



Terminal, git, GitHub y GitHub pages

Parte I





Inicio

{desafío}
latam_



10 minutos

`/*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.*/`

Objetivos de la sesión



Desarrollo

{desafío}
latam_



150 minutos

/* Terminal y manejo de archivos */

Introducción al terminal

El terminal, es una poderosa herramienta basada en una interfaz de texto que sirve para comunicarse directamente con un computador. Utiliza líneas de comandos para navegar por archivos y directorios, al mismo tiempo, se utiliza para interactuar con programas que no tienen interfaz gráfica.

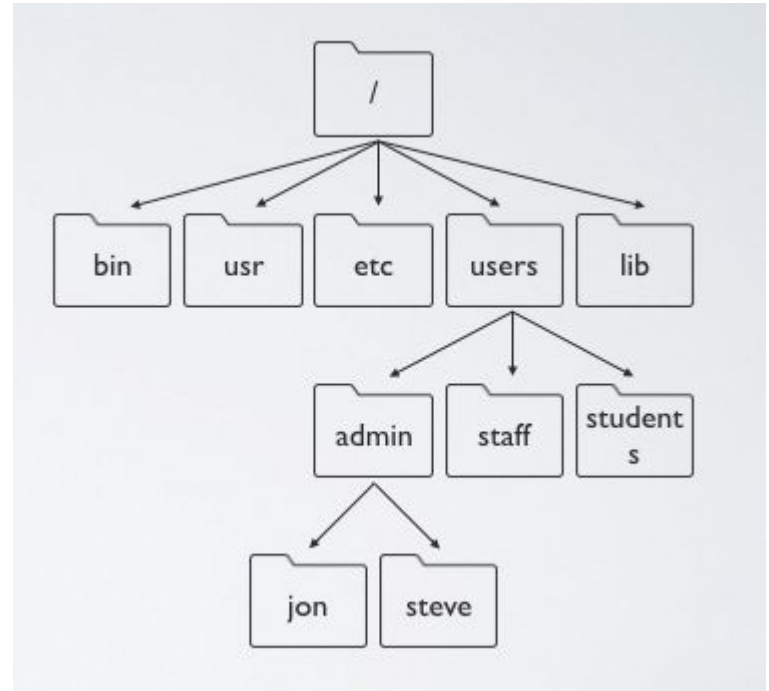
Inicialización de terminal

- 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 lo siguiente pwd y presiona enter.

```
[MacBook-Pro-de-Cristian:~ adacher$ pwd  
/Users/adacher  
MacBook-Pro-de-Cristian:~ adacher$
```

Utilizando el terminal

Listar archivos (ls)

El comando ls o sea, list, que como su nombre lo indica, hará una lista de los archivos y directorios contenidos en el directorio en el que se está ejecutando el comando.

```
Last login: Thu Apr 19 13:10:41 on ttys000
MacBook-Pro-de-Cristian:~ adacher$ ls
Adlm                Music               geth.log
AndroidStudioProjects  Pictures           logfile
Applications        ProjectsKeys       mapsexample01.jks
Desktop             Public             mapsexperiment.jks
Documents           StudioProjects    node_modules
Downloads           Test.jks           package-lock.json
Dropbox             deploy@167.99.158.104 prueba.jks
Library            final.md           prueba1.jks
LoginExampleSocial.jks flashg4key.jks     prueba3.jks
Movies             foo
MacBook-Pro-de-Cristian:~ adacher$
```

Utilizando el terminal

Comandos de navegación entre directorios (cd)

El comando llamado `cd` lo que significa `change directory`. Si queremos recorrer e introducirnos en algún otro directorio con `cd` debemos escribir `cd` + la ruta al archivo que queremos llegar. Por ejemplo, accederemos a la carpeta `Desktop` o `Escritorio`, dependiendo de tu sistema operativo, escribiendo:

```
cd Desktop
```

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.

Manejo de archivos y carpetas

Creación de archivos

Para crear un archivo nuevo, utilizaremos el comando `touch`. La forma de utilizarlo es `touch nombre_del_archivo.extension`, esto creará un nuevo archivo con el nombre que hayamos ingresado y la extensión que apunta al tipo de archivo.

```
touch archivo.extension
```

Manejo de archivos y carpetas

Creación de directorios

Para crear directorios usamos el comando mkdir, que significa make directory

```
mkdir directorio
```

Manejo de archivos y carpetas

Copia de archivos

Para copiar archivos, se utiliza el comando cp, el nombre del comando, luego se añade el archivo que vamos a copiar

```
cp archivo.extension ruta_destino/archivo_nuevo.extension
```


Manejo de archivos y carpetas

Copia de directorios

Para copiar directorios, también se utiliza el comando cp, agregándole la opción -r

```
cp -r directorio_copiado directorio_destino
```

Manejo de archivos y carpetas

Mover archivos y directorios

Para mover archivos, se utiliza el comando mv que significa move

```
mv archivo.extension directorio_destino/
```

Manejo de archivos y carpetas

Borrar archivos

con el comando `rm` podremos eliminar un archivo y con una opción en su sintaxis, también directorios completos.

```
rm archivo.extension
```

¡Importante!

