

A graphic on the left side of the slide. It features a 3D effect with four overlapping rectangular blocks in purple, orange, yellow, and blue. The text 'Agencia de Aprendizaje a lo largo de la vida' is written across these blocks in white. An orange arrow points to the right from the orange block.

Agencia de  
Aprendizaje  
a lo largo  
de la vida

# FULL STACK PYTHON

## Clase 1

HTML 1

# Conceptos básicos de HTML



# Les damos la bienvenida

Vamos a comenzar a grabar la clase

## Clase 00

### Presentación del curso

- Sobre Codo a Codo.
- Objetivo del curso y lineamientos de cursada.
- Conceptos básicos sobre Full Stack.
- Herramientas a utilizar.
- Instalación del software.

## Clase 01

### HTML 1 - Conceptos básicos de HTML

- Conceptos básicos de la web
- Proyecto web: ¿qué es?
- Concepto Cliente/Servidor
- Introducción a HTML. Etiquetas básicas y atributos.

## Clase 02

### HTML 2 - Continuando con HTML

- Listas y enlaces.
- Rutas absolutas y relativas.
- Elementos en bloque y en línea.

# Página, Sitio y Aplicación

# Página Web

Principales características de una página Web

- Es un documento HTML (texto, imágenes, videos, CSS, Javascript).
- Pueden ser estáticas (sólo código) o dinámica (con contenido extraído de una base de datos).



# Sitio Web

## Principales características de un sitio Web

- Son un conjunto de páginas Web estructuradas en un dominio.
- Se pueden crear escribiendo código o mediante CMS o builders.
- Pertenecen a empresas, organizaciones o personas.



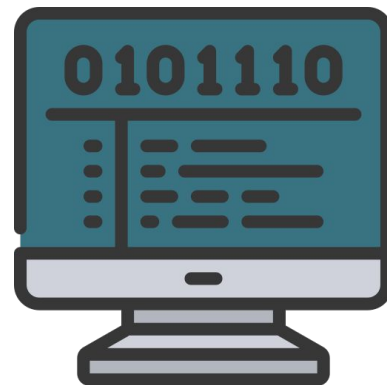
# Aplicación Web

Principales características de una aplicación Web

- Es un software desarrollado con tecnologías Web.
- Se crea mediante código, utilizando diversos lenguajes de programación, librerías, frameworks, etc.
- Son creadas por desarrolladores Web profesionales.

**Fuente:**

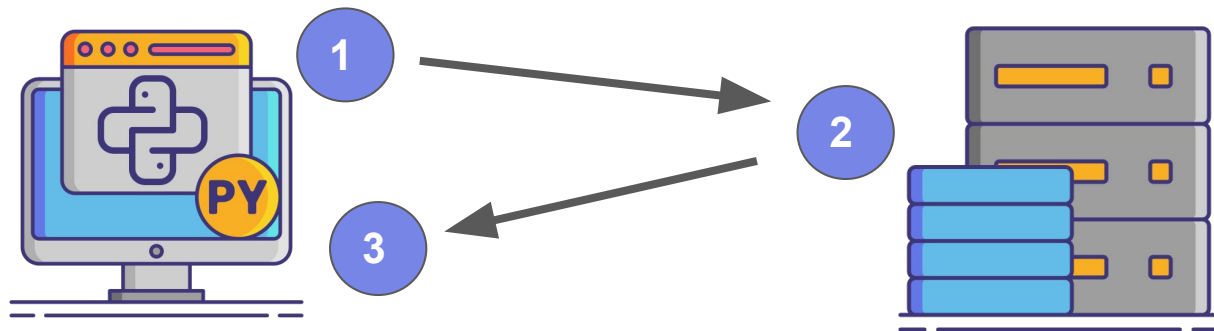
<https://ed.team/comunidad/pagina-web-vs-sitio-web-vs-aplicacion-web>





