

DIPLOMADO DE DISEÑO WEB

Valeria Jimeno Villegas



TEMARIO

1. Git y terminal (Estructura tu sitio)
2. HTML5 y CSS3 (Agregando barra de navegación)
3. HTML- Flexbox (Dividiendo secciones en columnas)
4. CSS Grid (Agregando filas y columnas con CSS Grid)
5. Responsive design (Adaptando a la vista desde un móvil)
6. CSS Frameworks (Reutilizando componentes visuales)
7. Preprocesadores de CSS (Optimizando la producción de CSS)
8. Una Web interesante: Transiciones y Animaciones (Animaciones y transiciones)

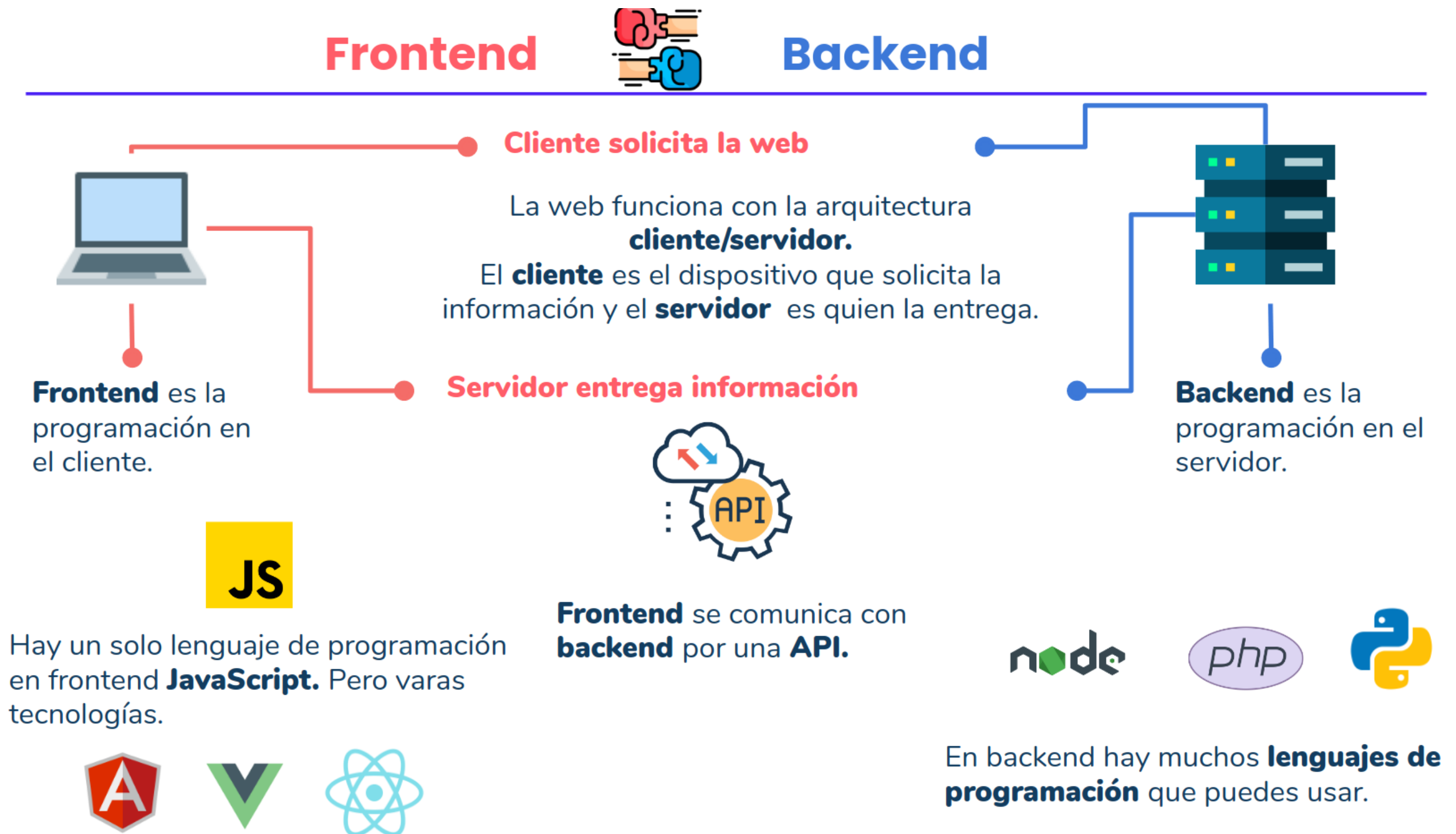


INTRODUCCIÓN

En este curso aprenderás las bases del **desarrollo front-end**, en el cuál podrás adquirir habilidades que te ayudarán a crear un sitio web estático desde cero, usando los lenguajes bases como **HTML y CSS**, además de ciertas herramientas que harán que tu flujo de desarrollo sea más rápido y profesional.