¡Ten mucho cuidado! Ya que los archivos eliminados de esta forma no van a parar en la papelera de reciclaje de tu computador.

Como vimos en las páginas anteriores, existen múltiples comandos que nos ayudarán a navegar a través de los archivos y directorios, facilitando nuestro trabajo. Para familiarizarnos con ellos, haremos un recorrido por los comandos que acabamos de revisar

Ejercicio Guiado: Manejando archivos y carpetas



Quiz



{desafío}
latam_

/* Introducción a git */

Introducción a Git

Git es un sistema de control de versiones gratuito, muy útil y ampliamente utilizado en el desarrollo.

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

Introducción a Git

Control de versiones

Utilizando git tendríamos acceso a todas las versiones guardadas, permitiendo incluso volver a una de ellas.

Introducción a Git

¿Cuándo debemos usar git?

La recomendación es usarlo siempre que trabajemos desarrollando código (o con documento en texto plano), 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. Algunos editores de texto traen incorporado formas automatizadas para usarlo, por ejemplo en Atom.
- También existen programas con interfaces gráficas como gitkraken o git Tower.
- Nosotros lo utilizaremos en nuestra terminal. Esto puede parecer a primera vista un poco más difícil, pero nos ayudará a entender mejor los conceptos más importantes.

Instalando Git

Instalando Git

Verificando si se encuentra instalado

El primer paso es verificar si ya tenemos instalado git en nuestro sistema. Esto lo podemos realizar escribiendo el comando

```
git --version
```

Instalando Git

En Linux

En Linux, ebemos utilizar el siguiente comando en la terminal.

```
sudo apt-get install git
```

Instalando Git

En Windows

En Windows, si seguiste las instrucciones proporcionadas en la lectura para instalar el terminal, no deberías tener problemas. Pero, si por algún motivo no lo tienes, sigue los siguientes pasos:

1. Entra al sitio [git/win](https://git-scm.com/win).
2. Descarga el archivo dependiendo de tu versión del sistema operativo.
3. Ejecuta el archivo descargado y sigue los pasos del instalador.

Configurando git

Configurando Git

Configurar es nuestro usuario en git.

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

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


Configurando Git

Una vez ingresados los comandos no veremos ninguna confirmación de la acción entonces puedes usar el comando.

```
git config --list
```

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

Si ves este mensaje es porque lo lograste.

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

Uso básico de git

Uso básico de git

Inicializando git

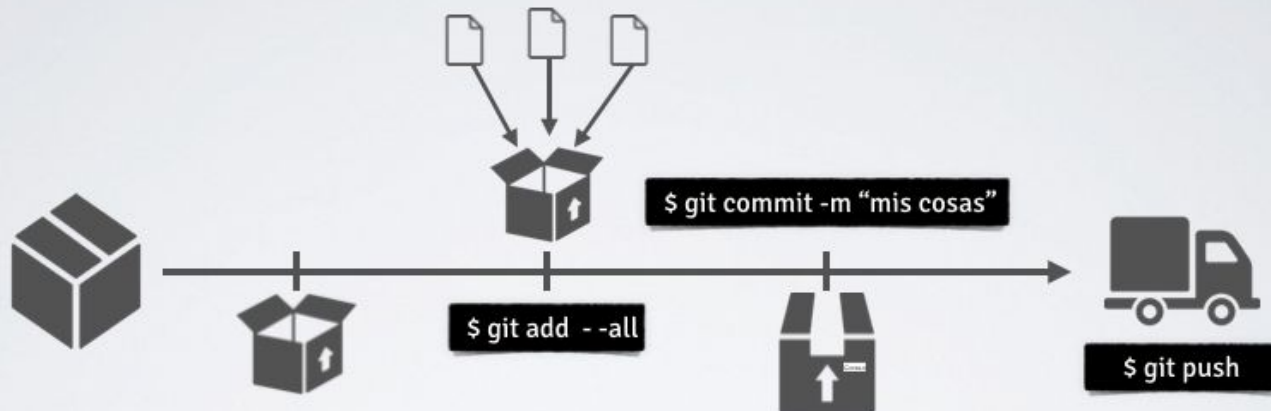
Nuestro primer paso será escribir en la carpeta de proyecto lo siguiente

```
git init
```

Uso básico de git

Usando git

FLUJO DE TRABAJO DE GIT



Uso básico de git

git add

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

```
git add --all
```

o

```
git add .
```

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

git push

El último paso del flujo consiste en enviar la caja a destino. Esto se hace vía comando:

```
git push
```

Uso básico de git

¿Local o remoto?

