

PROMETEO

Unidad 1



A group of people in a modern office setting, smiling and working on laptops. The image is overlaid with a semi-transparent dark blue filter. A large white number '1' is positioned on the left side of the image.

1

Tema 1: Entorno y herramientas para el desarrollo web

PROMETEO

Licencia



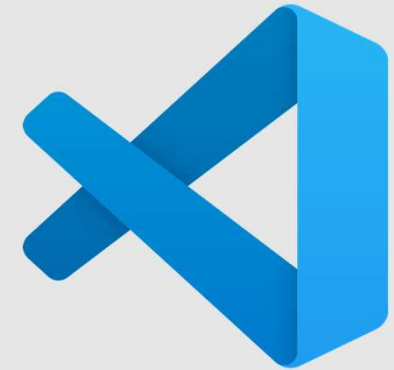
Toda la documentación de esta asignatura queda recogida bajo la licencia de Creative Commons

<https://creativecommons.org/licenses/by-nc-nd/4.0/>

En el caso de incumplimiento o infracción de una licencia Creative Commons, el autor, como con cualquier otra obra y licencia, habrá de recurrir a los tribunales. Cuando se trate de una infracción directa (por un usuario de la licencia Creative Commons), el autor le podrá demandar tanto por infracción de la propiedad intelectual como por incumplimiento contractual (ya que la licencia crea un vínculo directo entre autor y usuario/licenciatarario). El derecho moral de integridad recogido por la legislación española queda protegido aunque no aparezca en las licencias Creative Commons. Estas licencias no sustituyen ni reducen los derechos que la ley confiere al autor; por tanto, el autor podría demandar a un usuario que, con cualquier licencia Creative Commons, hubiera modificado o mutilado su obra causando un perjuicio a su reputación o sus intereses. Por descontado, la decisión de cuándo ha habido mutilación y de cuándo la mutilación perjudica la reputación o los intereses del autor quedaría en manos de cCutacia Juez o Tribunal.

01

Elección de un entorno: Visual Studio Code



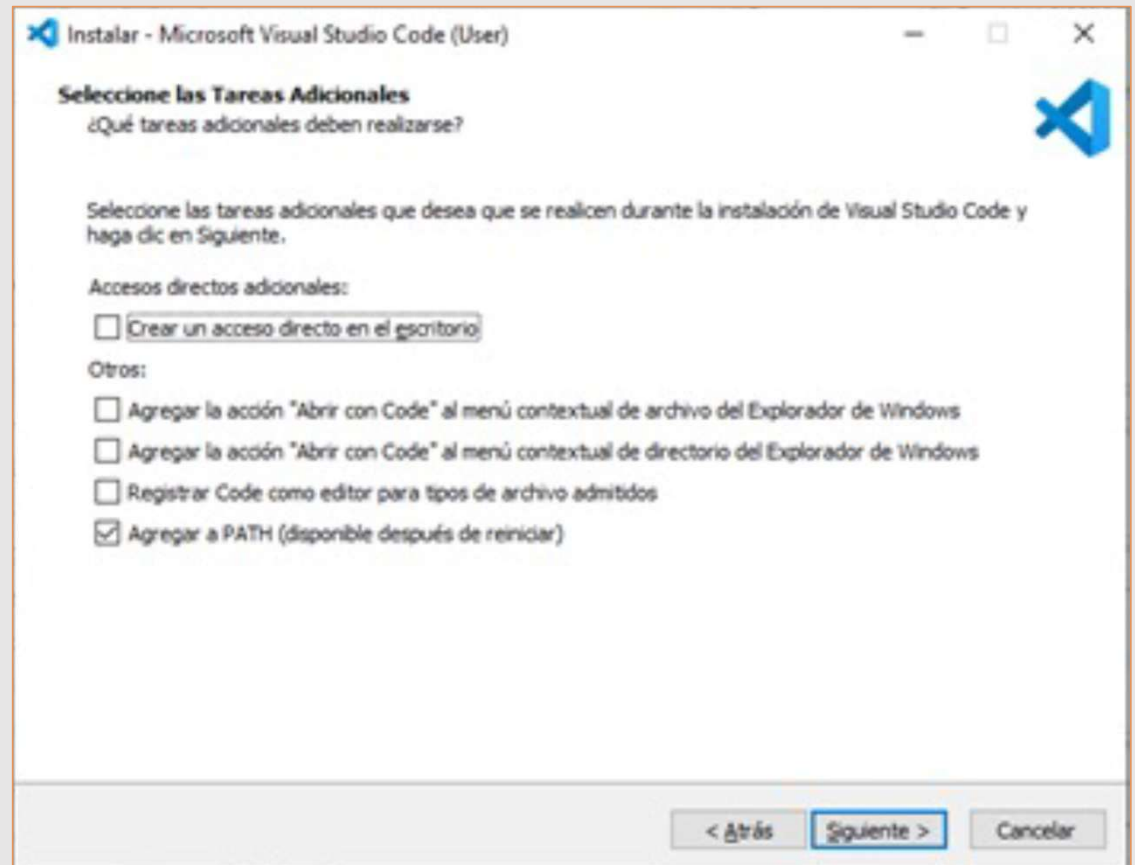
Visual Studio Code es uno de los editores de código libres más conocidos en todo el mundo. Su sencillez de uso y su interfaz gráfica son algunos de sus muchos puntos fuertes.

Con su instalación se incorporan los lenguajes de JavaScript (junto con TypeScript y Node.js), HTML y CSS, aunque con las extensiones es posible utilizar lenguaje Java, C++, PHP o Python. Así que es muy versátil y podemos adecuarlo a nuestras necesidades de una forma sencilla.

Descarga e instalación: <https://code.visualstudio.com/> Disponible para Linux, Mac y Windows

01 Instalación

Ejecutar el fichero que hemos descargado. La instalación es muy sencilla. Simplemente debemos ir aceptando y avanzando en el proceso de instalación.



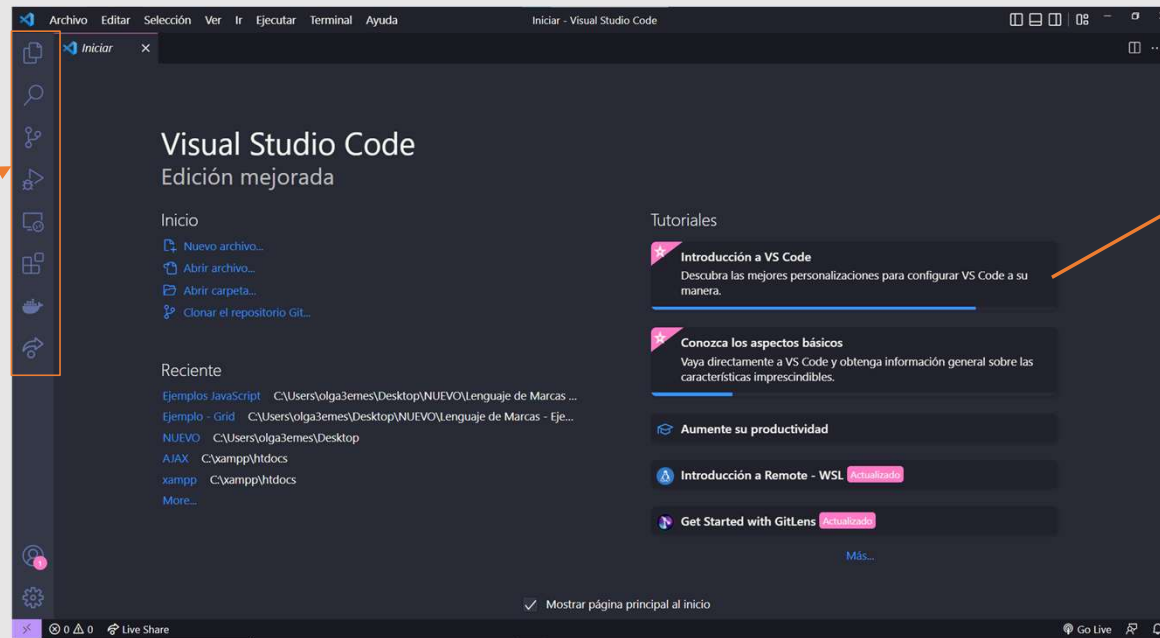
PROMETEO

01

Visual Studio Code

- Abrir Visual Studio Code. Nos aparecerá una ventana como la siguiente.

Aquí me salen más cosas, vosotros al principio tendréis menos, no pasa nada.

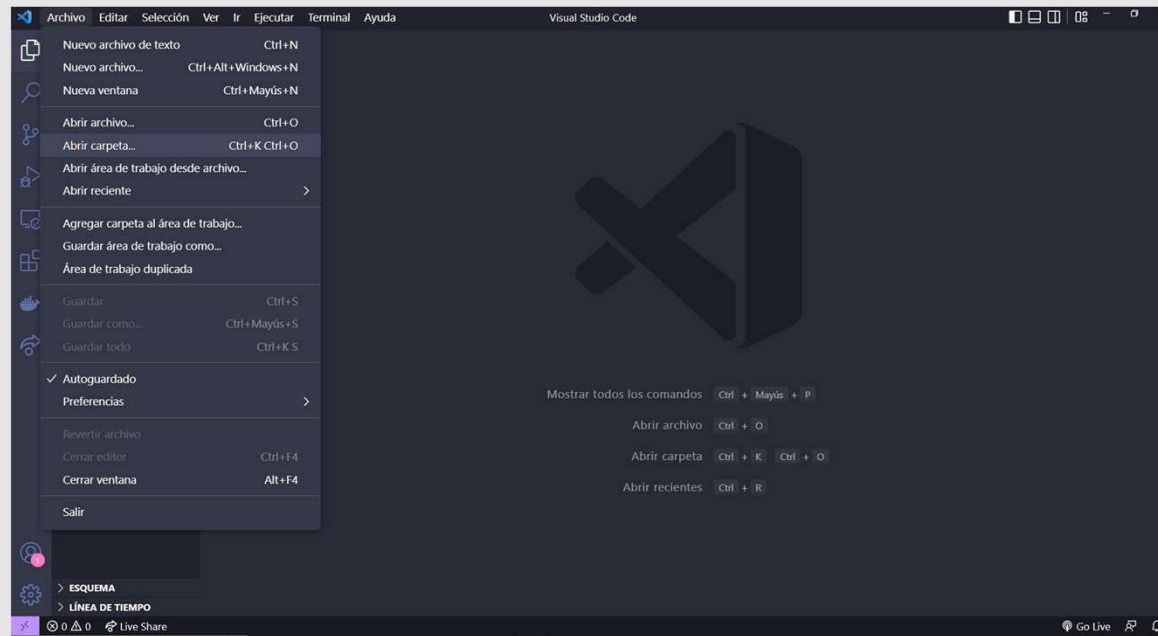


Tutoriales muy recomendables

01

Visual Studio Code

- Crear un nuevo fichero (el cual podemos guardar con extensión .js, .html, .py, etc), abrir uno existente o, por ejemplo, abrir una carpeta donde tengamos nuestros ficheros.