DESARROLLO WEB



FRONTEND

- Se encarga de la composición diseño e interactividad de la página.
- Se toma el diseño y lo convierte en un sitio
- Es lo que el usuario final ve y utiliza
- Se desarrollan programas que se ejecutan en un navegador
- Los lenguajes mas usados para el desarrollo de frontend son:
 - *HTML/CSS*
 - *Javascript*

BACKEND

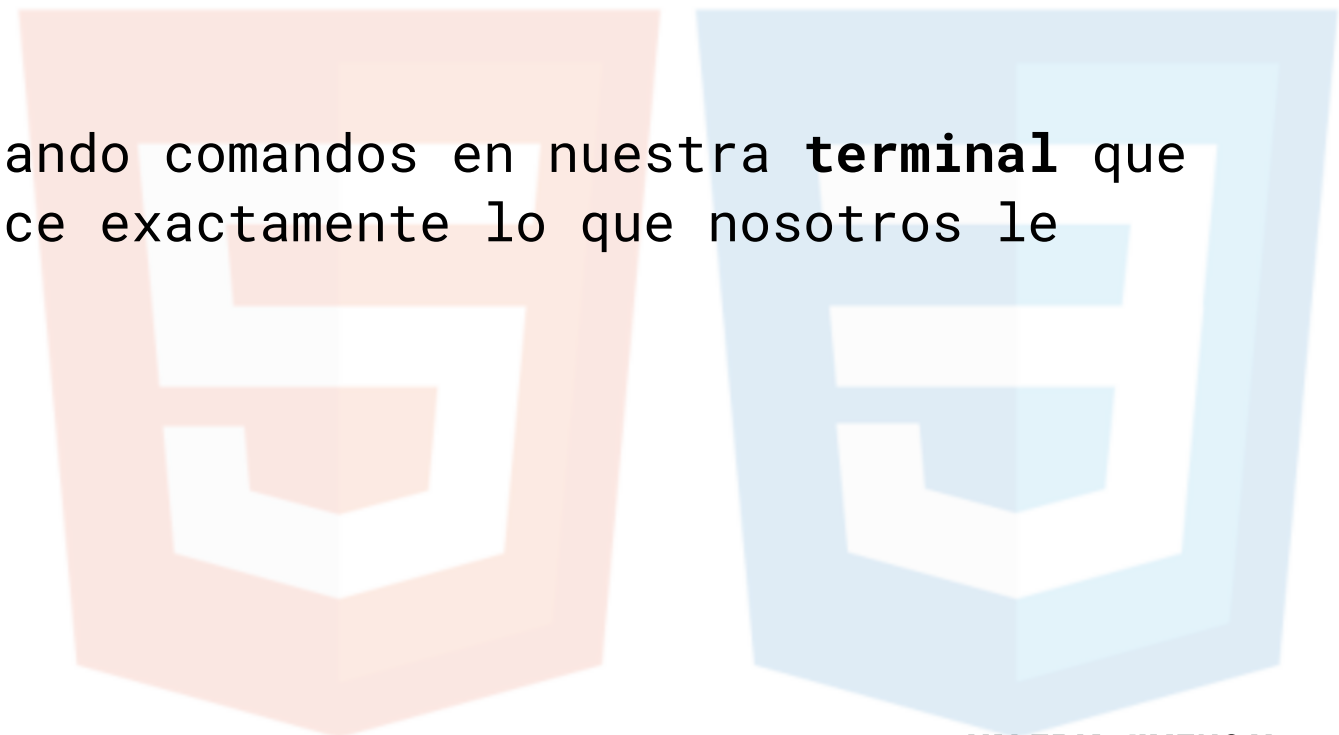
- Es la parte que no se puede ver
- Almacenamiento y manejo de datos
- Conexión con distintos servidores o proveedores
- Consiste de una aplicación que corre en un servidor, una base de datos y la conexión entre estos.
- Los lenguajes mas utilizados para el backend son
 - *Python*
 - *Java*
 - *PHP*
 - *Ruby*

INTRODUCCIÓN

En esta sesión vamos a estar interactuando principalmente con 2 lenguajes: **HTML y CSS**.