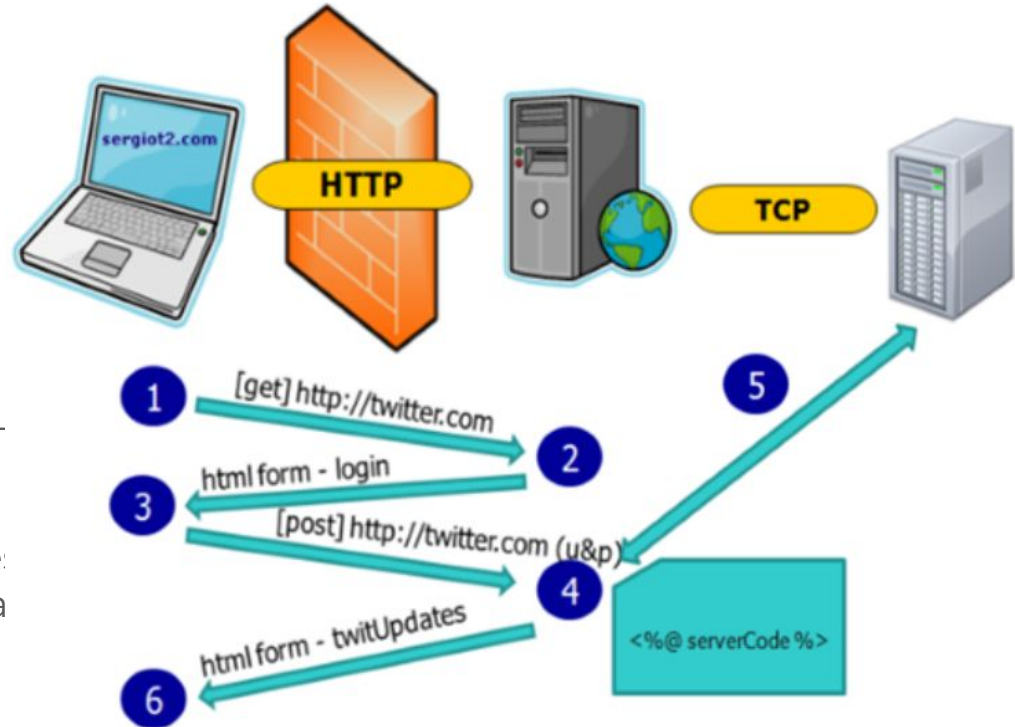
# Aplicación de Escritorio

1. El usuario accede a la aplicación.
2. La aplicación accede a un repositorio de datos para pedir información o para modificarla.
3. La aplicación muestra al usuario la información solicitada.



# Aplicación Web

1. El usuario ingresa la URL (dirección) en el navegador.
2. El servidor web recibe la solicitud y envía la respuesta en HTML al navegador.
3. En el navegador se ingresa la información de usuario y contraseña.
4. Dicha información se traduce a HTML
5. Dicha información se envía al servidor.
6. Si estos datos son válidos el usuario es redireccionado a la página que desea ingresar.



# Ventajas de una Aplicación Web

- Se puede ejecutar desde cualquier dispositivo que cuente con navegador y con conexión a Internet.
- No es necesario instalarlas en el dispositivo, basta con un navegador web.
- Son fáciles de actualizar y mantener.
- Su funcionalidad es independiente al Sistema Operativo instalado en la PC del usuario.
- Permite que las aplicaciones sean multiusuarios.
- Permite separar los datos almacenados en el servidor de base datos, del front end, donde se muestran al usuario.
- Las páginas web poseen estructuras similares que mejoran la experiencia en el uso de las diferentes aplicaciones.

# Aplicación Escritorio vs. Aplicación Web

## Aplicación Escritorio

- Es aquella que está instalada en el dispositivo del usuario.
- Es ejecutada directamente por el sistema operativo del dispositivo, ya sea Microsoft Windows, Mac OS X, Linux, Android, etc.
- Su rendimiento depende de diversas configuraciones de hardware como memoria RAM, disco duro, memoria de video, etc.

## Aplicación Web

- Es aquella que está instalada en un Servidor.
- Su ejecución requiere que el usuario disponga de:
  1. Un dispositivo con conexión a Internet
  2. Un navegador (Google Chrome, Internet Explorer, Mozilla Firefox, etc.)
- El usuario solo necesita renderizar contenido, lo que suele ser menos demandante de poder de cálculo.

# Front-End y Back-End

Concepto Cliente/Servidor

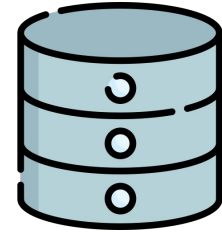
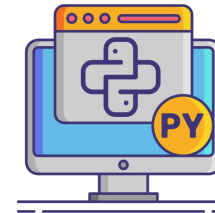
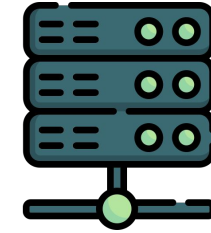
# Front-End

- Es la parte visible por el usuario de una página Web.
- Se relaciona con la creación de una página funcional (computadora / tablet o celular) que puede verse en un navegador (**cliente**).
- HTML, CSS y JavaScript.
- Editor de texto plano y navegador de Internet.