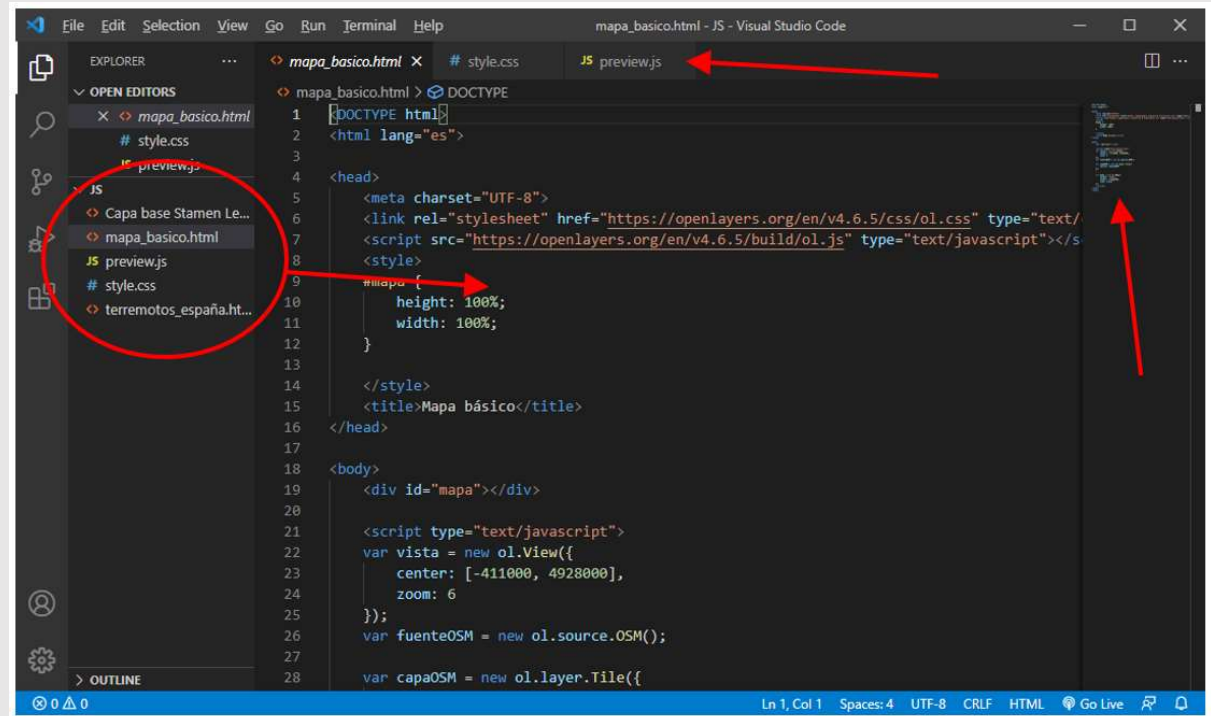


PROMETEO

01

Visual Studio Code

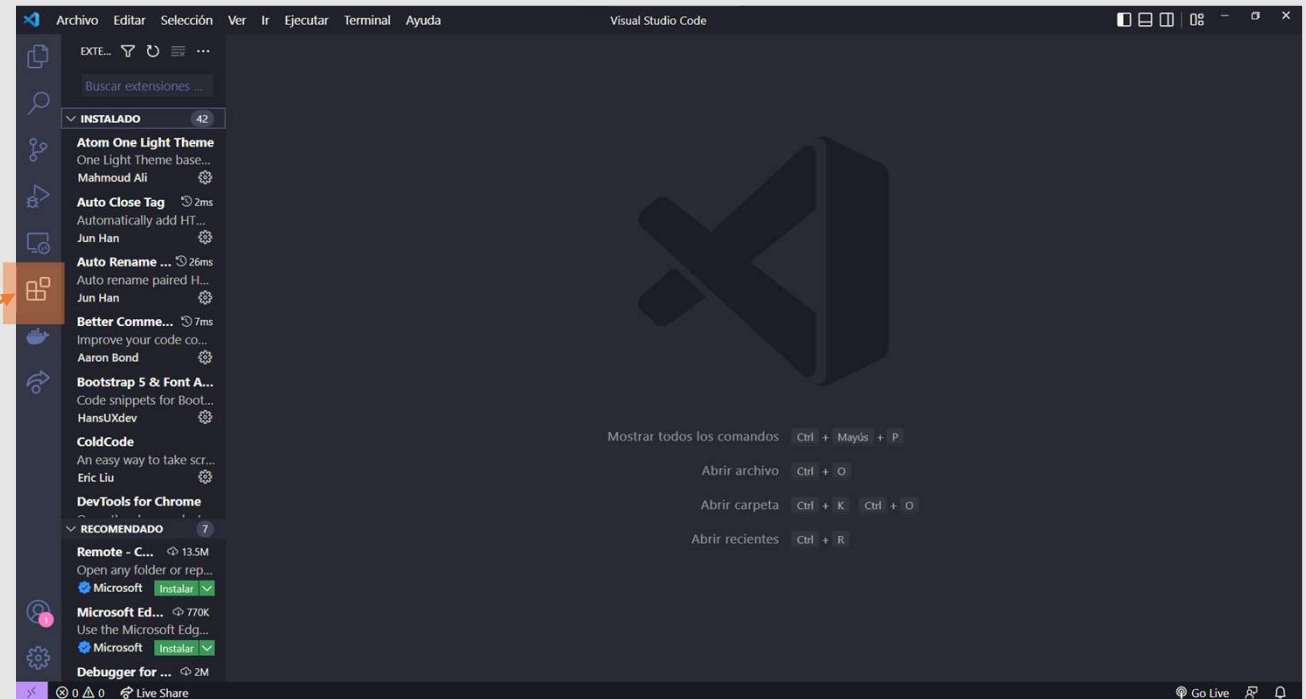
- Arrastrar desde el explorador de la parte izquierda los ficheros que queramos editar hacia el panel de edición. Podremos navegar fácilmente entre ellos utilizando la barra superior. En la parte derecha nos aparece un “minimapa” para localizarnos dentro del código que estamos editando.



PROMETEO

01 Extensiones

Pulsando sobre la opción extensiones, podemos instalar algunos complementos:



01

Extensiones

<https://marketplace.visualstudio.com/items?itemName=formulahendry.auto-rename-tag>

<https://marketplace.visualstudio.com/items?itemName=formulahendry.auto-close-tag>

<https://marketplace.visualstudio.com/items?itemName=aaron-bond.better-comments>

<https://marketplace.visualstudio.com/items?itemName=walkme.HTML5-extension-pack>

<https://marketplace.visualstudio.com/items?itemName=ecmel.vscode-html-css>

<https://marketplace.visualstudio.com/items?itemName=Zignd.html-css-class-completion>

<https://marketplace.visualstudio.com/items?itemName=xabikos.JavaScriptSnippets>

<https://marketplace.visualstudio.com/items?itemName=eamodio.gitlens>

<https://marketplace.visualstudio.com/items?itemName=AnbuselvanRocky.bootstrap5-vscode>

<https://marketplace.visualstudio.com/items?itemName=Meezilla.json>

<https://marketplace.visualstudio.com/items?itemName=christian-kohler.path-intellisense>

<https://marketplace.visualstudio.com/items?itemName=JakeWilson.vscode-picture>

<https://marketplace.visualstudio.com/items?itemName=PKief.material-icon-theme>

<https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer>

<https://marketplace.visualstudio.com/items?itemName=ms-vscode.live-server>

<https://marketplace.visualstudio.com/items?itemName=AnbuselvanRocky.bootstrap5-vscode>

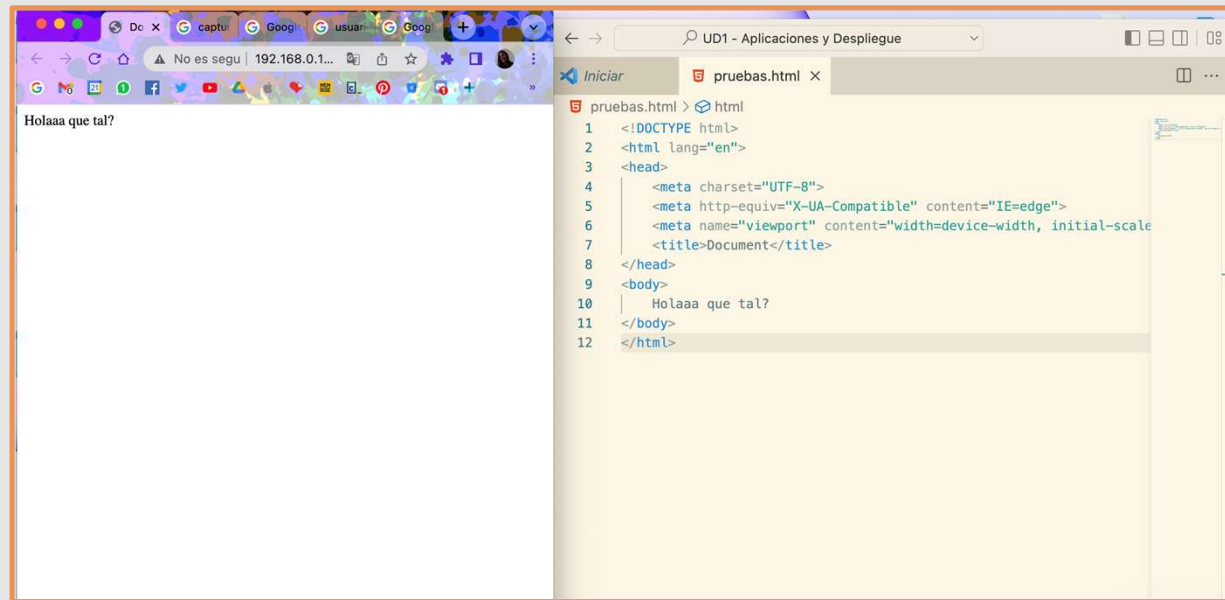
<https://marketplace.visualstudio.com/items?itemName=carbonidl.emmet-jss>

- Poco a poco las iremos poniendo en práctica...

01

Live server

- Esta aplicación nos permite actualizar cambios automáticamente sin tener que estar refrescando la página.
- Una vez instalado (diapositiva anterior), lo que haremos será abrir de nuevo nuestro archivo pruebas.html y haremos clic en la parte inferior derecha donde indica Go Live o pulsaremos botón derecho y "Open with Live Server" (Alt+L+O).
- A continuación, se nos abrirá una página donde cualquier modificación que hagamos se cambiará automáticamente:

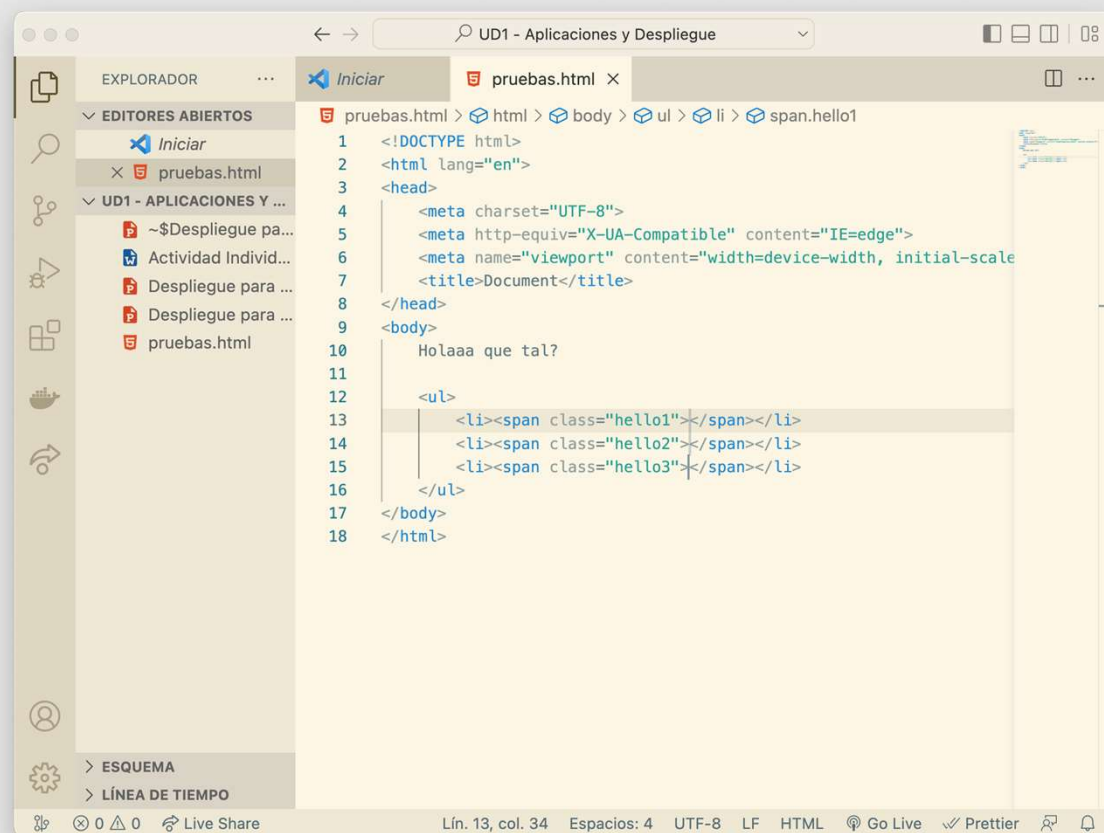


01 Emmet

Emmet es una herramienta que viene instalada por defecto en su versión 2.0 que nos permite escribir código HTML de una forma más ágil, por ejemplo si añadimos:

```
ul>li*3>span.hello$
```

Y hacemos tabulación, obtendremos lo siguiente:



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <title>Document</title>
8 </head>
9 <body>
10   Holaaa que tal?
11
12   <ul>
13     <li><span class="hello1"></span></li>
14     <li><span class="hello2"></span></li>
15     <li><span class="hello3"></span></li>
16   </ul>
17 </body>
18 </html>
```

01

Emmet

Ahora que ya hemos instalado Visual Studio Code vamos a probar EMMET y estos son algunos atajos útiles para HTML:

