

***RECUERDA PONER A GRABAR LA
CLASE***





¿DUDAS DEL ON-BOARDING?

MIRALO AQUI



Clase 02. DESARROLLO WEB

PRIMEROS PASOS CON HTML



OBJETIVOS DE LA CLASE

- Comprender la sintaxis de HTML.
- Conocer las diferentes etiquetas y el uso de cada una.
- Dominar el concepto de web semántica, y las etiquetas HTML5 de estructura.

GLOSARIO:

Clase 1

Sketch: es un dibujo rápido o bosquejo guía, que reproduce de manera muy sencilla un concepto, una idea o generalidad de un proyecto.

Wireframe: es la representación estática, en baja calidad, de un diseño.

Mockup: es la representación estática de un diseño, en calidad media o alta.

Prototipo: es la representación navegable del producto final.

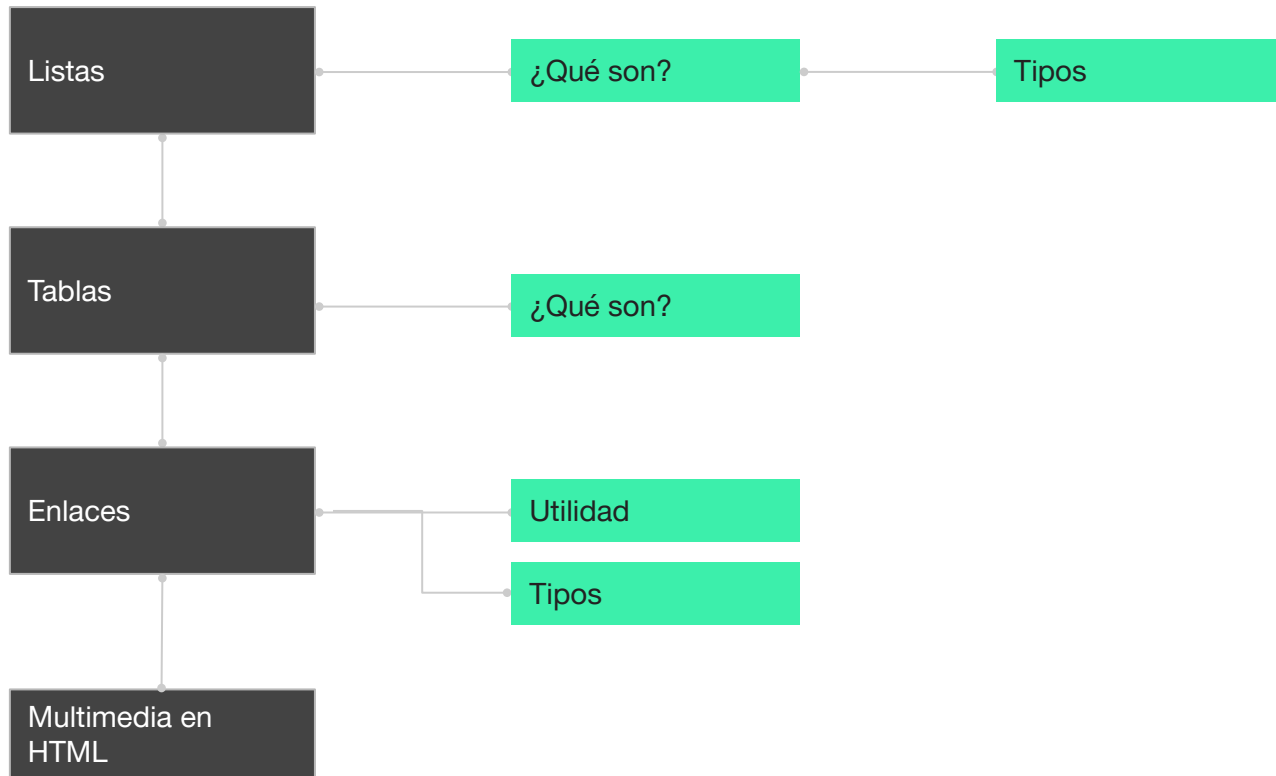
HTML: es un "lenguaje" de marcado de etiquetas, que permite crear documentos para web.

Doctype: 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.

