

INTRODUCCIÓN A DESARROLLO WEB

¿QUÉ ES EL DESARROLLO WEB?

“La planificación y el diseño de páginas de internet, con la interacción de medios como textos, imágenes, vídeos, sonido y enlaces a otras páginas web”.

- Marie Quilly (2014)

PRINCIPIOS BÁSICOS

- 👁️ **Navegabilidad:** ¿Dónde puede ir el usuario y de qué forma? ¿Cómo pasar de una página a otra?
- 🔍 **Interactividad:** ¿Cómo pasar de un medio a otro, o de una aplicación a otra? ¿Cómo relacionar los diferentes medios?
- 💻 **Arquitectura de la información:** ¿Cómo organizar esta última?

CONCEPTOS BÁSICOS SOBRE INTERNET



INTERNET

Se trata de una red de equipos de cálculo, que se relacionan entre sí a través del uso de lenguajes universales.



NAVEGADOR

Es el instrumento que permite a los usuarios de internet navegar entre las distintas páginas de sitios webs.



BUSCADOR

Es un sistema informático que busca todo tipo de información en la web, para arrojar la información solicitada al usuario.

¿QUÉ ES UN SITIO WEB?

Es un **espacio virtual** en Internet.
Se trata de un conjunto de páginas web, accesibles desde un mismo dominio o subdominio de la World Wide Web (WWW).

**SITIO WEB Y PÁGINA WEB
NO SON LO MISMO**



DIFERENCIA ENTRE PÁGINA WEB Y SITIO WEB

Páginas web



Sitio web



Un sitio web, está constituido por varias páginas web organizadas jerárquicamente.

EL MODELO CLIENTE-SERVIDOR

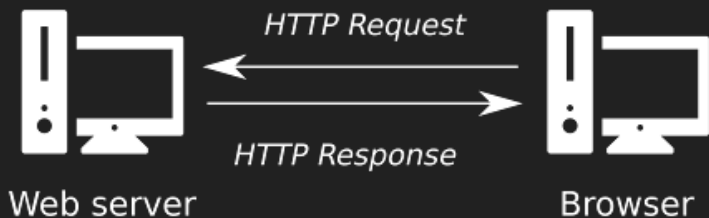
Al abrir un browser o navegador, e ingresar una página web, se lo suele hacer por su nombre, por ejemplo:

www.codingdojo.la

A lo largo de toda la red de internet, hay una serie de máquinas que hacen de “agenda” y nos dan la dirección IP. La dirección IP es un conjunto de números que identifica, de manera lógica y jerárquica, a una interfaz en red.

www.codingdojo.la = 159.89.87.61

EL MODELO CLIENTE-SERVIDOR



Cuando la petición llega al servidor, el mismo resuelve:

- Si el sitio efectivamente está en ese servidor.
- Qué directorio (o carpeta) corresponde con ese sitio web.
- Qué archivo está siendo solicitado (si no es ninguno, siempre se busca uno por defecto).
- Qué tecnologías conforman esos archivos.

La respuesta a estas peticiones será un conjunto de código **HTML**, **CSS** y **JS**.

HERRAMIENTAS BÁSICAS PARA EL DISEÑO Y DESARROLLO WEB

NAVEGADORES



Para probar tu web, necesitarás al menos un navegador instalado en tu computadora. Los más comunes son:

- Google Chrome: https://www.google.com/intl/es_es/chrome/
- Microsoft Edge: <https://www.microsoft.com/es-es/edge>
- Firefox: <https://www.mozilla.org/es-ES/firefox/new/>
- Safari: <https://www.apple.com/es/safari/>
- Opera: <https://www.opera.com/es>