Agregar git add y confirmar git commit sucede completamente dentro de nuestro computador, en el envío git push se usan lugares de destino. Esto lo aprenderemos en el capítulo de GitHub

Uso básico de git

Subiendo una nueva versión

podemos revisar todas las versiones de un proyecto con:

```
git log
```

Uso básico de git

Gestionando los cambios

Si queremos ver los cambios introducidos en la consola, podemos usar el siguiente comando

```
git diff
```

Uso básico de git

Gestionando los cambios

Para ver un resumen de qué archivos se han modificado podemos utilizar el siguiente comando:

```
git status
```

Vamos a utilizar git dentro de un proyecto. En nuestro caso, vamos a iniciar git en el sitio creado con HTML y CSS de un proyecto anterior llamado "meet&coffee".

Ejercicio guiado: Utilizando git en un proyecto (Parte I)



Quiz



{desafío}
latam_

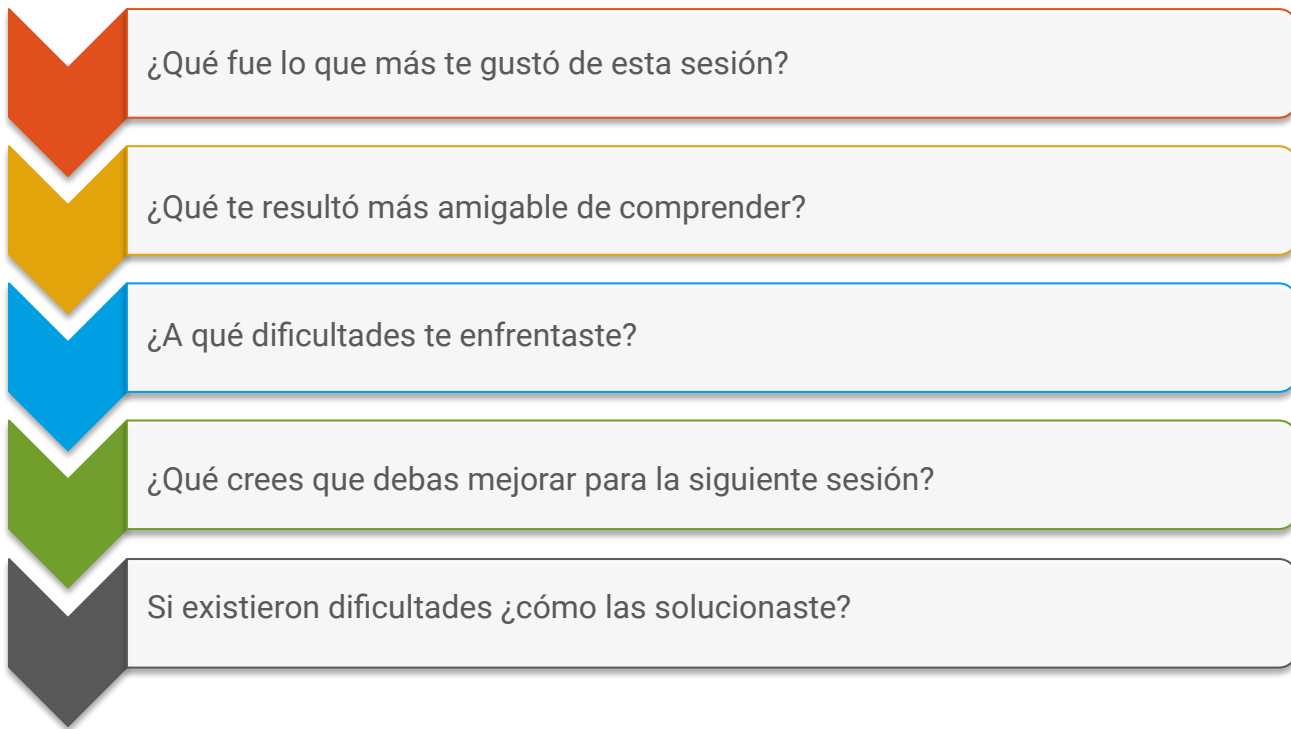


Cierre

{desafío}
latam_



20 minutos



¿Qué fue lo que más te gustó de esta sesión?

¿Qué te resultó más amigable de comprender?

¿A qué dificultades te enfrentaste?

¿Qué crees que debas mejorar para la siguiente sesión?

Si existieron dificultades ¿cómo las solucionaste?



*Academia de
talentos digitales*

www.desafiolatam.com



/DesafioLatam



/DesafioLatam



/DesafioLatam



/DesafioLatam