Debido a que son 2 lenguajes con **propósitos distintos**, es importante **mantener un orden** en los recursos que usaremos para construir nuestro sitio web, por lo cual es importante que usemos una estructura ordenada para tener un mejor control y entendimiento sobre cómo llevar a cabo nuestro proyecto.

Dicha estructura la construiremos usando comandos en nuestra **terminal** que indicarán a la computadora que realice exactamente lo que nosotros le pidamos.



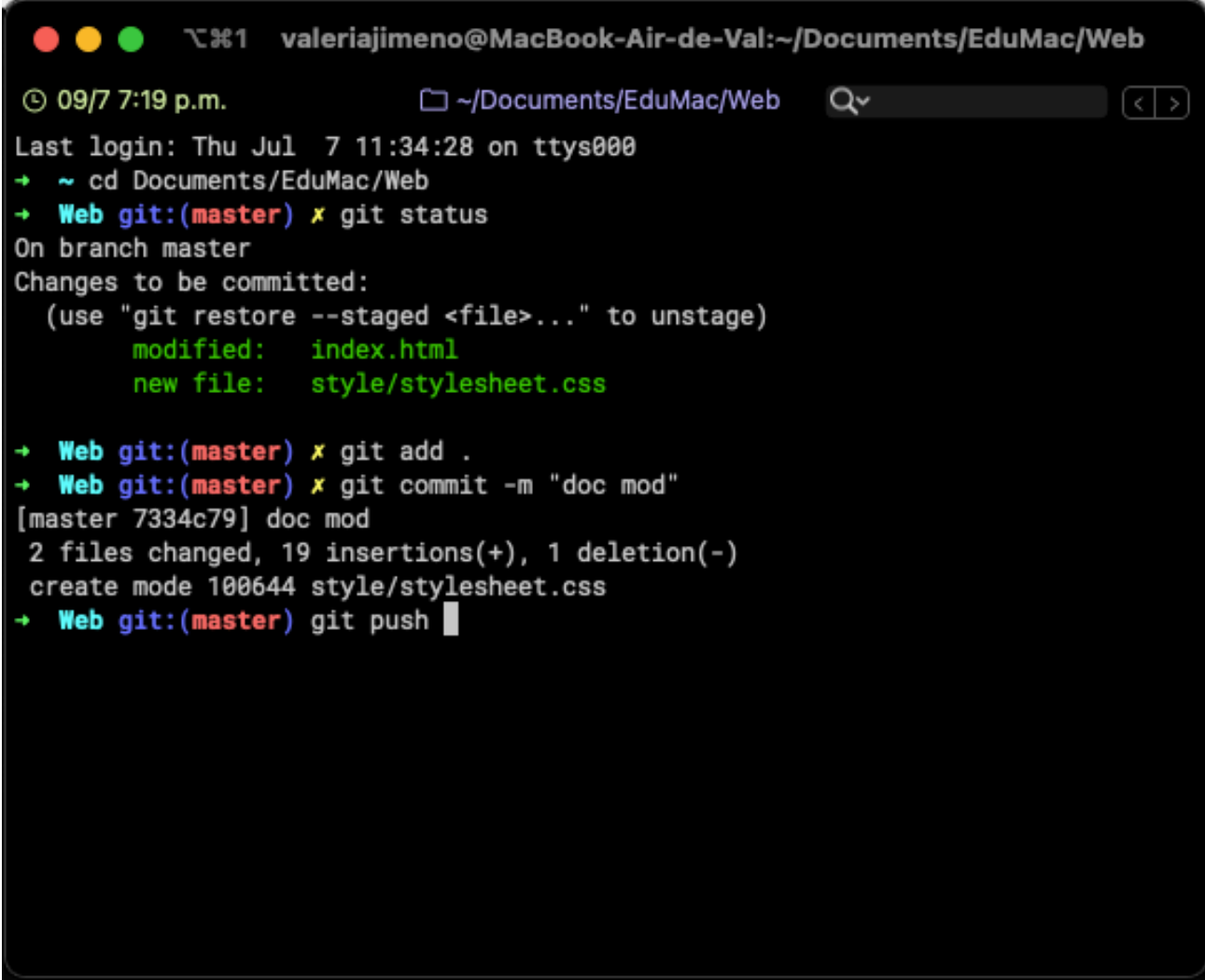
Sublime Text

**A SOPHISTICATED TEXT EDITOR FOR
CODE, MARKUP AND PROSE**



EMULADOR DE TERMINAL

- Un emulador de terminal es una aplicación que permite **virtualizar un intérprete de comandos** pero dentro de la propia interfaz gráfica.
- Tienen la ventaja de posibilitar la **interacción a través de la línea de comandos**, pero sin necesidad de salir del entorno gráfico.



```
valeriajimeno@MacBook-Air-de-Val:~/Documents/EduMac/Web
09/7 7:19 p.m. ~/Documents/EduMac/Web
Last login: Thu Jul 7 11:34:28 on ttys000
→ ~ cd Documents/EduMac/Web
→ Web git:(master) ✕ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html
        new file:   style/styleSheet.css

→ Web git:(master) ✕ git add .
→ Web git:(master) ✕ git commit -m "doc mod"
[master 7334c79] doc mod
2 files changed, 19 insertions(+), 1 deletion(-)
create mode 100644 style/styleSheet.css
→ Web git:(master) git push
```

ANATOMÍA DE UN COMANDO

Todos los comandos se componen de:

- Un **nombre** con el que se invoca el comando.
- Opciones que **modifican el comportamiento** del comando. (Son opcionales).
- Argumentos sobre los que **actúa** el comando. (También opcionales).

comando



ls -al



opciones

argumentos



Documents/

ANATOMÍA DE UN COMANDO



```
$ whoami
```

```
# This command will return your login name
```

```
$ pwd
```

```
# This command will return the path to the current folder you are working on
```

```
$ ls
```

```
# This command will list all the files and folders in the current directory
```

AYUDA Y DOCUMENTACIÓN: MAN

TERMINAL

Man es uno de los comandos más útiles que podrás encontrar, es por eso que lo ponemos en primer lugar. **Nos muestra la definición de un comando** y de los distintos atributos que se pueden usar. Para probarlo tan solo deberemos escribir **man [comando]**

```
#Syntax