```
div.principal>p*3>lorem  
img  
p[title='presentacion']{Este es mi texto}  
    header>nav>ul>li*5>a:link[target='_blank'].link_menu{Menu #}$  
link:css  
link:favicon  
meta:vp
```

Documentación Oficial:

<https://docs.emmet.io/cheat-sheet/>

01

Atajos para programar

- Ctrl + Shift + P: abre la paleta de comandos.
- Ctrl + Shift + N: abre una nueva ventana
- Ctrl + Shift + W: cierra una ventana
- F11: activa y desactiva la pantalla completa.
- Ctrl + O: permite abrir con VS Code un archivo existente en el dispositivo.
- Ctrl + N: abre un archivo nuevo.
- Ctrl + S: guarda el archivo.
- Ctrl + Shift + S: realiza la acción «Guardar Como».
- Ctrl + K + Ctrl + W: cierra todas las pestañas que estén abiertas.
- Ctrl + Shift + T: permite abrir la última o últimas pestañas cerradas.
- Ctrl + 2: divide la pantalla en dos.

01

Atajos para programar

- `html:5` + `enter`: cuando trabajamos con archivos `.html`, podemos escribir `html:5` nada más empezar a programar para que escriba automáticamente la estructura básica de un archivo HTML.
- `Ctrl` + `Alt` + Flechas: igual que en los casos anteriores, al trabajar con archivos `.html`, podemos emplear esta combinación para que el cursor se multiplique en diferentes líneas, de forma que podamos insertar un mismo contenido en diferentes filas a la vez.
- `Shift` + `Alt` + `F`: da formato al documento.

01

Ya solo queda practicar

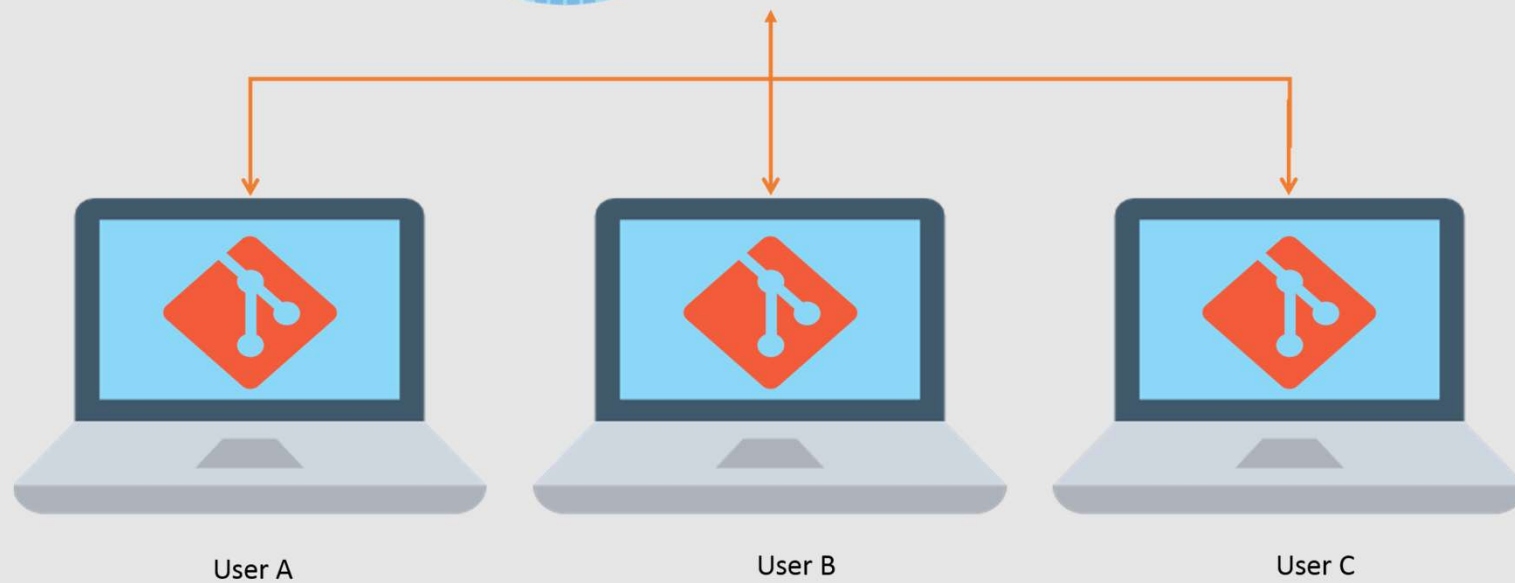
A medida que vayamos avanzando, cada vez será todo más fácil, no hay que tener prisa, solo constancia y ganas de aprender.

¡Tienes talento, así que ahora
solo necesitas constancia para
llegar lejos!

PROMETEO

01

Git y GitHub



PROMETEO

01 Git



¿Por qué Git?

- **Git es un sistema libre.** No puede haber limitaciones contractuales sobre el uso de Git. Sea cual sea el número de colaboradores, de proyectos o actualizaciones, Git siempre será gratuito.
- Git posee también todos los beneficios derivados de los **DVCS**. Es decir, que es posible trabajar en modo desconectado con su repositorio manteniendo las características avanzadas de colaboración cuando se está conectado.
- Git tiene un **rendimiento excelente**. Git incorpora muchas herramientas internas para mejorar su rendimiento (en particular para el uso de las ramas).
- Git es actualmente el sistema más **popular** y gana en popularidad cada año. Es difícil contar con estadísticas precisas sobre los sistemas de gestión de versiones instalados y usados en las empresas.

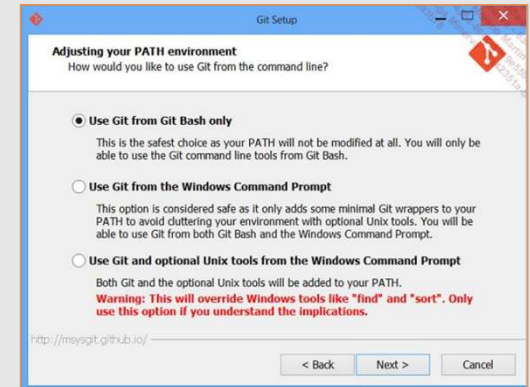
01 Git

Instalación en Linux

- Para instalar Git en un sistema Debian a partir de paquetes preexistentes, se puede utilizar el gestor de paquetes apt-get. Después de haber abierto el CLI, debemos utilizar el siguiente comando para instalar Git: `sudo apt-get install git`
- Una vez que la instalación ha terminado, para probar la instalación y obtener la versión de Git que fue instalada, debemos utilizar el siguiente comando: `git --version`.

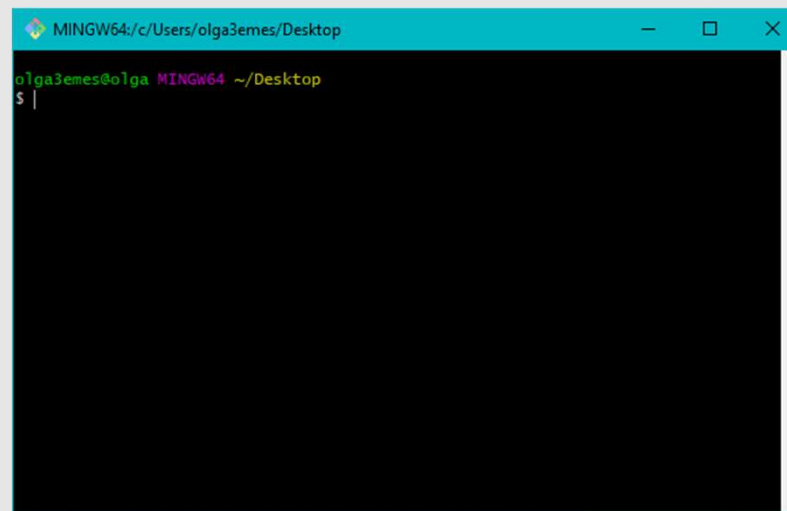
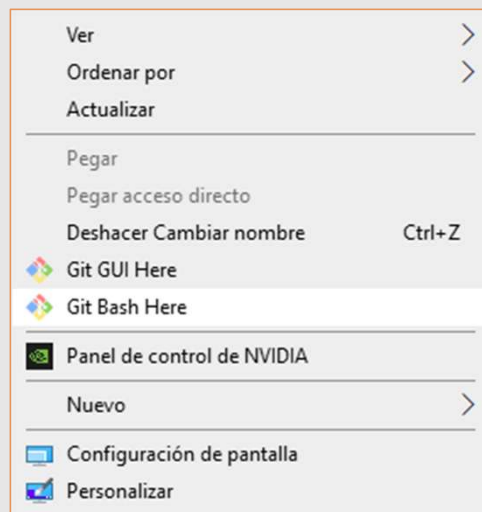
Instalación en Mac y Windows

- Para descargar Git debemos ir al sitio oficial, en la dirección : <https://git-scm.com/>
- A partir de la información enviada por el navegador, detecta la versión que se ha de instalar y ofrece un enlace de descarga en la parte inferior derecha en una imagen de Mac o Windows. Hay que descargar la aplicación. Y seguir el instalador.
- Con Mac también podemos usar brew .



01 Git

Una vez instalado Git, en nuestro sistema Windows aparecerá algo parecido a:



Mientras que en Linux y Mac no tiene por qué aparecer, dado que sus terminales ya soportan los comandos necesarios del kernel de Linux.

01

Git y GitHub

¿Qué es Github?

Plataforma para la gestión de repositorios Git y desarrollo colaborativo.

Para hacer uso de Github, nos basta con registrarnos en:
<https://github.com/>

Al crear una cuenta debemos verificar nuestro correo electrónico.

Se recomienda encarecidamente el uso de una dirección de correo electrónico profesional.

<https://learn.microsoft.com/es-es/shows/visual-studio-code/using-git-with-visual-studio-code-official-beginner-tutorial>



Uso de Git y GitHub de forma profesional

Apuntes extra de cómo utilizar Git – GitHub de forma avanzada desde ventana de comandos

Configuración con clave SSH

Verifica si ya existe una clave SSH

- Primero, revisa si ya tienes generadas claves SSH para tu maquina. Abre la terminal e ingresa el siguiente comando:

`ls -al ~/.ssh`

- Si ya generaste claves SSH, deberías ver una respuesta similar a esta:

```
-rw----- 1 user_name user_name 1766 Jul 7 2018 id_rsa
-rw-r--r-- 1 user_name user_name 414 Jul 7 2018 id_rsa.pub
-rw----- 1 user_name user_name 12892 Feb 5 18:39 known_hosts
```

- Si tus claves ya existen, avanza a la diapositiva “Copiar tu clave SSH pública”

Configuración con clave SSH

- Si no ves ninguna respuesta o recibes un mensaje de que el archivo o directorio no existe (No such file or directory), entonces ejecuta en la terminal:

```
mkdir $HOME/.ssh
```

- Luego genera un nuevo conjunto de claves con:

```
ssh-keygen -t rsa -b 4096 -C your@email.com
```

- Ahora revisa que tus claves existan con el comando `ls -al ~/.ssh` y asegúrate de que la salida de la terminal es similar a la mostrada mas arriba.

- Las claves SSH siempre son generadas como un par de claves públicas (id_rsa.pub) y privadas (id_rsa). Es extremadamente importante que nunca reveles tu clave privada, y solo uses tu clave pública para cosas como la autenticación en GitHub.

Agrega tu clave SSH al ssh-agent

`ssh-agent` es un programa que se inicia cuando inicias sesión y almacena tus claves privadas. Para que funcione correctamente, necesita estar ejecutándose y tener una copia de tu clave privada.

- Primero, asegúrate de que `ssh-agent` esta ejecutándose con:

`eval "$(ssh-agent -s)"` # for Mac and Linux

o:

`eval `ssh-agent -s``

`ssh-agent -s` # for Windows

- Luego, añade tu clave privada al `ssh-agent` con:

`ssh-add ~/.ssh/id_rsa`

Copia tu clave SSH pública

A continuación, debes copiar tu clave SSH pública en el portapapeles:

- Para Linux o Mac, imprime el contenido de tu clave pública a la consola con:

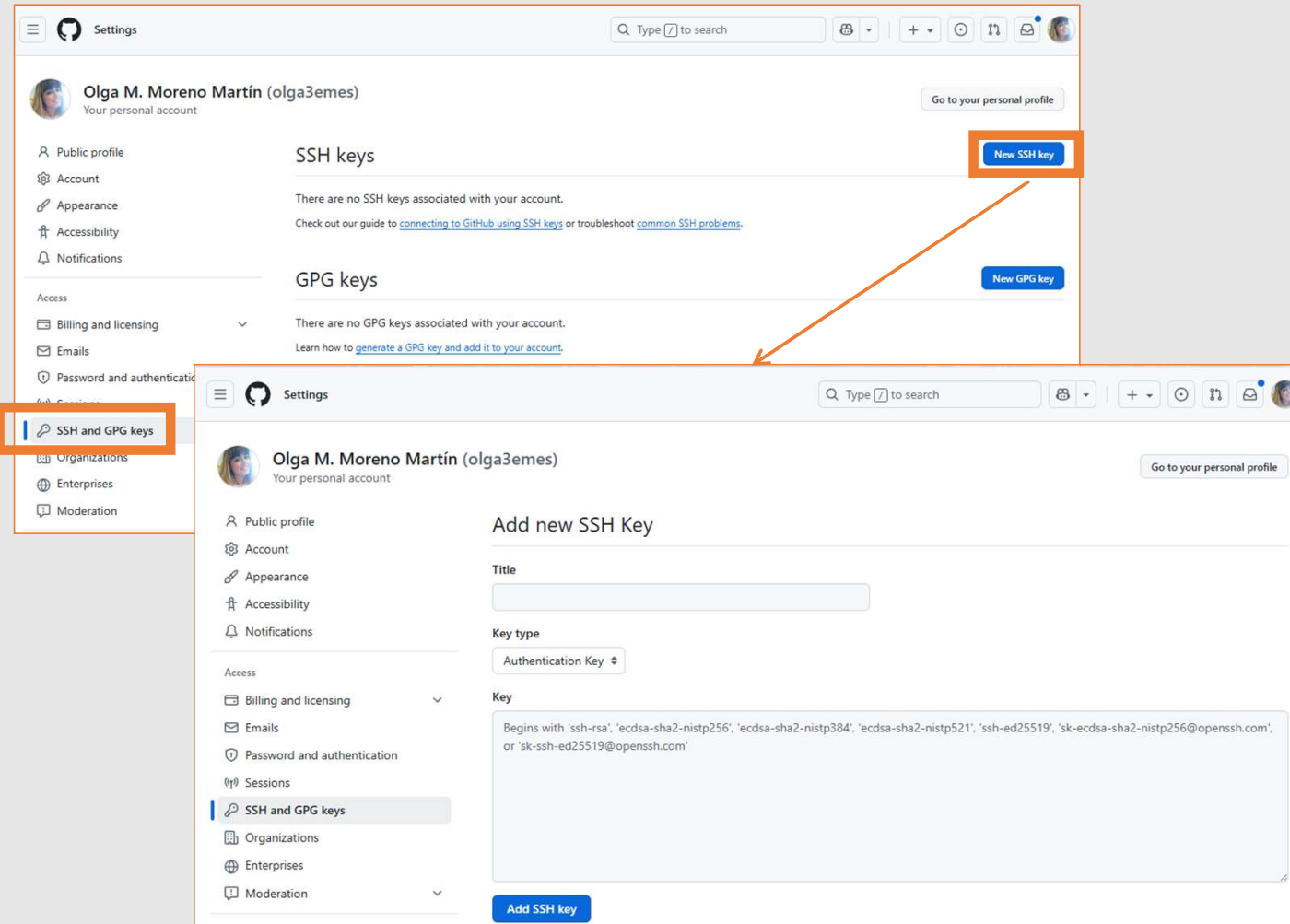
```
cat ~/.ssh/id_rsa.pub # Linux
```

- Luego selecciona y copia la respuesta.
- O para Windows simplemente ejecuta:

```
clip < ~/.ssh/id_rsa.pub # Windows
```

Añade tu clave pública SSH a GitHub

- Ve a la página de ajustes (Settings) de GitHub, click en SSH and GPG keys y en dar al botón "New SSH key":
- Luego ingresa un nombre reconocible y pega tu clave pública (id_rsa.pub) :



Comprueba la autenticación

- Finalmente, en la terminal vamos a ver si todo ha ido bien con:

```
ssh -T git@github.com
```

- Si seguiste correctamente todos los pasos, deberías ver este mensaje:

```
Hi your_user_name! You've successfully authenticated,  
but GitHub does not provide shell access.
```

Comandos básicos

Después de haber instalado Git, es muy útil saber cómo acceder a su ayuda. Puede ser necesario obtener más información sobre un comando y para ello necesitaremos la ayuda. Para mostrar la ayuda general que lista los comandos principales de Git, hay que utilizar el siguiente comando:

```
git --help
```

Este comando permite tener una lista y una breve explicación (en inglés) de los comandos principales de Git.

Para obtener una ayuda más completa sobre un comando específico de Git, hay que utilizar la siguiente sintaxis:

```
git comando --help
```

Git propone una serie de comandos que se dividen en dos categorías:

- Los comandos de porcelana (porcelain command): Estos son los comandos de alto nivel. Son los comandos que se utilizan a diario y para los que Git propone una serie de controles. Ej.: git commit o git add.
- Los comandos de fontanería (plumbing commands): Estos son los comandos de bajo nivel, es decir, que manipulan directamente la información que constituye el núcleo de Git. Estos comandos pueden ser destructivos para el repositorio y suelen ser aplicados por usuarios que dominan Git.

Para ver todos hay que usar:

```
git help -a
```


Comandos básicos

Configuración requerida

Existen dos parámetros de configuración que son necesarios antes de cualquier uso de Git. Estos parámetros de configuración son la dirección de e-mail del usuario del repositorio y su nombre.

1. Configurar el nombre de usuario

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

2. Configurar el e-mail de usuario

```
git config --global user.email email@dominio.com
```

Ya estamos listos para inicializar un repositorio local, usaremos:

```
git init
```

¡AVISO! Si tu ordenador está compartido con otros compañeros es muy posible que si usas Windows, necesites dirigirte al Administrador de credenciales y borrar los datos de aquellos usuarios que no estén en este momento utilizando el PC.

Comandos básicos

```
MINGW64:/c/Users/olga3emes/Desktop/repositorio/.git

olga3emes@olga MINGW64 ~/Desktop
$ mkdir repositorio

olga3emes@olga MINGW64 ~/Desktop
$ cd repositorio

olga3emes@olga MINGW64 ~/Desktop/repositorio
$ git init
Initialized empty Git repository in C:/Users/olga3emes/Desktop/repositorio/.git/

olga3emes@olga MINGW64 ~/Desktop/repositorio (master)
$ ls -la
total 36
drwxr-xr-x 1 olga3emes 197121 0 jul.  1 20:21 ./
drwxr-xr-x 1 olga3emes 197121 0 jul.  1 20:21 ../
drwxr-xr-x 1 olga3emes 197121 0 jul.  1 20:21 .git/

olga3emes@olga MINGW64 ~/Desktop/repositorio (master)
$ cd .git

olga3emes@olga MINGW64 ~/Desktop/repositorio/.git (GIT_DIR!)
$ ls
config  description  HEAD  hooks/  info/  objects/  refs/

olga3emes@olga MINGW64 ~/Desktop/repositorio/.git (GIT_DIR!)
$ |
```

Comandos básicos

Concepto básico: "Las 3 zonas de un repositorio Git"

Working Copy | Staging Area | Repository

working copy: la carpeta de nuestro proyecto

staging area: donde pondremos los archivos a "trackear" (también se le conoce por index o caché)

repository: donde se almacena toda la información de los cambios

Comparar entre las 3 zonas:

git diff (Working Copy → Staging Area)
git diff --staged (Staging Area → Repositorio)
git diff HEAD (Working Copy → Repositorio)



Extensión para ver los cambios:

<https://marketplace.visualstudio.com/items?itemName=mhutchie.git-graph>

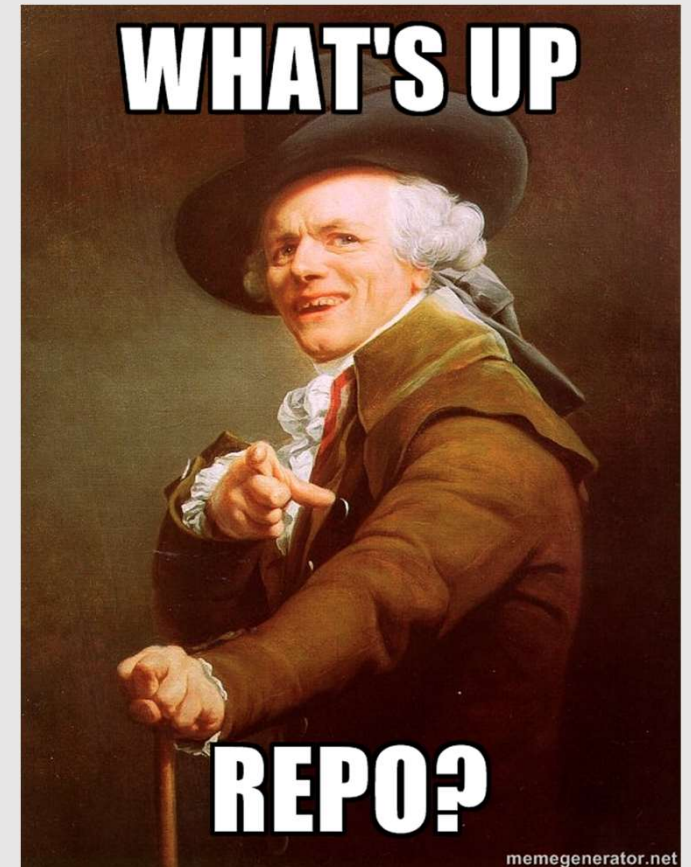
comandos básicos

¿CÓMO ESTÁ MI REPOSITORIO?

git status

Nos resume el estado de nuestro repositorio:

- qué archivos están en el staging area y cuales no
- qué archivos son "trackeados"
- qué archivos han sido modificados



Comandos básicos

DE WORKING COPY A STAGING AREA

`git add`

Añade el archivo al staging area

`git add <filename>`

Añade el directorio al staging area

`git add <folder>`

Añade los archivos .md al staging area

`git add *.md`

¿Y si me equivoco?

`git reset HEAD o <filename> o <folder> o *.md`

`git checkout --<filename>`

Para que sea más fácil:

Añadimos al staging area todos los cambios.

`git add -A / -all`

Añadimos los cambios y nuevos archivos, pero no los eliminados.

`git add .`

Añadimos los archivos modificados y eliminados, pero no los nuevos.

`git add -u`

Comandos básicos

DEL STAGING AREA AL REPOSITORIO:

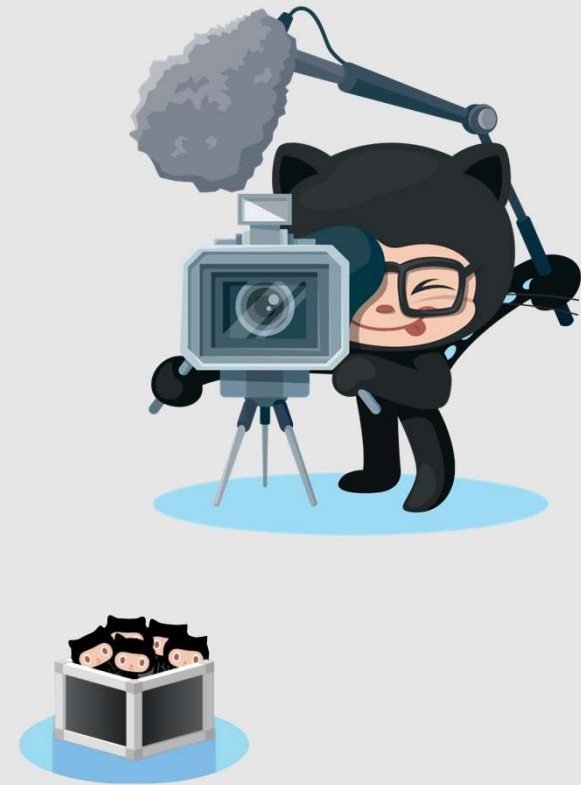
Hacemos un commit

¿Qué es un commit? Hacer un commit es como “hacer una foto”

Vale..., pero ¿qué es un commit?

- Un commit es un paquete que contiene:
- Uno o más «hunks» (el resultado de un diff)
- Un mensaje que describe qué cambios van en este commit
- Un hash SHA para identificar el commit (ID)
- Un enlace con su "commit padre"

`git commit`



Nos abrirá nuestro editor de texto para escribir un mensaje descriptivo del commit. Guardando el comentario como si fuera un archivo, creará el commit.

Hace el commit sin abrir el editor de texto usando el mensaje tras -m.

`git commit -m "Mensaje rápido para el commit"`

¡ATENCIÓN! Sólo se guardará lo que esté en el staging area

Comandos básicos

DESHACIENDO UN COMMIT

Para deshacer el último commit usaremos:

`git reset`

Dos estrategias:

- Deshacer el último commit pero mantener lo que había en mi working copy. Nuestro staging area queda vacío.

`git reset HEAD~1`

(o el ID del commit para retroceder más)

- Deshacer el último commit y lo que había en mi working copy de manera que todo quede como estaba antes. Nuestro staging area queda vacío.

`git reset --hard HEAD~1`

(o el ID del commit para retroceder más)

comandos básicos

VER LOS COMMITS

¿Cómo puedo ver los commits de mi repositorio?

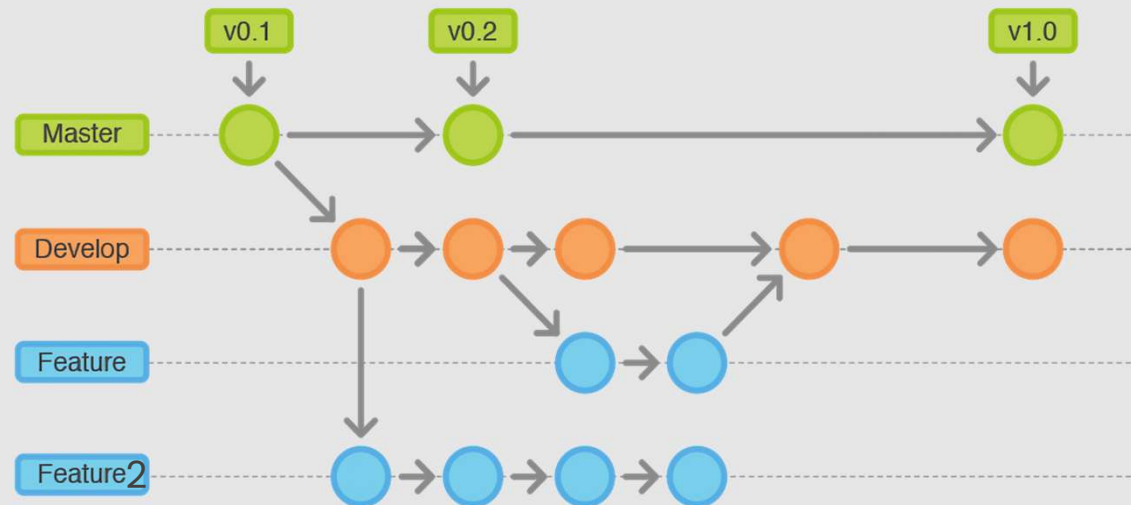
```
git log --graph --decorate --pretty=oneline  
git reflog
```



Ramas

RAMAS O BRANCHES, PARA LOS AMIGOS

- ¿Qué son?
Son directorios virtuales. Universos paralelos con un punto en común.
- ¿Para qué sirven?
Diferentes puntos de entrada al grafo.
Nos permiten desarrollar diferentes cosas en paralelo.



Ramas



¿CÓMO TRABAJAMOS CON LAS RAMAS?

- Ver las ramas existentes:

`git branch`

- Para crear una rama nueva usamos:

`git branch <NombreDeLaRama>`

¡El contenido de working copy y staging area no varía!

- Cambiar de una rama a otra:

`git checkout <NombreDeLaRama>`

¡El contenido de working copy varía!

- Eliminar una rama:

`git branch -D <NombreDeLaRama>`

¡El contenido de working copy y staging area no varía!

- No es más que eliminar un puntero del grafo (un punto de acceso al grafo)
- ¡Ojo! podemos dejar commits “inalcanzables”
- No podemos eliminar la rama en la que estamos

Ramas

MERGING: UNIENDO RAMAS

Para unir dos ramas usaremos:

`git merge <NombreDeLaRama>`

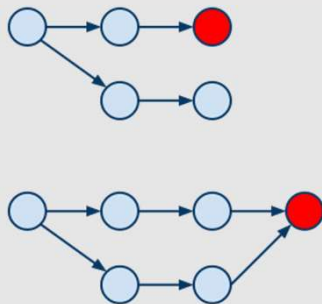
¿Quién absorbe a quién?

- La rama en la que estamos, absorbe la que le indicamos con git merge

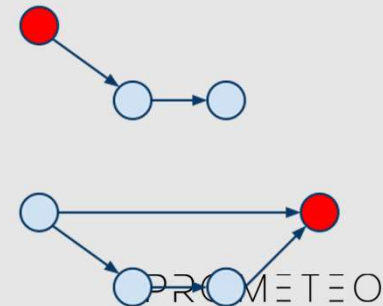
- Ejemplo:

Si estamos en 'carneAsada' y queremos hacer merge con 'salsa'
`git merge salsa`

- Cuando las ramas forman una lista. Hay dos opciones:
Con fast-forward (por defecto)



Sin fast-forward (`--no-ff`)



Ramas

RESOLVER CONFLICTOS

Cuándo hago git pull o git merge, únicamente, mi repositorio dice que hay un conflicto, es decir, da error... ¿Ahora qué hago?

1. Editar cada uno de los archivos en conflicto, quedándonos con el código que realmente nos interesa.
2. Hacer git add de esos archivos al staging-area.
3. Hacer git commit con los cambios.
4. Actualizar el repositorio remoto.

Para cancelar el merge que estábamos haciendo dejando todo como estaba.

`git merge --abort`



Trabajando en remoto

Para **CREAR UN REPOSITORIO REMOTO** podemos usar **Github**.

- Una vez creado podemos clonarlo con:
`git clone <repoUrl>`
- Es decir, copiamos un repositorio remoto para trabajar en nuestra máquina.
- Nos crea una carpeta con el mismo nombre que el repositorio.
- Ésta es la opción más cómoda
- También podemos crear un repositorio remoto usando:
`git remote add origin <UrlRepo>`
Añade <NombreRepo> como repositorio remoto



Olga M. Moreno Martín

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * / Repository name *

Great repository names are short and memorable. Need inspiration? How about [jubilant-chainsaw](#)?

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

☒ Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)
.gitignore template:

☒ Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)
License:

This will set `master` as the default branch. Change the default name in your [settings](#).

Grant your Marketplace apps access to this repository
You are subscribed to 1 Marketplace app

☒ LeanBoard
Whiteboard with sticky notes, connected with your GitHub issue

[Create repository](#)

Trabajando en remoto

Para **PUBLICAR NUESTROS CAMBIOS** en un repositorio remoto usaremos:

```
git push  
git push <NombreRepo> <branch>
```

Sube nuestros cambios en la rama <branch> a <NombreRepo>
Si nuestro repo no está actualizado, se rechazará

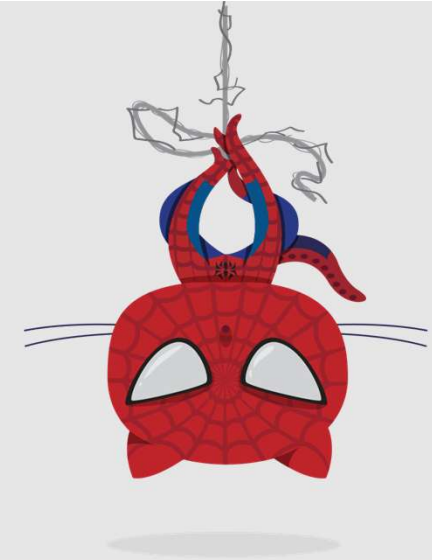
¿Cómo actualizo el repositorio?

```
git fetch  
git fetch <remote>
```

- Descarga todos los cambios de todas las ramas de <remote> desde la última vez que descargamos. Y crea nuevas ramas si es necesario.

```
git fetch <remote> <NombreRamaRemota>
```

- Descarga los cambios de la rama <NombreRamaRemota> de <remote> desde la última vez que descargamos. Crea una nueva rama si no la tenemos.



Trabajando en remoto



DESCARGAR Y APLICAR LOS CAMBIOS REMOTOS

`git pull`
`git pull <remote>`

`git pull` = `git fetch` + `git merge`

Perfecto para hacer actualizaciones directas

Entonces, el orden de las operaciones más corrientes queda:

- 1) `git add --all`
- 2) `git commit -m`
- 3) `git pull`
- 4) `git push`

Etiquetas: Tags y versiones

ETIQUETANDO

- Git tiene la posibilidad de etiquetar puntos específicos del historial como importantes.
- Esta funcionalidad se usa típicamente para marcar versiones de lanzamiento.
- Una etiqueta se une a un commit.

- Listar las etiquetas disponibles en Git es sencillo:

`git tag`

- Crear una etiqueta:

`git tag -a <nombreEtiqueta> -m "mensajito"`

Ejemplo:

```
$ git tag -a v1.4 -m 'my version 1.4'
```

```
$ git tag
```

```
v0.1
```

```
v1.3
```

```
v1.4
```

- La opción `-m` especifica el mensaje de la etiqueta, el cual es guardado junto con ella. Si no especificas el mensaje de una etiqueta anotada, Git abrirá el editor de texto para que lo escribas.

- Eliminar una etiqueta:

`git tag --delete <nombreEtiqueta>`



¡Ya hemos terminado!



Sesión 2

HTML

PROMETEO

Índice

1. HTML

1. Estructura básica HTML
2. Elementos básicos
3. Formatos y etiquetas de texto
4. Atributos
5. Hipervínculos o enlaces
6. Imágenes
7. Listas
8. Tablas

9. Formularios

10. Bloques

2. HTML5

1. Audio
2. Video
3. Barras de progreso
4. Formularios

¿Qué es HTML?

- HTML es un estándar desarrollado por W3C (World Wide Web Consortium). En <https://html.spec.whatwg.org/> se puede consultar su especificación actual, en la cual se basan estas diapositivas.
- HTML son las siglas de **Hyper Text Markup Language**
- HTML es el lenguaje de marcado estándar para crear páginas web
- HTML describe la estructura de una página web
- HTML consta de una serie de elementos
- Los elementos HTML le dicen al navegador cómo mostrar el contenido.
- Los elementos HTML etiquetan partes de contenido como "esto es un encabezado", "esto es un párrafo", "esto es un enlace", etc.

Estructura básica

```
<!DOCTYPE html>  
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title> Título de mi  
página </title>
```

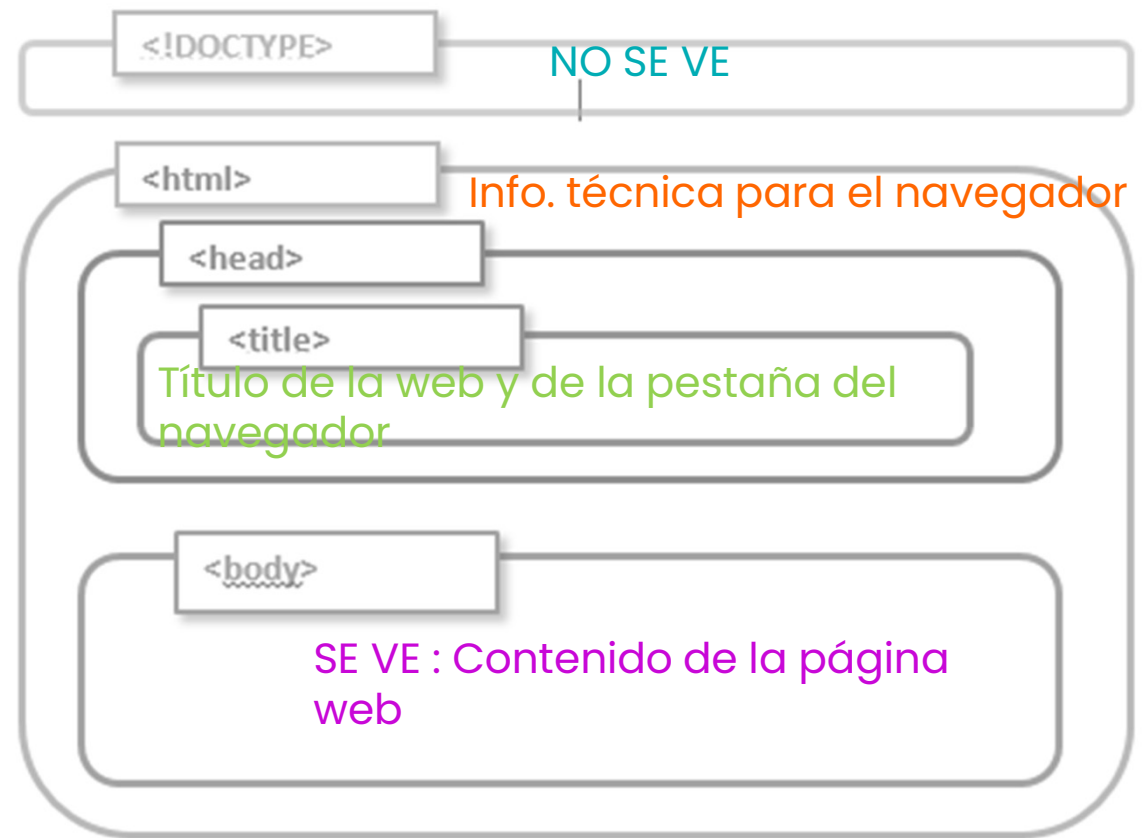
```
</head>
```