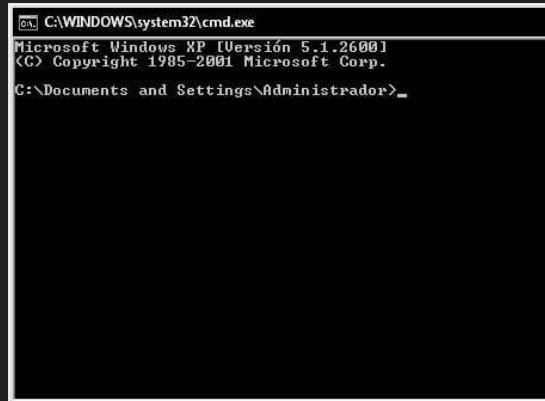
EDITORES DE CÓDIGO



Son programas que te permiten realizar o escribir código fuente (HTML, CSS, PHP, JavaScript) de tus proyectos. Al ser dinámicos, son idóneos para cuando desarrollas uno con varios lenguajes de programación. Algunos de ellos son:

- Sublime Text:
<https://www.sublimetext.com>
- Brackets: <https://brackets.io>
- Visual Studio Code:
<https://code.visualstudio.com>

TERMINAL O CONSOLA



En informática, una terminal o consola es la aplicación que se utiliza para interactuar con el computador a través de comandos. Todos los sistemas operativos la traen.

GIT / GITHUB



Git es un software de control de versiones, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

LENGUAJES PARA EL DESARROLLO WEB



Para el diseño y desarrollo web existen diferentes lenguajes, que nos permiten llevar un diseño o boceto, a una estructura que pueda interpretar un computador. Para esto, se utilizan dos lenguajes bases: HTML y CSS.

HTML

Es un lenguaje de marcado de etiquetas, que permite crear documentos para web.

Estaremos viendo HTML, incluyendo toda su estructura y etiquetas, para aprender a dar formato a la información que queremos mostrar.

CSS

El CSS, en español «hojas de estilo en cascada», es un lenguaje de 'diseño gráfico', utilizado para definir y crear la presentación de un documento estructurado, escrito en un lenguaje de marcado.

Es decir, modificaremos visualmente la estructura ya creada en HTML para adecuarla a las necesidades del cliente.


```
index.html x
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Diseño Web</title>
</head>
<body>
  <h1>Bienvenidos a Diseño Web</h1>
  <h3>Que la fuerza te acompañe!</h3>
  <ul>
    <li>Inicio</li>
    <li>Blog</li>
    <li>Contactos</li>
  </ul>
  <h3>Sobre el profesor</h3>
  <p>El profesor explica, hace y ustedes practican</p>
  <h3>Sobre ustedes</h3>
  <p>Aprenderan a diseñar una página web</p>
</body>
</html>
```

Bienvenidos a Diseño Web

Que la fuerza te acompañe!