# Back-End

- Corre en el **servidor**.
- Se encarga del manejo de algoritmos y la utilización de bases de datos para guardar o procesar información: usuarios, productos, claves, procesos de selección, filtros, listados y todo tipo de consultas sobre plataformas Web.
- Parte administrativa del sitio Web.
- Bases de datos, Python y frameworks asociados.



# HTML

**HyperText Markup Language:** Lenguaje de marcado de hipertexto.

Define la **estructura**, **semántica** y **contenido** de las páginas Web.

El navegador (cliente) interpreta el HTML y lo representan en pantalla.

Permite agregar texto, imágenes, enlaces, tablas, listas, formularios, etc.

Última versión del estándar: HTML 5.

No todos los navegadores son capaces de interpretar un mismo código de una manera unificada (siempre debemos probar que funcione en la mayoría de ellos).

Utiliza etiquetas (tags) que definen la estructura del **documento**.

Texto plano con extensión .html

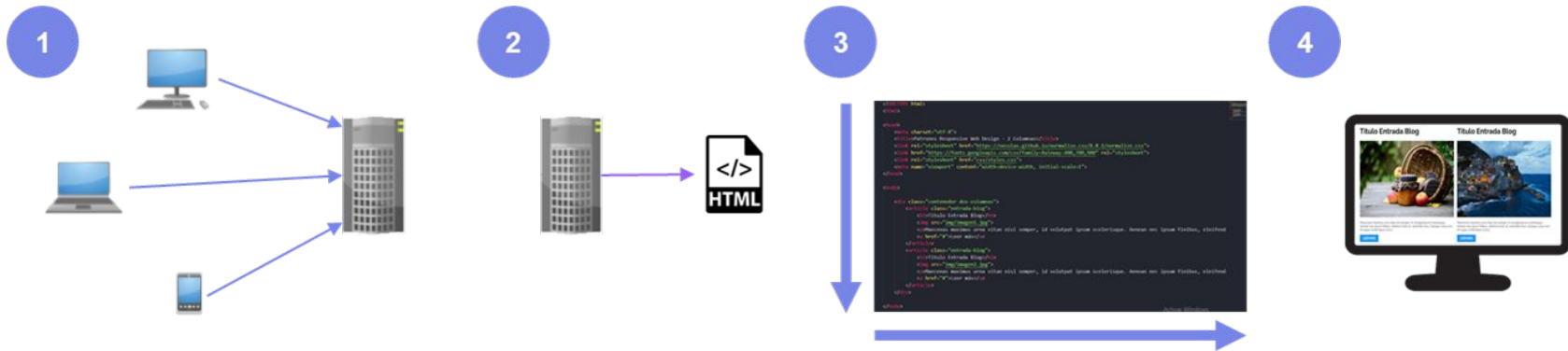
## Hipertexto:

Sistema que permite enlazar fragmentos de textos entre sí. Permite que la lectura no sea lineal, sino que el usuario acceda a la información a través de los ítems relacionados.



# HTML: ¿Cómo funciona?

1. El navegador (**cliente**) le pide información al servidor.
2. El servidor devuelve la información al cliente en un archivo HTML.
3. El navegador en el cliente lee el archivo de arriba hacia abajo y de izquierda a derecha para interpretar la información.
4. Tiene en cuenta las etiquetas que tiene el documento y las va renderizando en la pantalla (lo que se ve en el navegador).

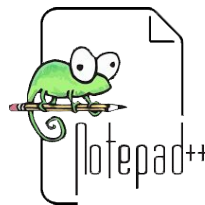


# Editores de texto ¿qué usamos para escribir HTML?

Necesitamos utilizar un editor de texto, que genere archivos de “texto plano”.

Hay muchos disponibles, desde el bloc de notas de tu sistema operativo hasta algunos multiplataforma como Visual Studio Code.

Nota: pueden hacer clic en las imágenes para descargar cada software.