```
<body>
```

```
    Mi primer documento  
    HTML
```

```
</body>
```

```
</html>
```



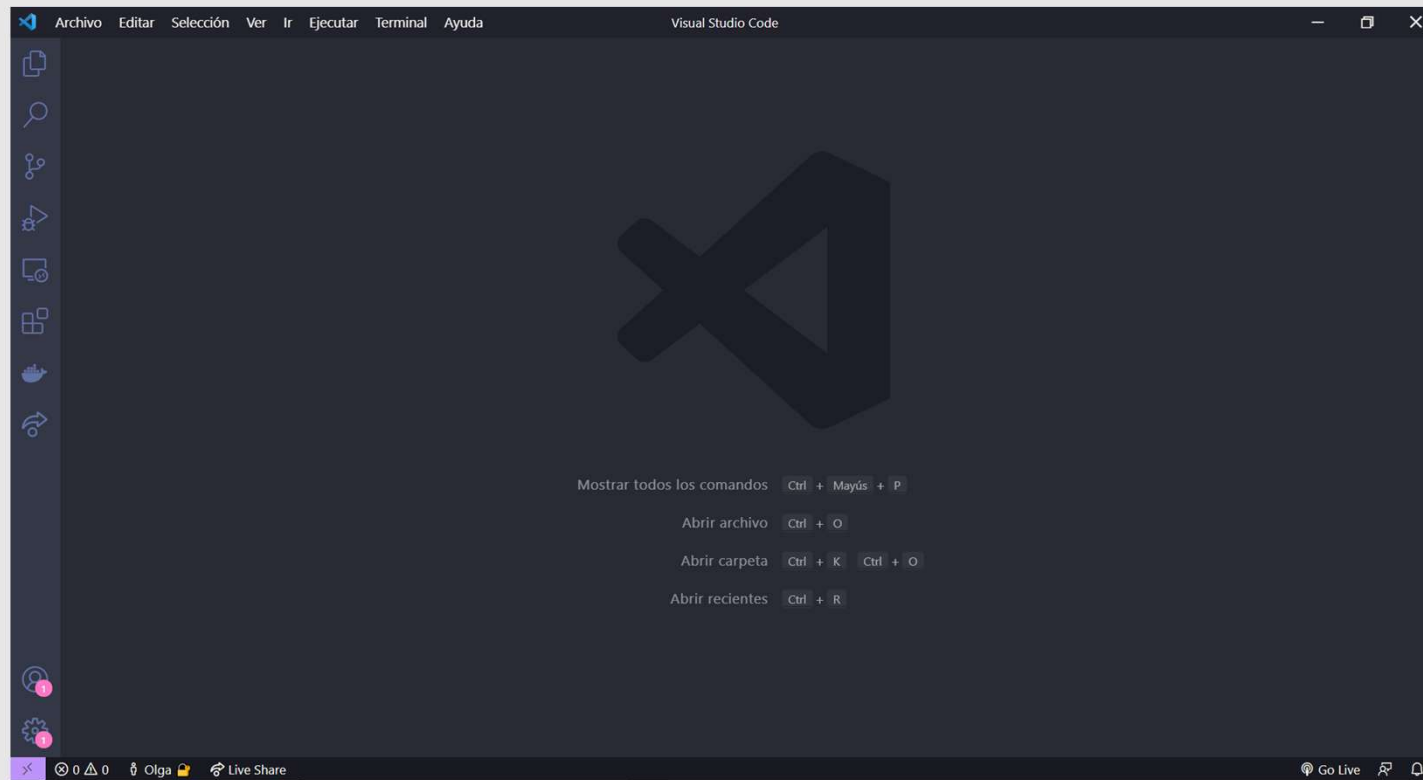
Usaremos Visual Studio Code



Visual Studio Code

Descarga e instalación:

<https://code.visualstudio.com/>



Olga M. Moreno Martín

PROMETEO 51

¿Qué es una etiqueta HTML?

- Una etiqueta es una marca que pone límites a un cierto elemento en HTML, para delimitar un elemento en HTML se emplea una etiqueta de **apertura** y una de **cierre**, son muy característicos por colocarse entre signos de menor y mayor (<>). Existe una cantidad limitada de etiquetas en HTML y ya están establecidas y solo queda aprenderlas y ¡A eso vamos!

<etiqueta> contenido </etiqueta>



Elementos básicos

- Encabezados o títulos
 - a. Los encabezados HTML se definen con las etiquetas <h1> a <h6>.
 - b. <h1>define el encabezado más importante.
 - c. <h6>define el encabezado menos importante.
- Título de la página
 - a. El título de la página web para la pestaña del navegador se define con la etiqueta <title>.
 - b. OJO va en el <head> de la página.
- Enlaces HTML
 - a. Los enlaces HTML se definen con la etiqueta <a>.
 - b. Ejemplo:

```
<a href="https://www.w3schools.com">Esto es un enlace</a>
```

Elementos básicos

Párrafos HTML

- a. Los párrafos HTML se definen con la etiqueta `<p>`.

Etiqueta Salto

- a. `
` indica un salto de línea, observa bien, no está mal escrito, es así como se escribe.
- b. `
` puede ir dentro de un párrafo, por ejemplo para hacer un salto de línea si es necesario.
- c. `
` es una etiqueta de apertura/cierre.

Etiqueta Línea

- a. `<hr/>` es una etiqueta que crea una línea horizontal o separador y también solo hay que digitar tal como aparece, para crear una línea horizontal.
- b. `<hr/>` es una etiqueta de apertura/cierre.

Elementos básicos

Comentarios

- a. Los **comentarios** en HTML nos permiten añadir a nuestro documento, alguna nota, una pequeña descripción, o un simple recordatorio que nosotros entendamos.
- b. Los comentarios no aparecerán en la página, solo será visible para el programador.
- c. Un comentario en HTML se indica de la siguiente manera:

```
<!-- Aquí escribe tu comentario -->
```

Formatos y etiquetas de texto HTML

Etiqueta HTML ****: Para texto en **negrita**.

Etiqueta HTML ****: Para *enfatizar texto*.

Etiqueta HTML **<i>**: Para texto en *cursiva*.

Etiqueta HTML **<u>**: Para texto subrayado.

Etiqueta HTML ****: Para **textos importantes**.

Etiqueta HTML **<big>**: Texto **grande**.

Etiqueta HTML **<small>**: Para texto pequeño.

Etiqueta HTML **<sub>**: Para texto subíndice.

Etiqueta HTML **<sup>**: Para texto superíndice.

Etiqueta HTML **<ins>**: Para texto insertado.

Etiqueta HTML ****: Para texto ~~borrado~~.

Etiqueta HTML **<mark>**: Para texto **marcado**.

Formatos y etiquetas de texto HTML

Diferencia entre `/` y `/<i>`

- a. Las etiquetas `` y `` se muestran visualmente iguales en los navegadores, lo mismo ocurre con las etiquetas `` y `<i>` pero hay diferencias.
- b. `` define texto en negrita, sin ninguna importancia adicional.
- c. `` define texto fuerte, con importancia semántica aumentada.
- d. `<i>` define el texto en cursiva, sin ninguna importancia adicional.
- e. `` define el texto enfatizado, con una importancia semántica aumentada.

Atributos en HTML

Lista de los atributos más usados en HTML:

- a. `id = " "`: Para identificar un elemento único.
- b. `class = " "`: Identificador múltiple.
- e. ~~`align = " "`~~: Alineación de contenido.
- d. ~~`border = " "`~~: Para darle borde al contenido.
- e. ~~`style = " "`~~: Para darle un estilo al contenido.
- f. ~~`background-color = " "`~~: Para color de fondo.
- g. `href = " "`: Para enlaces html.
- h. ~~`height = " "`~~: Para determinar altura.
- i. ~~`width = " "`~~: Para determinar ancho.
- j. `src = " "`: Para imágenes.

TODOS LOS TACHADOS A CSS (Lo veremos más adelante)

Los atributos, nos dan mayor poder al momento de personalizar y agregar ciertos aspecto en nuestras páginas web.

Los atributos HTML van siempre dentro de la etiqueta de apertura y sirven para modificar o agregar alguna característica al contenido.

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

Atributos en HTML

Valor de un atributo

- a. Cuando se hace uso de un atributo HTML se debe especificar su valor, este puede representar una cantidad exacta, característica específica o simplemente en que medida se ejecutará dicho atributo.
- b. El valor se expresa entre comillas después del signo de (=).
- c. Ejemplos:
 1. Id.- su valor puede ser cualquier palabra que sirve para identificar un elemento.
 2. align.- alineación, sus valores pueden ser: center, right, left, justify.
 3. border-style.- estilo de borde, los valores disponibles son: solid, dotted, dashed, double, etc.
 4. background-color: Para color de fondo para ciertos elementos como body, el valor puede ser cualquier color en formato palabra, hexadecimal, RGB, etc. ejemplo: red, green, #ff0000, etc.
 5. href.- para enlaces su valor puede ser la dirección de un carpeta o URL.
 6. height.- para definir altura, su valor puede ser una medida con su unidad de medida, por ejemplo: 10px, 12em, 100%.
 7. width.- define el ancho y su valor puede ser cualquier medida con su respectiva unidad de medida.

Atributos en HTML: id y class

Las **clases** y los **identificadores** facilitan la referencia a elementos HTML desde scripts y hojas de estilo. El atributo de clase se puede usar en una o más etiquetas y CSS lo usa para diseñar. Sin embargo, los identificadores hacen referencia a un solo elemento, lo que significa que el mismo identificador nunca debe usarse dos veces.

Los identificadores se utilizan normalmente con JavaScript y enlaces de documentos interno.

- a. **class** e **id** son atributos globales y, por tanto, pueden asignarse a cualquier elemento HTML.
- b. Los nombres de las clases deben comenzar con una letra (AZ o az) y pueden ir seguidos de letras, números, guiones y guiones bajos.
- c. En HTML5, los atributos **class** e **id** se pueden usar en cualquier elemento. En HTML 4.0.1, se les prohibió a las etiquetas `<base>`, `<head>`, `<html>`, `<meta>`, `<param>`, `<script>`, `<style>` y `<title>`.
- d. Un elemento puede tener una o más clases. Las clases están separadas por espacios y no pueden contener espacios en sí mismas.
- e. Un elemento solo puede tener un identificador y debe ser único en su contexto (es decir, una página web). Los identificadores tampoco pueden contener espacios en sí mismos.

Cómo poner hipervínculos o enlaces en HTML

Para definir enlaces o hipervínculos utilizaremos la etiqueta `<a>` de la siguiente manera:

`<a> texto del enlace `

Atributos para enlaces HTML

a. Existen varios atributos para los enlaces:

i. `href`

ii. `target`

iii. `rel`

Existen muchos más, pero estos son los principales. Ahora vemos como se usan y los valores que pueden tomar.

Cómo poner hipervínculos o enlaces en HTML

- Para indicar la dirección destino del enlace utilizaremos el atributo **href**, En todo caso quedaría de la siguiente manera:

` texto ancla `

- a. El atributo href contiene el valor de la url de destino
- b. El texto ancla es el texto visible, o el texto que contiene el enlace, el texto donde se hará clic.
- c. `<a>` puede ir dentro de un párrafo para un texto ancla específico.
- d. Se puede colocar enlace tanto a texto como a imágenes.

Cómo poner hipervínculos o enlaces en HTML

Atributo **target** para enlaces

- Por defecto, los enlaces se abren en una misma ventana del navegador web, para que el enlace se abra en una nueva ventana el atributo target toma el valor «_blank» de la siguiente manera:
- Los valores de target:
 - a. `_self`: Opción por defecto. Abre el documento en la misma ventana / pestaña en la que se hizo clic
 - b. `_blank` : Abre el documento en una nueva ventana o pestaña
 - c. `_parent` : Abre el documento en el marco principal
 - d. `_top` : Abre el documento en todo el cuerpo de la ventana.
 - e. `external`: Abre un sitio web externo

```
<a href="https://www.url.es/" target="_blank">Este enlace se abrirá en una nueva ventana</a>
```

Cómo poner hipervínculos o enlaces en HTML

Atributo **rel** para enlaces HTML

- El atributo **rel** de HTML sirve para indicar la relación que existe entre la página del cual sale el enlace y la página que recibe el enlace, muy importante para indicar a los motores de búsqueda dicha relación y también a los usuarios.
- Valores para el atributo rel:
 - a. **sponsored**: para indicar que el enlace es un enlace de pago.
 - b. **ugc**: indica que el enlace es un enlace generado por el usuario.
 - c. **nofollow**: indica a los motores de búsqueda que no siga el enlace como si fuera natural.

