

**{desafío}**  
**latam\_**

# Terminal, GIT, GitHub y GitHub pages \_

Parte I



# Terminal y manejo de archivos

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

# Introducción al terminal

The image shows a terminal window in the foreground and a GitHub repository page in the background. The terminal window is titled 'adacher@adacher-ThinkPad-T440p: ~/Documents/webfundamentals' and displays the output of a 'git status' command. It shows that the current branch is up to date with 'origin/master'. There are changes not staged for commit, including modified files like '.vscode/settings.json', 'expl/actividades/S1\_ACT1\_WF/readme.md', 'expl/actividades/S1\_ACT2\_WF/readme.md', 'expl/guiones/Unidad 01.md', 'expl/guiones/Unidad 02.md', 'expl/guiones/Unidad 04.md', 'expl/guiones/Unidad 05.md', and 'expl/dialogos/unidad 03/Unidad 03.md'. There are also untracked files like 'expl/guiones/1assets/unit01project.gif', 'expl/guiones/1assets/unit02project.gif', and 'expl/guiones/1assets/unit03project.gif'. The terminal prompt is 'adacher@adacher-ThinkPad-T440p:~/Documents/webfundamentals\$'. The background shows a GitHub repository page with a file explorer, a commit history table, and a pull request section.

```
adacher@adacher-ThinkPad-T440p: ~/Documents/webfundamentals
File Edit View Search Terminal Help
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   .vscode/settings.json
        modified:   expl/actividades/S1_ACT1_WF/readme.md
        modified:   expl/actividades/S1_ACT2_WF/readme.md
        modified:   expl/guiones/Unidad 01.md
        modified:   expl/guiones/Unidad 02.md
        modified:   expl/guiones/Unidad 04.md
        modified:   expl/guiones/Unidad 05.md
        modified:   expl/dialogos/unidad 03/Unidad 03.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        expl/guiones/1assets/unit01project.gif
        expl/guiones/1assets/unit02project.gif
        expl/guiones/1assets/unit03project.gif

no changes added to commit (use "git add" and/or "git commit -a")
adacher@adacher-ThinkPad-T440p:~/Documents/webfundamentals$
```

File	Commit	Time
js	first commit	3 months ago
vendor	first commit	3 months ago

Fuente: Desafío Latam.

# Inicialización de terminal

Para inicializar el terminal utilizaremos un atajo:

Linux	Mac	Windows
Presiona <code>ctrl + alt + t</code> .	Presiona <code>⌘ + espacio</code> , busca por spotlight terminal.	Busca el programa <code>git bash</code> y ábrelo.

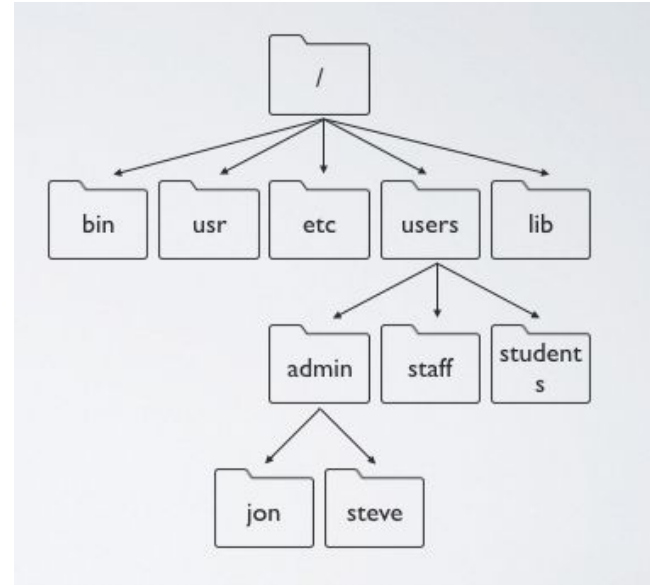
Fuente: Desafío Latam.

# Utilizando el terminal

Estudiaremos comandos básicos que nos ayudarán a movernos y ubicarnos dentro de los directorios de nuestro computador, como también listar los archivos o carpetas al interior de un directorio.

## Explicación de las estructuras de directorio

Los directorios o carpetas de nuestro computador tienen una estructura del tipo árbol.



Fuente: Desafío Latam.

# Conocer en qué directorio estamos

Una de las cosas más importantes que necesitamos saber al trabajar con la terminal, es conocer en qué directorio estamos trabajando.

Escribe en tu terminal lo siguiente `pwd` y presiona enter.

A screenshot of a terminal window with a dark background. The prompt is '[MacBook-Pro-de-Cristian:~ adacher\$'. The user has entered 'pwd' and the output is '/Users/adacher'. The prompt is now '[MacBook-Pro-de-Cristian:~ adacher\$' followed by a cursor.

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

Fuente: Desafío Latam.

# Listar archivos

Para usarlo escribe ls en tu terminal y presiona enter.

```
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$
```

Fuente: Desafío Latam.