man [ comando ]

man ls

# Salida

LS(1)                                User Commands                                LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
    fied.

    Mandatory arguments to long options are mandatory for short options
    too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author

Manual page ls(1) line 1 (press h for help or q to quit)
```

LISTAR ARCHIVOS Y CARPETAS: LS

El siguiente comando que debes conocer es **ls**. Sirve para **listar los archivos y carpetas** que hay dentro del directorio en el que estés.

Agregue **-l** para obtener una lista detallada.

Agregue **-t** para ordenar los resultados por tiempo de archivo.

Agregue **-S** para ordenar por tamaño de archivo.


Agregue **-r** para invertir la clasificación.

Agregue **-a** para incluir archivos ocultos.

DIRECTORIO

DIRECTORIO ACTUAL: PWD

CAMBIAR DE DIRECTORIO: CD



```
# Sintaxis  
pwd  
  
# Sintaxis  
cd [ ruta ]
```

CREAR UN NUEVO DIRECTORIO: MKDIR

El comando `mkdir` sirve para crear un nuevo directorio.

💡 Nota:

Hay que tener en cuenta que lo crea por defecto en el directorio en el que te encuentres (te lo indica siempre en la terminal).

Si quieres crearlo en otro directorio **deberías de incluir la ruta**.

```
# Sintaxis
```

```
mkdir [ carpeta ] [ directorio ]
```

```
valeriajimeno ~ mkdir  
Documentos/nuevo_directorio
```

```
# Salida
```

```
nuevo_directorio maqueta_web web
```

CREAR UN NUEVO DOCUMENTO: TOUCH

El comando **touch** sirve para crear un nuevo archivo vacío si este no existe.

```
# Sintaxis
```

```
touch [ archivoNuevo ]
```

```
valeriajimeno ~/Documentos touch  
index.html
```

```
valeriajimeno ~/Documentos ls
```

```
# Salida
```

```
nuevo_directorio maqueta_web web  
index.html
```


REMOVED UN DIRECTORIO: RMDIR

El comando **rmdir** sirve para eliminar un directorio vacío.

💡 Nota:

Puede eliminar cualquier carpeta siempre que la carpeta esté vacía. Si queremos borrar un directorio que contenga más archivos, podremos hacer uso del atributo **-rm -r [directorio]**



Sintaxis

```
rmdir [ carpeta ] [ directorio ]
```

```
valeriajimeno ~ rmdir  
Documentos/nuevo_directorio
```

Salida

```
maqueta_web web
```

EJERCICIO

Crea una nueva carpeta, ingrese la nueva carpeta usando cd y luego cree varios archivos txt usando touch.