Visual Studio Code

# Página Web

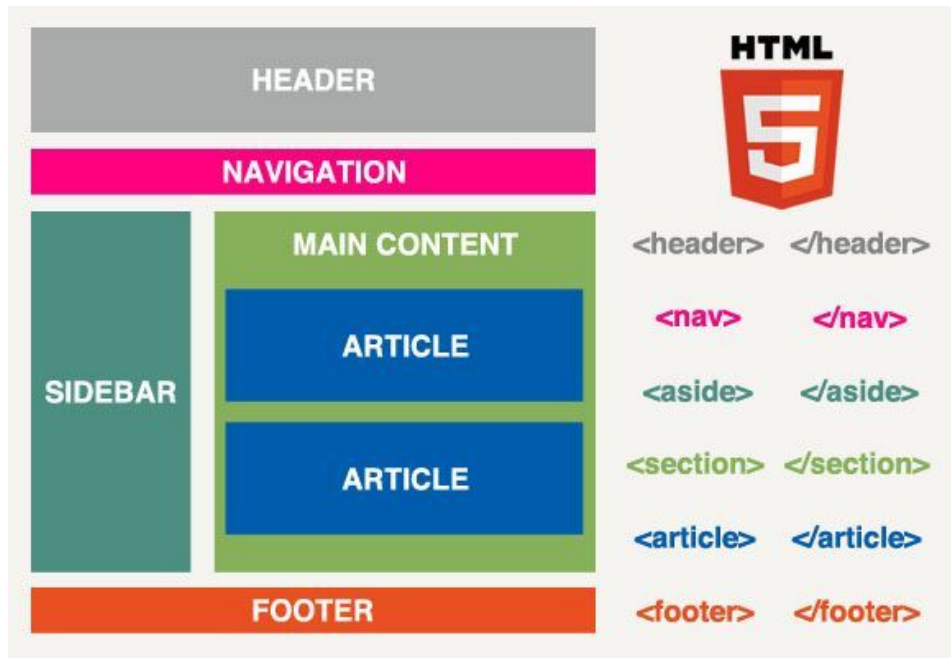
## Estructura

# Estructura de una Página Web (ejemplo)

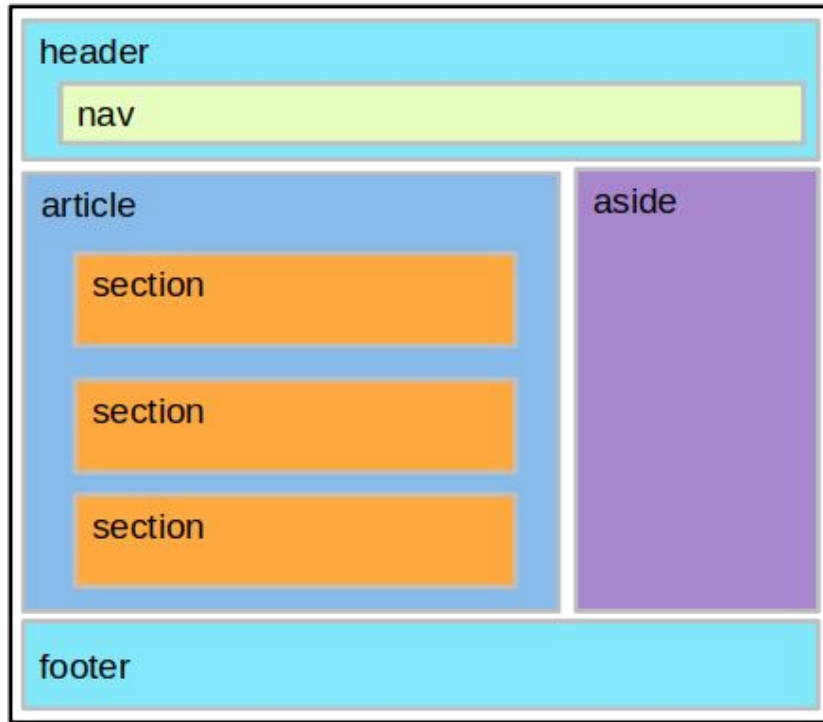
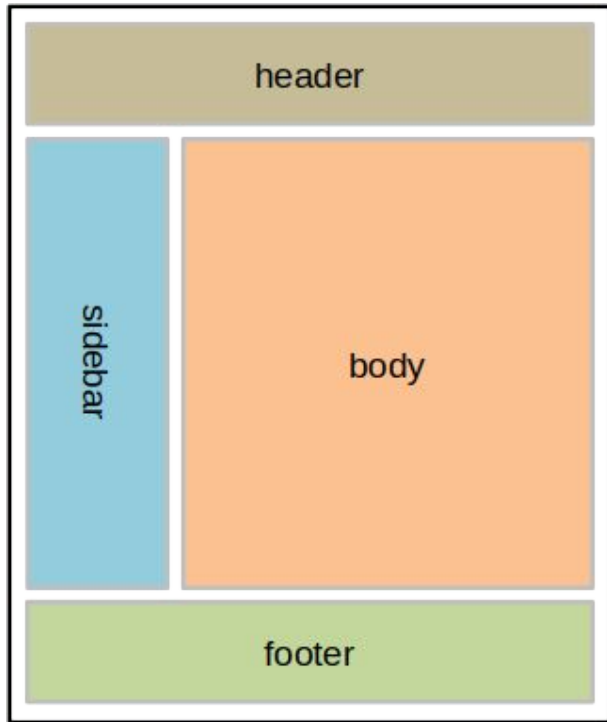
Estas etiquetas ayudan a definir la clase de contenido tendrá una página Web