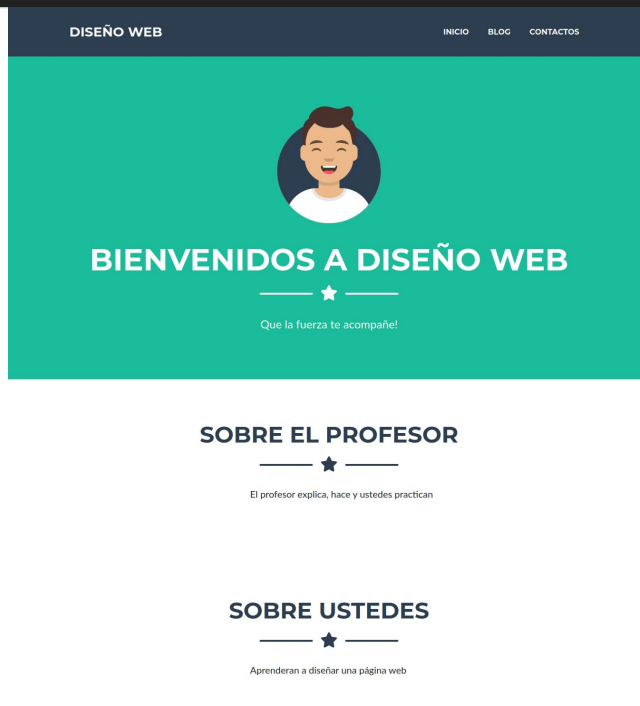
- Inicio
- Blog
- Contactos

Sobre el profesor

El profesor explica, hace y ustedes practican

Sobre ustedes

Aprenderan a diseñar una página web



COMENZANDO CON HTML

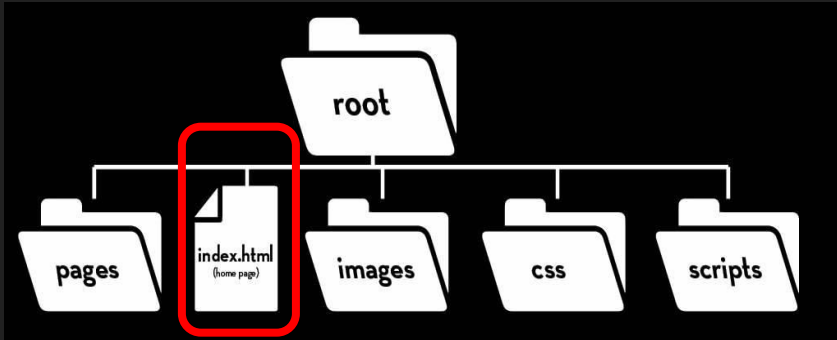
CONVENIO DE ARCHIVOS

- No deben tener espacios, acentos/tildes, eñes, ni símbolos o si son varias palabras usar guiones “ - “ o “ _ “. Ejemplo: “mi-pagina-web”
- Tienen que estar escritos en minúsculas.
- Deben tener la extensión “.html” (es la forma en que el servidor web sabe que se trata un documento web).



Atención: Al guardar archivos en Windows, verificar que no se guarden con la doble extensión de archivos (archivo.html.txt).

DOCUMENTO PREDETERMINADO



👉 Es el archivo que se carga cuando todavía no has hecho en ningún link (por ejemplo, al pedir www.codingdojo.la no dice qué archivo quieres).

Debe llamarse **index.html**, ya que es el nombre estandarizado.

ARQUITECTURA DE UN SITIO

```
<a href="productos.html">Productos</a>
```

Los **enlaces**, también conocidos como **links** o **anchors**, se utilizan para relacionar partes del mismo documento. Por defecto, se visualizan azules y subrayados.

Para crear uno, es necesario utilizar la etiqueta de ancla `<a>` con el atributo `"href"`, que establecerá el destino al que apunta.

TIPOS DE ENLACES



ENLACES RELATIVOS

```
<a href="/contacto.html">Contacto</a>
```

```
<a href="/imagenes/mapa.jpg">Ver mapa</a>
```

Los enlaces relativos son aquellos que apuntan a páginas ubicadas dentro del mismo proyecto. Si la página referenciada se encuentra en el mismo directorio, alcanza con mencionar el nombre de la misma para generar el enlace.

En caso de que el archivo se encuentre en un directorio específico, el mismo deberá ser mencionado.

ENLACES ABSOLUTOS

Los enlaces absolutos son aquellos cuyo destino apunta a un documento que está fuera del sitio, y debe ser especificado utilizando la URL completa:

```
<a href="https://www.codingdojo.la/frontend">Curso de Frontend</a>
```


ETIQUETAS HTML

```
<etiqueta>  
  Contenido  
</etiqueta>
```

Las etiquetas HTML están delimitadas por un inicio y un final de cada elemento. Lo que se encuentra dentro de la etiqueta (el contenido) es lo que estás formateando.



CASI todas las etiquetas vienen de a pares: una etiqueta *abre*, otra *cierra*.

ATRIBUTOS DE LAS ETIQUETAS

```
<etiqueta atributo="valor">  
    Contenido  
</etiqueta>
```


Todas las etiquetas aceptan atributos.


✨ Cualquier característica que pueda ser diferente entre una etiqueta y la otra.

El valor que tendrá va entre comillas y cada una puede tener más de un atributo, separados entre sí por espacios. Los mismos sólo van en la de apertura.