```
<a href="https://www.twitter.com/" rel="nofollow">Este enlace, es un enlace nofollow de ejemplo</a>
```

Cómo poner hipervínculos o enlaces en HTML

Enlace a un ancla (I)

- Los anclajes se pueden usar para saltar a etiquetas específicas en una página HTML. La etiqueta `<a>` puede apuntar a cualquier elemento que tenga un atributo `id`. Los anclajes se usan principalmente para saltar a una subsección de una página y se usan junto con las etiquetas de encabezado.
- Supongamos que ha creado una página (`page1.html`) sobre muchos temas:

`<h2>First topic</h2>`

`<p>Content about the first topic</p>`

`<h2>Second topic</h2>`

`<p>Content about the second topic</p>`

- Una vez que tenga varias secciones, es posible que desee crear una Tabla de contenido en la parte superior de la página con enlaces rápidos (o marcadores) a secciones específicas.

```
<h2>First topic</h2>
<p>Content about the first topic</p>
<h2>Second topic</h2>
<p>Content about the second topic</p>
```

Cómo poner hipervínculos o enlaces en HTML

Enlace a un ancla (II)

- Si asigno un atributo id a los temas, puedo vincularlos.

a. `<h2 id="Topic1">First topic</h2>`

b. `<p>Content about the first topic</p>`

c. `<h2 id="Topic2">Second topic</h2>`

d. `<p>Content about the second topic</p>`

- Ahora se puede usar el ancla en la tabla de contenido:

`<h1>Table of Contents</h1>`

`Click to jump to the First Topic`

`Click to jump to the Second Topic`

- Estos anclajes también se adjuntan a la página web en la que se encuentran (page1.html). Por lo tanto, puede enlazar el sitio de una página a otra haciendo referencia a la página y al nombre del ancla.

```
<h2 id="Topic1">First topic</h2>
<p>Content about the first topic</p>
<h2 id="Topic2">Second topic</h2>
<p>Content about the second topic</p>
```

```
<h1>Table of Contents</h1>
<a href="#Topic1">Click to jump to the First Topic</a>
<a href="#Topic2">Click to jump to the Second Topic</a>
```

Remember, you can always `look back in the First topic` for supporting information.

Cómo poner hipervínculos o enlaces en HTML

Enlace a una dirección de correo electrónico

- Use mailto: dentro del atributo href para crear un enlace que abra el programa de correo electrónico del usuario (para permitirle enviar un nuevo correo electrónico):

```
<a href="mailto:someone@example.com">Send email</a>
```

Botón como enlace

```
<a href="mailto:someone@example.com">Send email</a>
```

- Para usar un botón HTML como enlace, debe agregar código JavaScript.
- JavaScript le permite especificar qué sucede en ciertos eventos, como hacer clic en un botón:

```
<button onclick="document.location='pagina.html'">HTML Tutorial</button>
```

```
<button onclick="document.location='pagina.html'">HTML Tutorial</button>
```

Imágenes

La etiqueta `` se utiliza para incrustar una imagen en una página web. Las imágenes no se insertan técnicamente en una página web; las imágenes están vinculadas a páginas web. La etiqueta `` crea un espacio de espera para la imagen de referencia. La etiqueta está vacía, solo contiene atributos y **no** tiene etiqueta de **cierre**. Tiene dos atributos:

- a. **src**: especifica la ruta a la imagen
- b. **alt**: especifica un texto alternativo para la imagen

``

```

```


Imágenes

El atributo **src**

El atributo **src** obligatorio especifica la ruta (URL) a la imagen.

Cuando se carga una página web; es el navegador, en ese momento, el que obtiene la imagen de un servidor web y la inserta en la página. Por lo tanto, asegúrese de que la imagen permanezca en el mismo lugar en relación con la página web, de lo contrario, sus visitantes obtendrán un icono de enlace roto.

Imágenes en otra carpeta

Si tiene sus imágenes en una subcarpeta, debe incluir el nombre de la carpeta en el atributo **src**, además del nombre de la imagen.

El atributo **alt**

El atributo **alt** es obligatorio proporciona un texto alternativo para una imagen, si el usuario por alguna razón no puede verla (conexión lenta, un error en el atributo **src** o si el usuario usa un lector de pantalla). El valor del atributo **alt** debe describir la imagen:

```

```

```

```

Imágenes

Tamaño de imagen: **ancho** y **alto**

α. Puede utilizar el atributo `style` para especificar el ancho y el alto de una imagen.

```

```

α. Alternativamente, puede utilizar los atributos `width` y `height`:

```

```

```

```

α. Los atributos `width` y `height` siempre definen el ancho y el alto de la imagen en píxeles.

```

```



Sin embargo, utilizar el atributo `style`, evita que las hojas de estilos cambien el tamaño de las imágenes.

Listas en HTML

Cómo hacer listas

Para empezar, existen dos tipos de listas html en general: **listas ordenadas** y **listas sin ordenar**, a continuación veamos como se implementa cada uno y sus diferencias.

- Listas **no ordenadas** `` en HTML
 - a. Una lista no ordenada en HTML es aquella en la que los elementos no están enumerados ni marcados por algún carácter que designe orden. Los elementos están simplemente listados con alguna viñeta predeterminada como puede ser un punto u otro tipo como veremos más adelante.
 - b. Etiqueta ``
 - i. Una etiqueta `` contiene a todos los elementos de la lista y dentro de ella irán los atributos que afectan a toda la lista en bloque.
 - c. Etiqueta ``
 - i. Los elementos de la lista propiamente dichos, se especifican en una etiqueta ``, de tal manera que cada elemento es independiente de otro elemento.

Listas en HTML

- Ejemplo de lista desordenada en HTML

Como ejemplo demostrativo veamos una lista sin ordenar de las estaciones del año:

```
<ul>
```

```
<li> primavera</li>
```

```
<li> verano </li>
```

```
<li> otoño </li>
```

```
<li> invierno </li>
```

```
</ul>
```

- primavera
- verano
- otoño
- invierno

```
<ul>
```

```
<li> primavera</li>
```

```
<li> verano </li>
```

```
<li> otoño </li>
```

```
<li> invierno </li>
```

```
</ul>
```

Listas en HTML

- Listas ordenadas en HTML ``
 - a. Una lista ordenada en HTML es aquella en la que los elementos están enumerados o marcado con algún signo alfanumérico que indique el orden de los elementos.
 - b. Etiqueta ``
 - c. Se utiliza la etiqueta html `` para crear listas ordenadas, entre la etiqueta de apertura y cierre irán los elementos de la lista dentro de una etiqueta `` respectivamente.
 - d. Etiqueta ``
 - e. En cualquier caso la etiqueta `` sirve para indicar los elementos de la lista, independientemente de si son ordenadas o desordenadas.

Listas en HTML

Ejemplo de listas ordenadas en HTML:

- A continuación veamos un ejemplo concreto de la lista ordenada en HTML de los días de la semana:

```
<ol>  
  <li> lunes</li>  
  
  <li> martes </li>  
  
  <li> miércoles </li>  
  
  <li> jueves </li>  
  
  <li> viernes </ li>  
  
  <li> sábado </li>  
  
  <li> domingo</li>  
  
</ol>
```

```
<ol>  
  <li> lunes</li>  
  
  <li> martes </li>  
  
  <li> miércoles </li>  
  
  <li> jueves </li>  
  
  <li> viernes </li>  
  
  <li> sábado </li>  
  
  <li> domingo</li>  
  
</ol>
```

1. lunes
2. martes
3. miércoles
4. jueves
5. viernes
6. sábado
7. domingo

Listas en HTML

Listas dentro de listas o sublistas

- a. Ingresando etiquetas `` o `` justo después de una etiqueta `` podemos conseguir sublistas o listas dentro de listas.
- Ejemplo de sublistas en HTML:

En el siguiente ejemplos crearemos un lista con sublistas dentro de ella, para ello tomaremos la lista de algunos continentes y dentro sublistas de países o regiones dentro de ellas, solo a modo de ejemplo:

```
<ul>
  <li>América</li>
  <ol>
    <li>América central</li>
    <li>América sur</li>
    <li>América norte</li>
  </ol>
  <li>África</li>
  <ol>
    <li>África Septentrional</li>
    <li>África Occidental</li>
    <li>África Central</li>
    <li>África Oriental</li>
    <li>África Meridional</li>
  </ol>
  <li>Antártida</li>
  <ul>
    <li>Antártida oriental</li>
    <li>Antártida occidental</li>
  </ul>
  <li>Asia</li>
  <li>Europa</li>
  <li>Oceania</li>
</ul>
```

Lista de terminologías o lista de definiciones

Lista de **terminologías** o lista de **definiciones**

Una lista de terminologías o definiciones en HTML con sus respectivas etiquetas, nos servirá para dar un significado semántico a los términos y sus definiciones.

- Etiqueta **<dl>**.- Esta etiqueta se utiliza para crear una lista de definiciones si lo deseamos.
- Etiqueta **<dt>**.- Esta etiqueta contiene el término que queremos definir.
- Etiqueta **<dd>**.- Esta etiqueta contiene la definición del término que le precede.

Todas éstas se usan con su respectiva etiqueta de cierre, veamos un ejemplo:

```
<dl><dt>Mamifero</dt><dd> Especie de animales que maman</dd></dl>
```

```
<dl><dt>Vertebrado</dt><dd> Especie de animales que poseen huesos</dd></dl>
```

```
<dl><dt>Reptil</dt><dd> Especie de animales que se arrastran por el suelo</dd></dl>
```

```
<dl><dt>Mamifero</dt><dd> Especie de animales que maman</dd></dl>  
<dl><dt>Vertebrado</dt><dd> Especie de animales que poseen huesos</dd></dl>  
<dl><dt>Reptil</dt><dd> Especie de animales que se arrastran por el suelo</dd></dl>
```


Listas en HTML

Cómo poner **viñetas** en listas de HTML

Poner viñetas en una lista HTML es tan fácil como poner estilo a nuestro elemento de lista y para ello emplearemos el atributo STYLE con unos valores pensados para este fin, cuando veamos CSS, dejaremos de usar style, veamos:

a. Atributo **style**

Para crear listas con viñetas en HTML utilizamos el atributo style, esto indicará que la lista tendrá un estilo especial, emplea una propiedad y en este caso será la propiedad de list-style-type y básicamente hace referencia al tipo de viñeta, finalmente el valor de dicha propiedad determina el tipo de viñeta que se mostrará, en nuestro caso queremos poner viñetas diferentes a los predeterminados ¿verdad? Pues bien, conozcamos los posibles valores que puede tomar y su respectiva descripción.

Listas en HTML

Valores para tipos de viñeta HTML:

- a. disc: este es el valor predeterminado.
- o circle: muestra marcador es un círculo hueco.
- square: muestra marcador es un cuadrado.
- 2.1. decimal: marcador es un número decimal, que comienza con 1.
- iv. lower-roman: marcador es un número romano en minúscula (i, ii, iii, iv, v, etc.)
- vi. upper-roman: el marcador es un número romano en mayúscula (I, II, III, IV, V, etc.)
- ❖ Además de estos existen más : lower-greek, lower-latin, upper-latin, lower-alpha, upper-alpha, none.

Listas en HTML

- Cómo poner imágenes como viñetas en listas de HTML
 - α. Poner imágenes como viñetas de listas HTML es tan sencillo como emplear la propiedad `list-style-image` y especificar cuál es la imagen en los valores, ya sea indicando la URL de la imagen o su nombre si se encuentra en la misma carpeta del documento HTML

```
<ul style="list-style-image:url(corazon.png);">  
  <li> Primer elemento</li>  
  <li> Segundo elemento</li> </ul>
```

- α. Si la imagen deseada se encuentra ubicada en una URL, entonces colocamos dicha URL en vez del nombre del archivo, de la siguiente manera:

```
<ul style="list-style-image:url(https://cdn.pixabay.com/photo/2012/04/10/22/57/heart-26790__340.png);">...
```

```
<ul style="list-style-image:url(https://cdn.pixabay.com/photo/2012/04/10/22/57/heart-26790__340.png);">...
```

Tablas en HTML

- Como sabemos una tabla es una estructura formada por filas y columnas, las filas son las distribuciones horizontales y las columnas son las distribuciones verticales.

En HTML la estructura de una tabla tiene que ver con etiquetas, una etiqueta `<table>` indica que es un elemento de tabla, dentro de ella va otra etiqueta `<tr>` y esta señala las filas de una tabla y finalmente, dentro de cada fila expresada con etiqueta `<tr>` va otra etiqueta `<td>` y es básicamente un cuadrito de la tabla contenida en una fila y dentro de ella irá el contenido deseado.

- Etiqueta `<table>`

Para definir una tabla en HTML empleamos la etiqueta `<table>`

- Etiqueta `<tr>`

Para crear una fila dentro de una tabla empleamos la etiqueta `<tr>` si deseamos varias filas, entonces crearemos varias etiquetas `<tr>` con su respectiva etiqueta de cierre. Veamos un ejemplo de una tabla con 3 filas.

- Etiqueta `<td>`

Esta etiqueta definirá una columna y está dentro de `<tr>`.

`<td>` Prácticamente serán los contenedores, dentro irán los contenidos de tabla, puede ir texto, imágenes, e incluso más tablas y otros elementos HTML,

Tablas en HTML