Describen su significado tanto para el navegador como para el desarrollador. A través de ellas los navegadores y buscadores reconocen patrones y una estructura determinada.

Debemos respetarlas porque ayudan al navegador a entender su significado para mostrarlo en pantalla y ayudan a los buscadores a reconocer el contenido y la estructura del sitio.



# Estructura de una Página Web (ejemplo)



# Estructura de una Página Web (código)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

**Tip:** En VSC si escribimos `html:5` y presionamos TAB este código se escribirá automáticamente.

# Estructura de una Página Web (código)

**<!DOCTYPE html>** Indica que la versión corresponde a HTML5.

**<html lang= "es">** Es la etiqueta principal que engloba al resto de las etiquetas, el atributo *lang* define el tipo de lenguaje.

**<head>** Es la cabeza del documento que contiene los metadatos de la página web.

**<meta charset="utf-8"/>** Indica al navegador qué tipo de caracteres contiene la página, con el atributo *charset* vamos a indicar el conjunto de caracteres que vamos a usar y con el valor "utf-8" abarcamos a la mayoría de los sistemas de escritura.

**<title>** Indica el título de la página Web, que se visualiza en la barra de título del navegador.

**<body>** Es el cuerpo del documento donde va a estar todo el contenido que vamos a mostrar.

# Sintaxis

## Las etiquetas



# Las etiquetas

- Definen qué función cumple cada elemento dentro de la página.
- Estructura básica, tienen una apertura y un cierre que describen la información contenida entre ellas, aunque algunos casos solamente tienen apertura, como la etiqueta <br> (line break o salto de línea):

<b>Texto de ejemplo</b>    Texto de ejemplo

<p>Hola, estamos en el párrafo 1</p>  
<p>Ahora hemos cambiado de párrafo</p>

Hola, estamos en el párrafo 1

Ahora hemos cambiado de párrafo

<p>Este es un párrafo <br> que continúa  
en este renglón</p>

Este es un párrafo  
que continúa en este renglón

# Etiquetas básicas

**<h1>, <h2>, <h3>....<h6>:** Encabezados, numerados del 1 al 6 por orden de relevancia. Es importante respetar ese orden para que el navegador entienda la estructura de la página.

```
<h1>Encabezado en H1</h1>  
<h2>Encabezado en H2</h2>  
<h3>Encabezado en H3</h3>  
<h4>Encabezado en H4</h4>  
<h5>Encabezado en H5</h5>  
<h6>Encabezado en H6</h6>
```