ANIDAR ETIQUETAS

Es posible meter una etiqueta dentro de la otra, de hecho, esto es más común de lo que parece. Lo más importante es tener presente que: *Siempre se cierran en orden inverso a la apertura.*

```
<elemento>  
  <subelemento>  
      
  </subelemento>  
</elemento>
```

```
<elemento>  
  <subelemento>  
      
  </subelemento>  
</elemento>
```

ETIQUETAS CERRADAS Y ABIERTAS

- Las etiquetas cerradas encierran un contenido, por lo general texto.
- Las etiquetas abiertas sirven, entre otras cosas, para incluir elementos como imágenes, líneas, entre otros.

En el ejemplo tenemos una etiqueta cerrada llamada `<p></p>`, que engloba un texto y una etiqueta abierta `<hr/>` para incluir una línea horizontal. El signo “/” se utiliza para las etiquetas de cierre. En estas últimas se pone a continuación del signo “<”, mientras que en las abiertas delante del signo “>”.

```
<p>Este es un párrafo con texto en su interior - Etiqueta Cerrada </p>
```

```
<hr/> <!-- Esto es una línea horizontal - Etiqueta abierta -->
```

ETIQUETAS ABIERTAS EN HTML5

Ya no es una obligación poner el signo “ / ”. Por ejemplo, `` funcionará correctamente, y lo mismo sucederá con `
`, `<hr>`, los meta tag.

```
<p>Este es un párrafo con texto en su interior - Etiqueta Cerrada </p>
```

```
<hr> <!-- Esto es una línea horizontal - Etiqueta abierta -->
```

ESTRUCTURA BÁSICA DEL DOCUMENTO

<!DOCTYPE html>: Cuando escribes tu documento HTML, lo primero que debes hacer es escribir el *DOCTYPE*, el cual declara el tipo de documento. Es decir, sirve para indicar que tu documento está escrito siguiendo la estructura determinada por un *DTD* concreto. Un *DTD* es la definición del tipo de documento.

(<https://www.w3.org/QA/Tips/Doctype.html.es>)

<html>: Etiqueta inicial, que define que el documento está bajo el estándar de HTML. Abre y cierra, por lo tanto es fundamental no olvidar la etiqueta **</html>** al finalizar el documento, pues sino no cargará correctamente el contenido de mi sitio.

ESTRUCTURA BÁSICA <HTML>

<head>: Es la parte privada del documento, que se utiliza como un espacio de comunicación entre el sitio web y el navegador. Esta etiqueta envuelve otras importantes como *<title>*, las etiquetas *<meta>* y aquellas relacionadas con la importación de documentos CSS y JS.

<body>: Encierra el contenido propiamente dicho del sitio.



Ambos deben estar dentro de un elemento principal: la etiqueta *<html/>*.

ESTRUCTURA BÁSICA <HEAD>

<title>: La etiqueta title define el título de la página, el cual será visualizado en la solapa del navegador.

<meta>: Se utiliza para añadir información sobre la página (ya sean palabras clave, el autor, la descripción del sitio, etcétera), la cual pueden valerse los buscadores. También puede definir el idioma y la codificación en la cual está escrita la página.


```
1  <!DOCTYPE html> <!-- Tipo de Documento-->
2  <!------->
3  <html lang="en"> <!-- Etiqueta inicial-->
4  <!------->
5      <!-- Cabecera, recursos y
6      datos importantes sobre nuestra web-->
7  <head>
8      <meta charset="UTF-8">
9      <title>...</title>
10 </head>
11 <!------->
12 <body>      <!-- Cuerpo de nuestra web-->
13 </body>
14 <!------->
15 </html>      <!-- Fin de nuestra web-->
16
```