```
<table border="1px">  
<tr>  
  <td> elemento a </td>  
  <td> elemento b </td>  
  <td> elemento c </td>  
  <td> elemento d </td>  
  <td> elemento e </td>  
</tr>
```

```
<tr>  
  <td> elemento A </td>  
  <td> elemento B </td>  
  <td> elemento C </td>  
  <td> elemento D </td>  
  <td> elemento E </td>  
</tr>
```

```
<tr>  
  <td> elemento AA </td>  
  <td> elemento BB </td>  
  <td> elemento CC </td>  
  <td> elemento DD </td>  
  <td> elemento EE </td>  
</tr>
```

```
<tr>  
  <td> elemento AAA </td>  
  <td> elemento BBB </td>  
  <td> elemento CCC </td>  
  <td> elemento DDD </td>  
  <td> elemento EEE </td>  
</tr>  
</table>
```

Tablas en HTML

Etiqueta `<th>`

- a. Utilizaremos `<th>` para poner encabezado a la tabla en su primera fila.
- Atributo `colspan` y `rowspan`

En muchas ocasiones es necesario combinar celdas de una tabla, como verás, algunas veces es necesario combinar celdas de una misma fila y en otras celdas de una misma columna.

- Combinar filas en tablas HTML

Para combinar celdas de distintas filas en una tabla HTML emplearemos el atributo `rowspan` dentro del elemento `<td>` desde donde se extenderá la combinación y en su valor colocaremos el número de filas que serán combinadas.

- Combinar columnas de tablas en HTML

Ahora aprenderemos a combinar elementos de la misma fila y de diferentes columnas, es bastante sencillo, simplemente emplearemos el atributo `COLSPAN` en el elemento desde donde se hará la combinación, a partir de ahí se combinará el número de celdas especificada en el valor del atributo `colspan`.

- Etiquetas `<thead>`, `<tbody>` y `<tfoot>` para tablas

Estas etiquetas sirven para identificar la cabecera, el cuerpo y el pie de la tabla, en algunas ocasiones necesitamos definir cuál será la cabecera de nuestra tabla, el cuerpo y el pie de tabla. Además de brindar a los elementos de un significado para los navegadores, estas etiquetas pueden ser útiles para dar estilo a nuestras tablas.

Tablas en HTML

- Ejercicio 1: Realiza el horario de clases utilizando la etiqueta <th> para los días de la semana y el atributo rowspan cuando sea necesario.
- Ejercicio 2: Crea la siguiente tabla:

Rojo	Verde	
Café	Marrón	Tinto
Naranja		
Blanco	Negro	

- Ejercicio 3: usa las etiquetas <thead>, <tbody> y <tfoot> para crear la siguiente tabla:

Paquete 1	Paquete 2	Paquete 3	Paquete 4	Paquete 5
Matemática	Ciencias	Matemática	Sociales	Arte
Inglés	Comunicación	Biología	Historia	Física
Computación	Historia	Economía	Química	Redacción
10 \$	20 \$	30 \$	40 \$	50 \$

- Atributo **bgcolor** para color de tabla: Se puede utilizar el atributo bgcolor para colocar un color de fondo a un elemento, de esa manera podemos poner color a cada elemento de una tabla HTML.

Formularios

Los formularios son elementos de gran importancia y utilidad al momento de recopilar información de nuestros usuarios a través de una página web. Crear un formulario en html implica construir los campos de información, estos pueden ser: campos de texto, de número, selectores, desplegables, casillas de verificación y entre otras más.

Etiqueta `<form>`

Utilizaremos la etiqueta `<form>` para indicar al navegador que el elemento será un formulario, `<form>` se emplea para cualquier tipo de formulario HTML, ya sea uno de contacto, suscripción, registro, login, encuesta, etc.

`<form>...Aquí irán los campos del formulario...</form>`

Etiqueta `<input>`

Crear los campos en el formulario es tan sencillo como utilizar la etiqueta `<input>` antecedita de lo que desees obtener. Por ejemplo:

```
<form>  
  Tu nombre: <input/>  
  Tu apellido: <input/>  
</form>
```

```
<form>  
  Tu nombre: <input/>  
  Tu apellido: <input/>  
</form>
```


Formularios

Atributo type

El atributo type va dentro de la etiqueta `<input>` y su valor indicará que tipo de campo se está creando.

- Valores de atributo type en formularios HTML
 - a. **Text**: para campos de texto
Radio: para casillas de selección
 - b. **Submit**: para botones de envío
Password: para campos de contraseña
 - c. **Reset**: para botones de resetear
File: para campos de selección de archivo
 - d. **Checkbox**: para casillas de verificación

```
<form>
```

```
Nombre:<input type="text"/> <br/><br/>
Apellidos:<input type="text"/> <br/><br/>
Contraseña: <input type="password"/> <br/><br/>
casilla de verificación: <input type="checkbox"/> <br/><br/>
casilla de selección: <input type="radio"/> <br/><br/>
selección de archivo: <input type="file"> <br/><br/>
botón de envío: <input type="submit">
```

```
</form>
```

```
<form>
```

```
Nombre:<input type="text"/> <br/><br/>
Apellidos:<input type="text"/> <br/><br/>
Contraseña: <input type="password"/> <br/><br/>
casilla de verificación: <input type="checkbox"/> <br/><br/>
casilla de selección: <input type="radio"/> <br/><br/>
selección de archivo: <input type="file"> <br/><br/>
botón de envío: <input type="submit">
```

```
</form>
```

Formularios

Crear una desplegable con opciones:

- Etiqueta `<select>`

Utilizaremos la etiqueta `<select>` para crear un campo desplegable, este tipo de campos en HTML permiten seleccionar una opción del desplegable y además son muy útiles en esas ocasiones en las que buscamos buena presentación de espacio en el formulario.

- Etiqueta `<option>`

Utilizaremos la etiqueta `<option>` dentro de la etiqueta `<select>`. `<option>` contiene una opción del desplegable y podemos implementar la cantidad de opciones que deseemos. Por defecto estará seleccionada la primera opción, además es necesario pasar el cursor y clic para desplegar las opciones, una vez seleccionada la opción, esta regresa a su estado inicial pero con la opción seleccionada.

- Atributos para etiqueta `<select>` en formularios HTML
 - a. Atributo **size**: Utiliza el atributo `size` para determinar la cantidad de opciones a mostrar y su valor en números.
 - b. Atributo **multiple**: Utiliza el atributo `multiple` con valor también múltiple para permitir que el desplegable sea de múltiple selección de tal manera que permita seleccionar varias opciones.

```
<form>
    ELIGE UN COLOR:
    <select>
        <option>Blanco</option>
        <option>Negro</option>
        <option>Azul</option>
        <option>Rojo</option>
        <option>Verde</option>
    </select>
</form>
```

```
<form>
    ELIGE UN COLOR:<select>
        <option>Blanco</option>
        <option>Negro</option>
        <option>Azul</option>
        <option>Rojo</option>
        <option>Verde</option>
    <select/>
</form>
```

Formularios

Crear un campo de texto grande en formularios

Un campo de texto con varias líneas disponible en el formulario puede ser necesario para permitir a nuestros usuarios enviar extensos textos y para ello haremos uso de la etiqueta `<textarea>`.

Etiqueta `<textarea>`

La etiqueta `<textarea>` se utilizara para crear un área de texto de varias líneas, puede servirnos por ejemplo para un capo de mensaje, veamos un ejemplo concreto:

- El campo de texto multilínea posee en la esquina inferior derecha un asistente que permite ampliar el campo con el cursor.

```
<form>
  Nombre:
    <input type="text"/> <br/><br/>
  Correo electrónico:
    <input type="email"/> <br/><br/>
  Tu mensaje:
    <textarea name="mensaje"> </textarea> <br/>
<br/>
    <input type="submit" value="Enviar ahora"/>
</form>
```

```
<form>
  Nombre:
    <input type="text"/> <br/><br/>
  Correo electrónico:
    <input type="email"/> <br/><br/>
  Tu mensaje:
    <textarea name="mensaje"> </textarea> <br/><br/>
    <input type="submit" value="Enviar ahora"/>
</form>
```

Formularios

- Atributo **action**

El valor del atributo action será la URL de la página que queremos que cargue después de que se envíe la información de un formulario. Este atributo va dentro de la etiqueta <form> y lo podemos utilizar si deseamos.

```
<form action="URL">
```

- Atributo **method**

El atributo method puede tomar como valores a **GET** o **POST**, pero no los 2, y esto indica el método HTTP que será utilizado cuando se envíe la información. Ejemplo:

```
<form action="URL" method="get">  
<form action="URL" method="post">
```

- a. Cuando se usa el valor get, los datos del formulario podrán ser vistos en la dirección de la página.
- b. Cuando se usa el valor post los datos enviados no serán visibles en la dirección de la página.
- Por seguridad se recomienda utilizar usar el valor post, para datos sensibles.

Formularios

- Atributo **name**

Utiliza el atributo name para ponerle un nombre al formulario

```
<form name="miformulario" action="URL" method="post">
```

Para que el formulario funcione correctamente utilizaremos el atributo name para cada campo del formulario, así será posible recibir correctamente los datos enviados.

- a. El atributo name será necesario para procesar la información de cada campo y que estas lleguen a su destino correctamente.

Hemos aprendido a crear un formulario en HTML estático y además con muchos detalles, sin embargo aún no podemos enviar y recibir información a través de ella, para ello es necesario que los datos se envíen a un servidor y ser procesados.

```
<form>
  Nombre: <input name="nombre" type="text"/> <br/><br/>
  Correo electrónico: <input name="correo" type="email"/> <br/><br/>
  Tu mensaje: <textarea name="mensaje"> </textarea> <br/><br/>
  <input type="submit" value="Enviar ahora"/>
</form>
```

Formularios

- La etiqueta <label>
 - a. La etiqueta label se usa para darle una descripción a un elemento de un formulario.
 - b. El atributo **for** de <label> debe ser **igual** al atributo **id** del elemento relacionado para unirlos. Una etiqueta también se puede vincular a un elemento colocando el elemento dentro del <label> elemento.

```
<form action="#">
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label><br><br>
  <input type="submit" value="Submit">
</form>
```

Características y diferencias de `<div>` y ``

La principal diferencia entre las etiquetas `<div>` y `` es que en el primer caso se define como un elemento en bloque y en la segunda como un elemento en línea.

Etiqueta `<div>`

- a. La etiqueta `<div>` viene de división, prácticamente funciona y sirve de contenedor de bloque, es decir puede contener uno o varios elementos HTML dentro de ella, es bastante utilizada para agrupar varios elementos que luego serán apuntados con CSS con la finalidad de dar un estilo adecuado a dicho grupo, o también puede servir para organizar mejor nuestra página web.
- b. En el pasado, se solía utilizar para organizar las secciones de una página web tales como la cabecera, la navegación o menú, el cuerpo de página, los pies de página, barras laterales, etc. Se le solía poner un atributo `id` o `class` con su respectivo nombre como: cabecera, cuerpo, footer, etc. para luego darle estilo con CSS. Sin embargo desde que apareció HTML5 es mejor emplear las nuevas etiquetas HTML5 como: `<header>`, `<article>`, `<footer>`, etc. para organizar la estructura de una página.

Etiqueta ``

- a. La etiqueta `` también funciona como un contenedor pero en este caso será un contenedor en línea. Por ejemplo para destacar una palabra o expresión dentro de un párrafo utilizaremos la etiqueta `` para encerrar esa palabra o expresión y luego ponerle un estilo mediante un atributo. O también podemos ponerle un identificador, ya sea `id` o `class` para luego apuntarlo y maquetarla con CSS.

Características y diferencias de <div> y

Ejemplo: etiqueta **div**

- a. Ejercicio: crear tres bloques contruidos con la etiqueta <div>, claramente diferenciados por el color de fondo de cada bloque. Hay que establecer las propiedades de altura y anchura.

Ejemplo: etiqueta **span**

- a. Ejemplo de código HTML empleando la etiqueta , en este caso utilizaremos span para resaltar una fracción de texto dentro de un párrafo, de tal manera que el color de letra será de rojo y el fondo de color amarillo.

```
<p>
    Este es un párrafo en el que insertaremos un elemento span a continuación
    para diferenciarlo
    <span style="background: yellow; color: red">
        Texto especial dentro de etiqueta span con fondo amarillo y color rojo
    </span>
    ese texto especial sigue dentro del párrafo y en línea, forma parte del
    párrafo, pero podemos especificar alguna característica
</p>
```




PROMETEO