## Encabezado en H1

### Encabezado en H2

#### Encabezado en H3

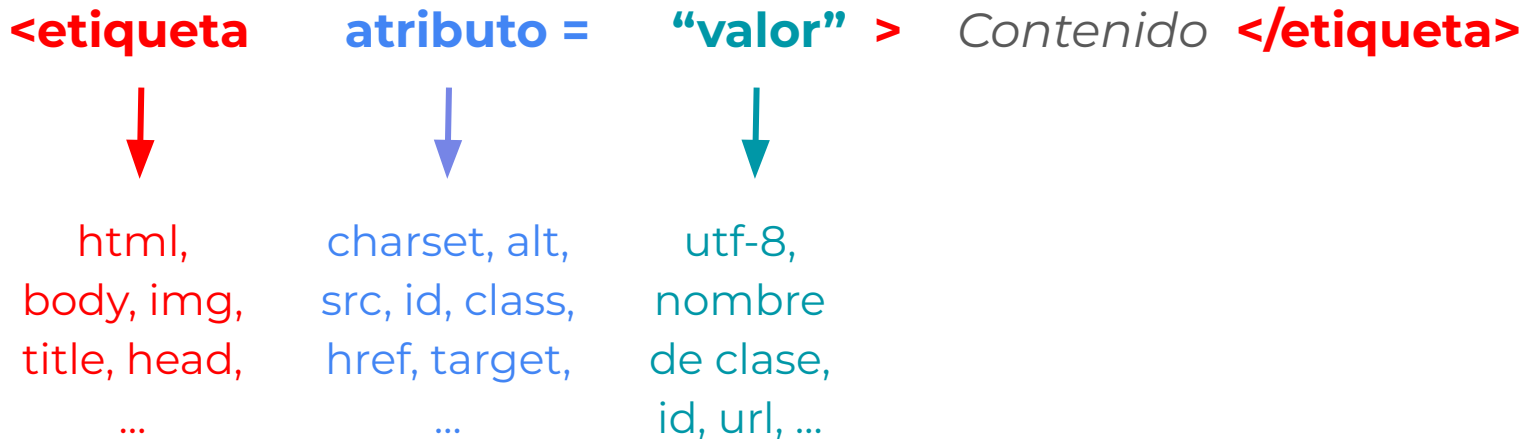
#### Encabezado en H4

#### Encabezado en H5

#### Encabezado en H6

**Tip:** En VSC si escribimos h1 y presionamos TAB la etiqueta de cierre se escribe automáticamente

# Las etiquetas



## Ejemplo:

```
<a href="https://www.google.com" target="_blank">Ir a Google</a>
```

# Etiquetas básicas

**<p>**: Representa un párrafo.

```
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit.  
Facilis labore eius nesciunt, voluptate modi quis tempora.  
Odit nisi voluptatem praesentium unde commodi optio,  
voluptate minima minus porro provident veritatis tempore  
voluptatum harum vero.</p>  
<p>Lorem ipsum dolor sit amet consectetur adipisicing  
elit.</p>
```

**Tip:** En VSC con si escribimos Lorem100 se escribirá un texto de ejemplo de 100 palabras. Y escribiendo p\*3 se crearán 3 etiquetas **<p>**.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Facilis labore eius nesciunt, voluptate modi quis tempora. Odit nisi voluptatem praesentium unde commodi optio, voluptate minima minus porro provident veritatis tempore voluptatum harum vero.

Lorem ipsum dolor sit amet consectetur adipisicing elit.

# Ejemplos de páginas Web

<https://www.w3schools.com/>

<https://instintobinario.com/category/hardware/>

<https://blog.ida.cl/estrategia-digital/diferencias-aplicacion-web-sitio-web/>

## Ejemplos de lo que **no** hay que hacer:

<http://arngren.net/>

Artículo: <https://grafix.es/las-webs-mas-horrorosas-del-mundo/>

## Cómo pensar la estructura de un Sitio Web:

<https://coach2coach.es/estructura-sitio-web/>

# Actividad Clase 1

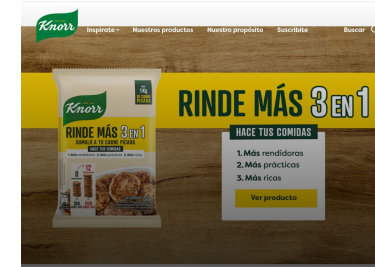
Analizar los siguientes sitios  
web



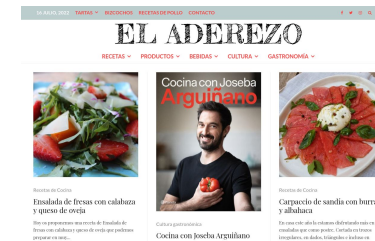
# Actividad:

Investigar los siguientes sitios web e identificar:

- Header (encabezado)
- Footer (pie)
- Nav (menú)
- Párrafos
- Imágenes
- Links
- Iconos
- etc.



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



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

# Tarea para el proyecto:

- Empezar a formar un grupo/equipo de 4 personas.
- Comenzar a pensar en el contenido del proyecto web.
- Recuerden que la temática es libre.
- Recomendamos visitar sitios web de similares temáticas.



# No te olvides de dar el presente

## **Recordá:**

- **Revisar la Cartelera de Novedades.**
- **Hacer tus consultas en el Foro.**

**Todo en el Aula Virtual.**

**Muchas gracias por tu atención.**

**Nos vemos pronto**