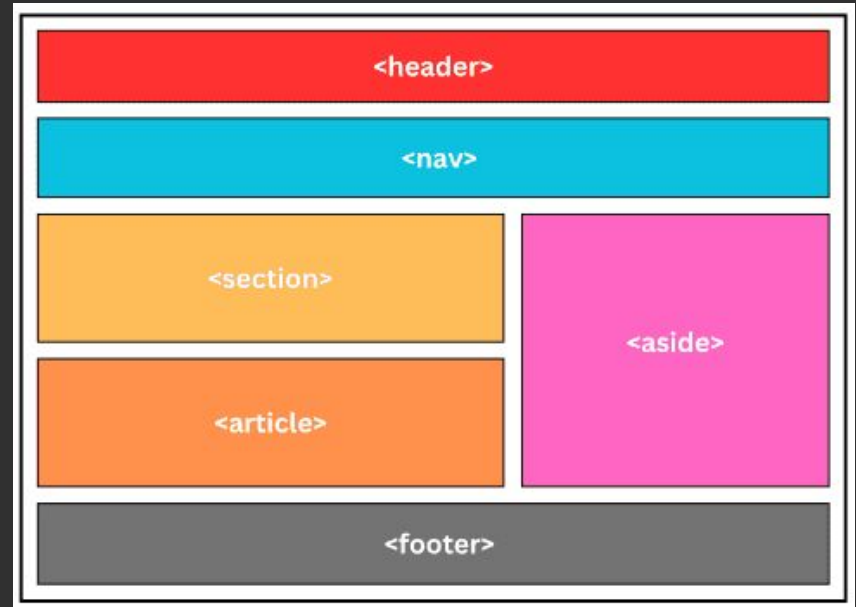
TIPOS DE ETIQUETAS: GRUPO GENERAL

Todas las etiquetas que van dentro del `<body> </body>` se dividen en dos grupos:

- **Elementos de bloque:** Son aquellos que, sin ser modificados por CSS, ocupan el 100% del ancho del contenedor, y se mostrarán uno abajo del otro.
(https://developer.mozilla.org/es/docs/Glossary/Block-level_content)
- **Elementos en línea:** Sólo ocupan el ancho que diga el contenido, y se verán uno al lado del otro.
(https://developer.mozilla.org/es/docs/orphaned/Web/HTML/Inline_elements)

HTML5

HTML5 incorpora etiquetas semánticas que no sólo generan estructura, sino que también definen su contenido.



UTILIZANDO GIT

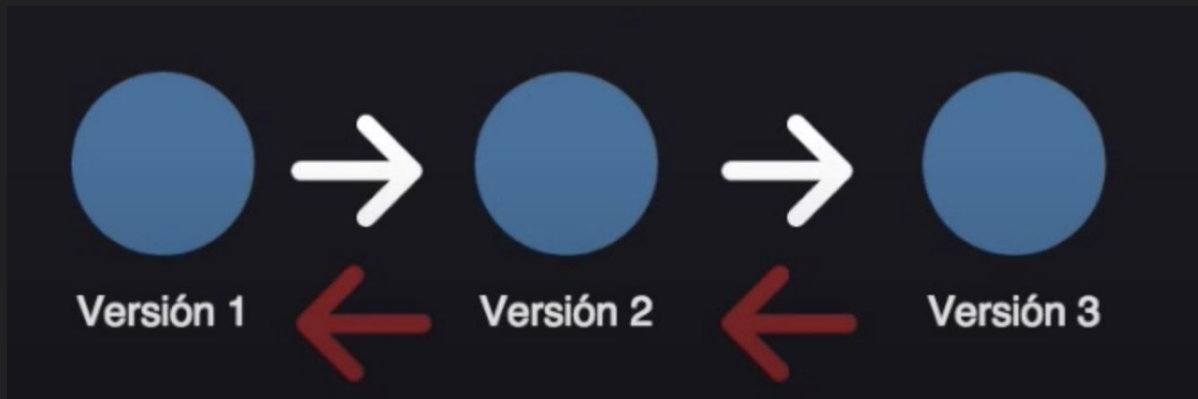
¿QUÉ ES GIT?



Git es un **sistema de control de versiones** gratuito y de código abierto, diseñado para manejar desde pequeños a grandes proyectos de manera rápida y eficaz. Se entiende como control de versiones a todas las herramientas que nos permiten hacer modificaciones en nuestro proyecto. Este sistema registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo.

¿QUÉ ES GIT?

Con GIT, podemos ir a versiones anteriores, muy útil para errores, y para la organización.