POSIBLES RESPUESTAS



```
$ mkdir my-files  
$ cd my-files  
$ touch file-1.txt  
$ touch file-2.txt  
$ touch file-3.txt
```



```
$ mkdir my-files  
$ cd my-files  
$ touch file-1.txt file-2.txt file-  
3.txt
```

ARCHIVOS

MOSTRAR UN ARCHIVO

Puede mostrar el contenido del archivo en la terminal con el comando **cat**.

COPIAR ARCHIVO

Con el comando **cp [archivo a copiar] [nuevo nombre de archivo copiado]**

MOVER UN ARCHIVO

Con el comando **mv [archivo] [path]**

GIT & GITHUB

Git es un sistema de control de versiones, esto quiere decir que te permite mantener un historial de cambios y que puedas ver los cambios realizados en cualquier momento.

GitHub es un sistema de gestión de proyectos y control de versiones de código, así como una plataforma de red social diseñada para desarrolladores.

💡 **Nota:**

Git es el programa que se ejecuta en su computadora y Github es donde su trabajo se almacena en línea.

HTML

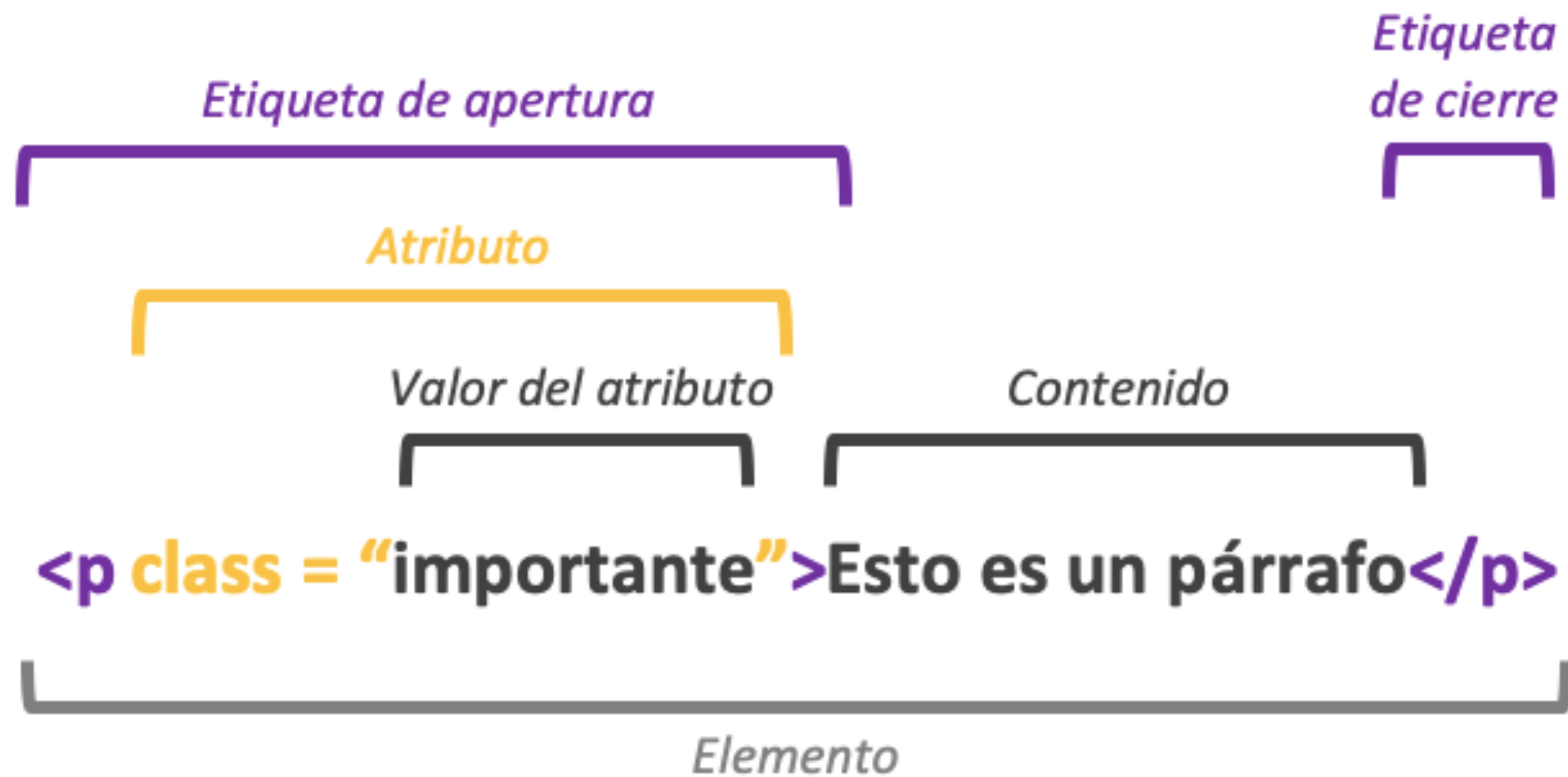
Para definir el contenido de nuestra página web, es necesario hacer uso de HTML (**HyperText Markup Language**).

