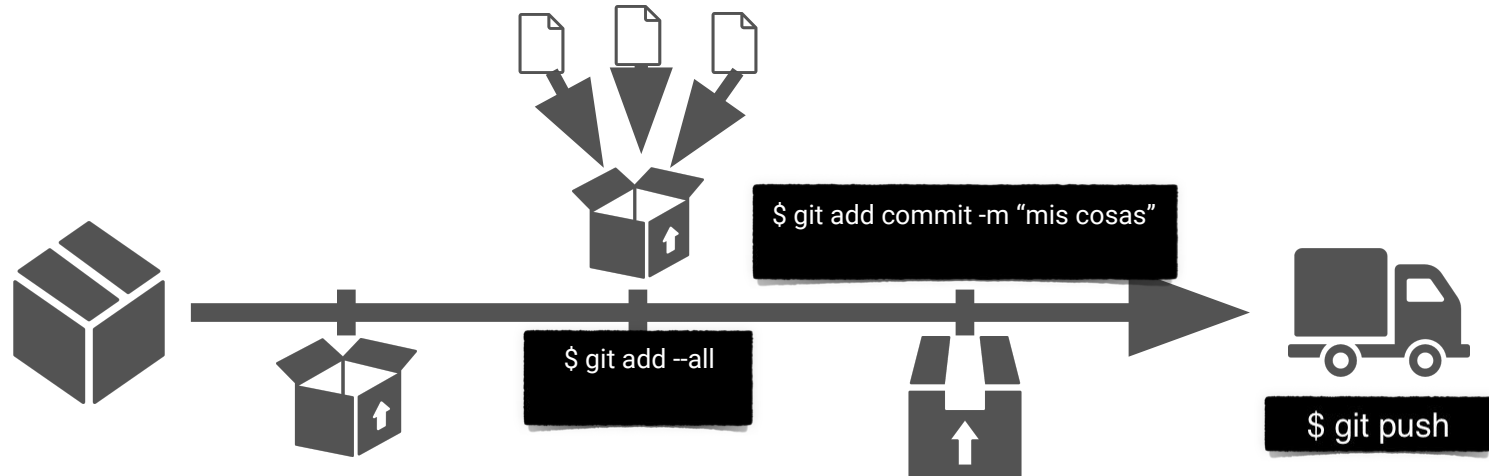
# Configurar Git

```
$ git config --global user.name "Tu nombre"
```

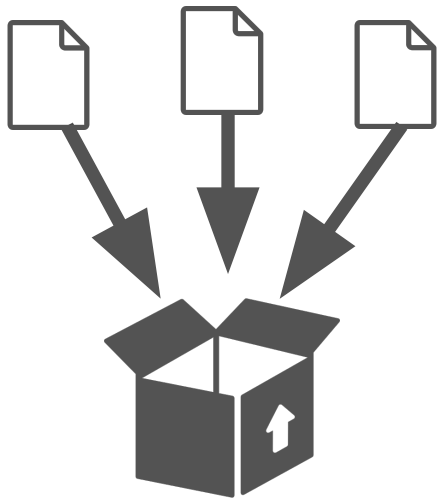
```
$ git config - -list  
user.name= tu nombre  
user.mail= tucorreo@mail.com
```

- Escribir en el terminal el comando **git config --global user.name**, seguido de **tu nombre** dentro de comillas dobles.
- Revisar si la configuración está correcta usando el comando **git config - -list**.

# Flujo de trabajo de Git



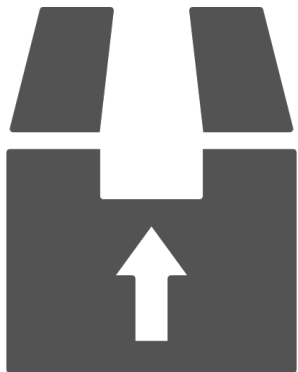
## Agregar cosas a la caja



- Ahora que está abierta, es momento de agregar todas nuestras cosas a la caja.
- Esta acción es igual a escribir **git add --all** o uno a uno con **git add [nombre\_de\_archivo]**.

```
$ git add --all
```

## Cerrar y etiquetar la caja



```
$ git commit -m "mis cosas"
```

- Luego de agregar todas nuestras cosas, habrá que etiquetar y enlistar las cosas de nuestra caja.
- Esta acción es igual a escribir **git commit -m "mis cosas"**.



# Enviar la caja



- A continuación debemos enviar la caja a destino.
- Esta acción es igual a escribir **git push origin master**.

```
$ git push origin master
```

# Revisar la caja



```
$ git status
```

```
On branch master  
nothing to commit, working tree clean/  
mi-usuario/Desktop/mi-carpeta)
```

- **git status** nos ayudará a saber en qué parte del flujo de trabajo nos encontramos.
- Es similar a revisar el proceso dentro de una lista.

**/\* Desafío \*/**

# Panel de discusión

**{desafío}**  
**latam\_**

*Academia de  
talentos digitales*

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