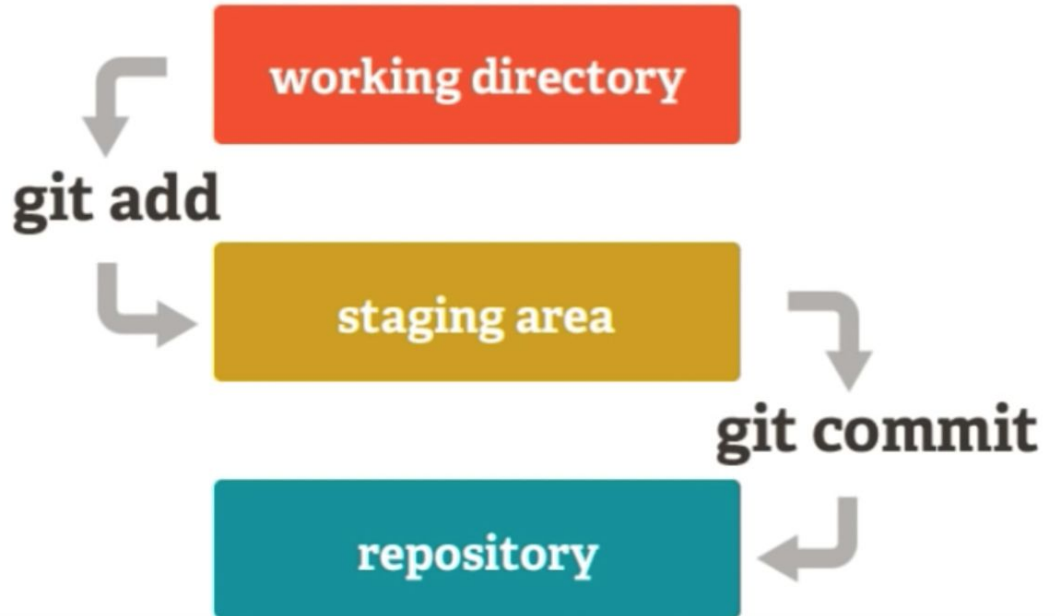


GIT: LOS 3 ESTADOS

1er estado (el que trabajamos)
“preparamos las cajas”.

2do estado (archivos listos)
“agregamos las cajas listas”.

3er estado (registro de todos
los archivos) *“lote listo”.*



¿QUÉ ES GITHUB?



👉 **Git** es uno de los sistemas de control de versiones más populares entre los desarrolladores.

👉 Parte de su popularidad se la debe a **GitHub**, un excelente servicio de alojamiento de repositorios de software con este sistema.

INSTALACIÓN Y CONFIGURACIÓN DE GIT

Lo primero es lo primero: tienes que instalarlo.

Puedes obtenerlo de varias maneras, las dos principales son instalarlo desde código fuente, o instalar un paquete existente para tu plataforma.

👉 Ve a <https://git-scm.com/> y descarga el paquete de instalación “download (nro versión) for Windows” (o si tienes otro sistema operativo, dirá Mac o Linux).

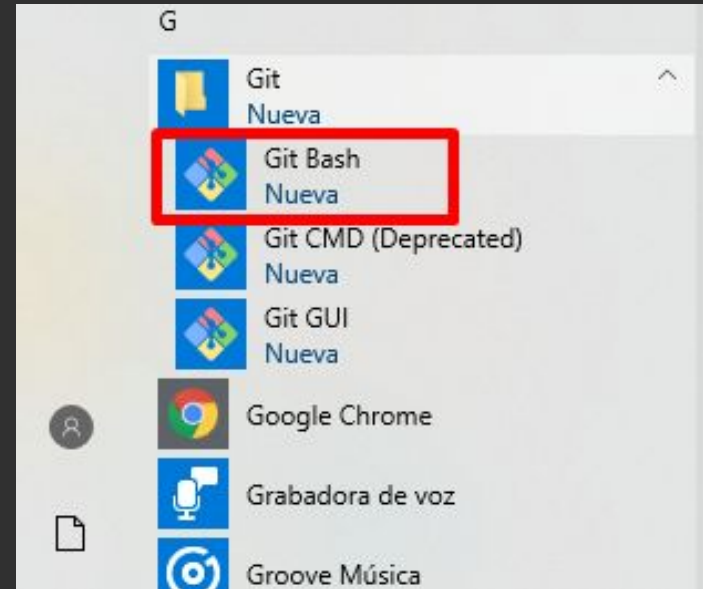
¿CÓMO ABRIR LA TERMINAL?