HTML es un **lenguaje de marcado** que nos permite expresar al navegador web (programa que usamos para navegar en internet, ejemplo: Internet Explorer, Google Chrome, Mozilla Firefox, Safari, etc) lo que queremos que se muestre a través de una **sintaxis basada en etiquetas**.

¿QUÉ SON LAS ETIQUETAS HTML?

- Una etiqueta es un **delimitador** de un elemento de nuestro sitio
- Sirven para **ordenar los elementos** y tener acceso a ellos
- Las etiquetas se escriben entre **< >**
- En general las etiquetas deben tener un **inicio y un final**, es decir una etiqueta que abre y una que cierra.
- La sintaxis es la siguiente para abrir una etiqueta es: **<NOMBRE-DE-ETIQUETA>**
- La sintaxis para cerrarla es: **</ NOMBRE-DE-ETIQUETA>**
- Hay excepciones, algunas solo se utiliza la de abrir.
- El contenido de una etiqueta es lo que se escriba entre el inicio y el fin de esta.
- Separados por espacios encontramos atributos. Estos siguen la sintaxis
- **nombre-atributo="valor-atributo"**.
- Los atributos pueden variar dependiendo del tipo de etiqueta (las imágenes usan `src` y `alt` pero los párrafos no).

TAGS



ESTRUCTURA BÁSICA DE UN CÓDIGO HTML

- Para indicar que nuestro código es HTML la primera etiqueta siempre es: `<!DOCTYPE html>`, esta etiqueta solo se abre.
- Para indicar en donde comienza el sitio, se usa la etiqueta `<html>`, esta etiqueta envuelve todo el programa, al final debe cerrarse `</html>`
- El encabezado del sitio se delimita con la etiqueta `<head>`, esta también debe cerrarse `</head>`
- Dentro del encabezado va la etiqueta del título `<title>`, que también se cierra `</title>`
- Para el cuerpo del sitio se utiliza la etiqueta `<body>`, el contenido de esta etiqueta es toda la **estructura** de nuestro sitio, y se cierra `</body>`

ESTRUCTURA BÁSICA DE UN CÓDIGO HTML

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title></title>  
  </head>  
  <body>  
  </body>  
</html>
```

ESTRUCTURA BÁSICA DE UN CÓDIGO HTML

```
<html>
```

```
<head>
```

```
<title>Page title</title>
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

```
</body>
```

```
</html>
```

COMENTARIOS

Los comentarios son **bloques de texto** que **no se muestran en la página web pero que sirve de información importante** para los programadores que leen/escriben el código.

La sintaxis es `<!--`, seguido del texto que queremos comentar y cerrando el comentario con `-->`. Tener en cuenta que un comentario no se cierra hasta no encontrar la etiqueta de cierre.

Esto no mostraría nada en la web:

```
<!-- Este es un comentario mal cerrado y que no permitirá mostrar el párrafo  
  
<p>  
Este párrafo está bien escrito pero no se mostrará por el comentario mal cerrado  
</p>
```

CSS (CASCADING STYLE SHEETS)

Es un **lenguaje de estilos** que nos permite personalizar la apariencia de los elementos que hemos agregado en nuestro HTML.

Los **estilos se aplican a través de selectores**, que es una sintaxis particular para saber qué elemento queremos personalizar su apariencia. La sintaxis es la siguiente:

```
selector{  
    nombre-propiedad: valor-propiedad;  
}  
  
/*Ejemplo*/  
  
body{  
    background-color: red;  
}
```

ELEMENTOS DE CIERRE AUTOMÁTICO

- `
`
- ``
- `<meta>`
- `<wbr>`
- `<embed>`
- `<input>`
- `<param>`
- `<hr>`
- `<link>`
- `<source>`