# Comandos de navegación entre directorios

Aprenderemos a navegar entre directorios, para ello utilizaremos el comando llamado `cd` lo que significa change directory.

Escribe `cd /` y presiona enter, en la terminal.

Si queremos recorrer e introducirnos en algún otro directorio con `cd` debemos escribir `cd` + la ruta al archivo que queremos llegar.

Podemos comprobar que estamos en la carpeta correcta escribiendo `pwd`.

# Anatomía de un comando

Todos los comandos tienen un nombre que los distingue, por ejemplo `ls` y `pwd`, serían el nombre del comando.

Algunos comandos como `cd` además tienen uno o más argumentos.

## Importante

Linux es sensible a las mayúsculas. Esto implica que es distinto escribir `CD` o `cd`.



# Manejo de archivos y carpetas

<b>Creación de archivos</b>	Para crear un archivo nuevo, utilizaremos el comando <code>touch</code> : <code>touch archivo.extension.</code>
<b>Creación de directorios</b>	Para crear un directorio, se utiliza el comando <code>mkdir</code> : <code>mkdir directorio.</code>
<b>Copia de archivos</b>	Para copiar archivos, se utiliza el comando <code>cp</code> : <code>cp archivo.extension ruta_destino/archivo_nuevo.extension.</code>
<b>Copia de directorios</b>	Para copiar directorios, también se utiliza el comando <code>cp</code> : <code>cp -r directorio_copiado directorio_destino.</code>
<b>Mover archivos y directorios</b>	Para mover archivos, <code>mv</code> ; para mover directorios, <code>cp</code> : <code>mv archivo.extension directorio_destino/</code> <code>mv directorio_origen directorio_destino.</code>
<b>Borrar archivos</b>	Para eliminar un archivo utilizaremos el siguiente comando <code>rm</code> : <code>rm archivo.extension.</code>

## Manejando archivos y carpetas

### Contexto

Como vimos en este capítulo, 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.

Sigamos el paso a paso.

**Ejercicio  
guiado**

Tenemos la siguiente estructura de carpetas:

```
alumno |--- apuntes      |--- modulo1.doc
        |--- modulo2.doc
        |--- modulo3.doc
        |--- documentos
        |--- ejercicios |--- modulo1.html
                        |--- modulo2.html
                        |--- modulo3.html
                        |--- modulo4.html
                        |--- modulo5.html
```

- Se pide crear la estructura de carpetas y archivos por medio de consola.
- Se pide crear una carpeta por cada módulo, manteniendo la estructura de subcarpetas.
- Se pide eliminar las carpetas obsoletas.

## Ejercicio Propuesto (1)

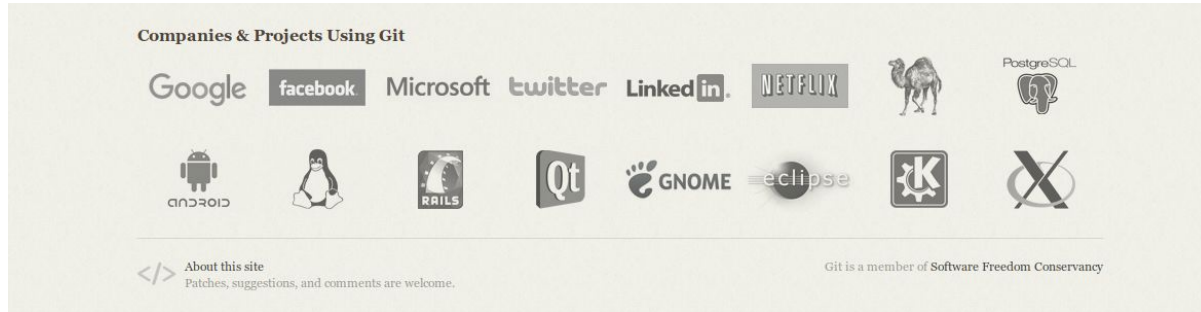
# Introducción a Git

- Emplear las etapas del versionamiento de Git, para mantener un repositorio de versiones.

## Competencias

# Introducción a git

Git es un sistema de control de versiones gratuito, muy útil y ampliamente utilizado en el desarrollo. Fue creado con la idea de ayudar a manejar proyectos no importando su tamaño.