Para abrir la línea de comandos de Windows o símbolo del sistema, tan sólo tienes que ir a **Inicio > Ejecutar o Buscar > CMD.exe** y se abrirá una pequeña ventana que te recordará al antiguo MS-DOS, o a **Inicio->Git->Git Bash**.

Para abrir la terminal de Mac OS haz clic en el icono **"Finder"** situado en el Dock, luego selecciona **"Aplicaciones > Utilidades"**, y finalmente dale doble clic al icono **"Terminal"**.

¿CÓMO ABRIR LA TERMINAL?

Buscar en su menú el **Git Bash**, para abrir la terminal e iniciar con los comandos.



COMANDOS BÁSICOS

- 👉 **LS:** Sirve para ver el contenido de una carpeta.
- 👉 **CD:** Sirve para entrar en una carpeta o salir de ella (**CD ..**).
- 👉 **TOUCH:** Sirve para crear archivos.
- 👉 **MKDIR:** Con este comando crearás una carpeta nueva. Con **RMDIR** podrás eliminarla.
- 👉 **RM:** Es el comando para eliminar un archivo. Recuerda que no irá a la Papelera, así que piensa muy bien antes de borrar algo. Y para eliminar carpeta usa el comando **RM -RF**.
- 👉 **CLEAR:** Limpia la consola.
- 👉 **EXIT:** Cierra la ventana de la línea de comandos o símbolo del sistema.

VERIFICANDO VERSIÓN DE GIT

Escribe **git --version** y presiona Enter.

```
john@MyShopSolutions: ~$ git --version  
git version 2.17.1  
john@MyShopSolutions: ~$
```

CONFIGURANDO GIT POR PRIMERA VEZ

Tu identidad

Lo primero que deberías hacer cuando instalas Git es establecer tu nombre de usuario y dirección de correo electrónico. Esto es importante porque las confirmaciones de cambios (commits) en Git usan esta información, y es introducida de manera inmutable en los commits que envías.

CONFIGURANDO GIT POR PRIMERA VEZ

1

Elige un nombre de usuario que recuerdes fácil, y el email que usarás en Github.

2

Establece el nombre con el comando: `git config --global user.name "Nombre Apellido"`.

3

Establece el correo a usar con el comando: `git config --global user.email johndoe@example.com`

CONFIGURANDO GIT POR PRIMERA VEZ

/* Paso 1*/

```
john@MyShopSolutions: ~$ git config --global user.name "John Doe"
```

/* Paso 2*/

```
john@MyShopSolutions:~$ git config --global user.email johndoe@example.com
```


COMPROBANDO TU CONFIGURACIÓN

Vamos a comprobar si guardamos bien el usuario usando el comando: **git config --list**

```
john@MyShopSolutions: ~$ git config --list
/* Se puede ver el usuario, el email y otros parámetros
que dependerán de cada sistema operativo */
user.name=John Doe
user.email=johndoe@example.com
color.status=auto
color.branch=auto
color.interactive=auto
color.diff=auto
```

COMPROBANDO TU CONFIGURACIÓN

Puedes también comprobar qué valor tiene la clave nombre en Git ejecutando: **git config user.name**

Puedes consultar de la misma manera **user.email**

```
john@MyShopSolutions: ~$ git config user.name  
John Doe
```

OBTENIENDO AYUDA

Si alguna vez necesitas ayuda usando Git, hay dos formas de ver la página del manual (manpage) para cualquier comando de Git:

```
/* Los dos comandos que disparan la ayuda de Git */  
john@MyShopSolutions: ~$ git help config  
john@MyShopSolutions: ~$ git config --help
```