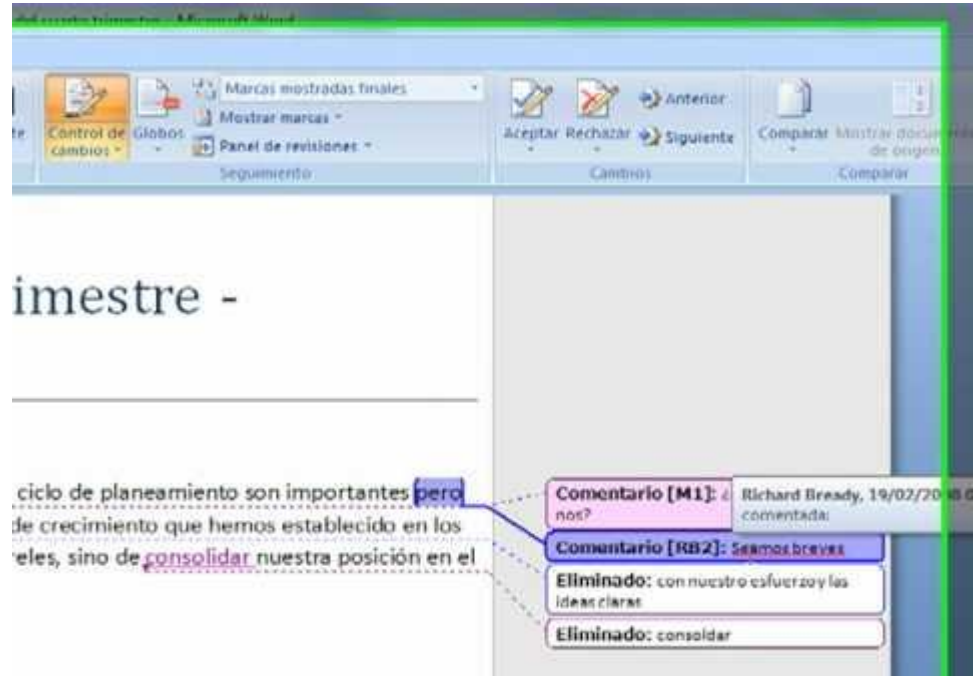
Sitios que utilizan Git.

Fuente: [Openwebinars](#)



# Control de versiones

Imaginemos un editor de documento de texto como Word, en el cual vamos añadiendo cambios y guardándolos. Si cerramos el programa solo tendremos los últimos cambios guardados. Utilizando git tendríamos acceso a todas las versiones guardadas.



Fuente: Desafío Latam.

## ¿Cuándo debemos usar git?

La recomendación es usarlo siempre que trabajemos desarrollando código (o con documento en texto plano).

## Formas de uso de git

Algunos editores de texto traen incorporado formas automatizadas para usarlo, por ejemplo en Atom.



Fuente: [Miro Medium](#).

# Instalando git

Vamos a instalar git en nuestro computador. Ahora veamos qué instalación necesitas dependiendo del sistema operativo que tengas (Mac/Linux, Windows).

## 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` en nuestra terminal.

# Configurando git

Ahora que ya tenemos git instalado, nuestro siguiente paso será configurarlo en nuestro equipo.

Para ello utilizaremos los siguientes comandos:

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

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

# Uso básico de git

Realizaremos una demostración de los pasos comunes para crear un proyecto con git y manejar cambios.

Inicializando git	Usando git	git add	git commit
<p>Siempre que queramos trabajar con git, nuestro primer paso será escribir en la carpeta de proyecto lo siguiente:</p> <pre>git init.</pre>	<p>Metáfora de la mudanza. Realizaremos 3 acciones importantes: añadir, confirmar y enviar.</p>	<p>Agregamos nuestros archivos y cambios utilizando el comando git add seleccionando uno o varios archivos.</p> <pre>git add --all</pre> <p>O</p> <pre>git add.</pre>	<p>Luego, debemos confirmar estos cambios, y agregarle una etiqueta con una descripción. Esto se logra con</p> <pre>git commit -m "Nombre o descripción del commit".</pre>

# Uso básico de git

git push	¿Local o remoto?	Subiendo una nueva versión	Gestionando los cambios
<p>El último paso del flujo consiste en enviar la caja a destino. Esto se hace vía comando:</p> <p><code>git push</code>.</p>	<p>El uso que haremos de git será <code>git init</code> para iniciar git en un proyecto y luego, por cada conjunto de cambios significativos: <code>git add</code>, <code>git commit</code> y <code>git push</code>.</p>	<p>Finalmente podemos revisar todas las versiones de un proyecto con:</p> <p><code>git log</code>.</p>	<p>Si queremos ver los cambios introducidos en la consola, podemos usar un comando llamado <code>git diff</code> para ver qué ha cambiado.</p>

## Utilizando git en un proyecto

Vamos a utilizar git dentro de un proyecto.

En nuestro caso vamos a iniciar git en el sitio creado con HTML y CSS de una experiencia anterior llamado "meet&coffee".

## Ejercicio guiado (Parte I)

Toma un proyecto anterior o incluso la estructura de carpetas que hemos creado en el ejercicio 1 y realiza los pasos para iniciar git y versionar los cambios de tu proyecto.

1. Inicializa git en la carpeta contenedora.
2. Agrega los archivos al stage.
3. Genera el primer commit.
4. Muestra el status.

## Ejercicio Propuesto (2)



**{desafío}**  
**latam\_**

*Academia de  
talentos digitales*

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