MAPA DE CONCEPTOS

MAPA DE CONCEPTOS CLASE 2

¡Para
recordar!



CRONOGRAMA DEL CURSO

Clase 0



Inducción al curso de Desarrollo Web



CONCEPTOS BÁSICOS



HERRAMIENTAS A UTILIZAR
EN EL CURSO



EJEMPLO EN VIVO

Clase 1



Prototipado y conceptos básicos de HTML



SKETCH



PRÁCTICA DE LO VISTO
EN CLASE



NUEVO DOCUMENTO

Clase 2



Primeros pasos con HTML



LISTAS



FORMULARIOS



HTML



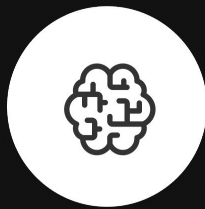
WIREFRAME Y ESTRUCTURA
DEL PROYECTO



GUIÓN DE LA CLASE

Accede al material complementario [aquí](#).





¡PARA PENSAR!

HTML no es un lenguaje de programación

¿VERDADERO O FALSO?
CONTESTA LA ENCUESTA DE ZOOM



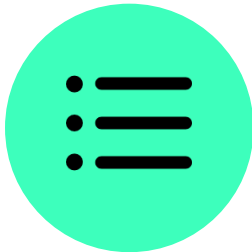
CODERHOUSE

LISTAS

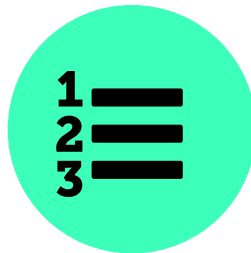
LISTAS

👉 **HTML** permite agrupar elementos que tienen más significado de forma conjunta. El menú de navegación de un sitio web, por ejemplo, está formado por un grupo de palabras. Aunque cada palabra por separado tiene sentido, de forma conjunta constituyen el menú de navegación de la página, por lo que su **significado conjunto** es mayor que por separado. Esto se denomina **listas**.

TIPOS DE LISTAS



Listas no ordenadas



Listas ordenadas



Listas de definición

Ejemplos: paso a paso de un procedimiento (como una receta de cocina), características de una persona, galería de imágenes, el menú de una página web, entre otros.

¿VIÑETAS O NÚMEROS?

- Las **listas numéricas** establecen un orden en la lectura de sus ítems.
- Las **listas de viñetas** no representan ningún orden o importancia entre sus ítems. Son elementos compuestos.



define una lista ordenada de artículos (numéricas).



define una lista de artículos sin orden (viñetas).



define un artículo de una lista.

VEAMOS UN EJEMPLO



Ambas listas se deben insertar mediante la etiqueta `` (list-item).

Ejemplo de servicios de una empresa (lista de viñetas/sin orden):

```
<ul>
  <li>Empresa</li>
  <li>Producto</li>
  <li>Servicios</li>
  <li>Contacto</li>
</ul>
```



- Empresa
- Producto
- Servicios
- Contacto

ANIDAR LISTAS



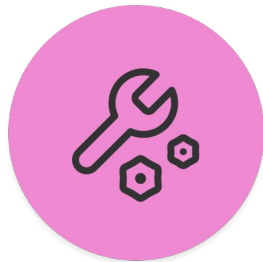
Es probable que te veas en la necesidad de crear una estructura de sublistas como la siguiente:

- Computadoras portátiles:
 - Procesador I4.
 - Procesador I5.
- Computadoras de escritorio:
 - Procesador Pentium.
 - Procesador Celeron.

Ejemplo
en vivo



¡VAMOS A PRACTICAR LO VISTO!



LISTAS

Crea dos listas usando las etiquetas de HTML.



¡A PRACTICAR!

Crea dos listas usando las etiquetas de HTML: una **con viñetas** que contenga cinco nombres, y otra **ordenada** con 5 pasos para preparar un mate. Cuentas con 15 minutos para completar la actividad.

TABLAS

TABLAS

👉 Son un **conjunto de celdas organizadas**, dentro de las cuales es posible alojar distintos contenidos. HTML dispone de una gran variedad de etiquetas y atributos para crear tablas. **Sirven para representar información tabulada, en filas y columnas.**

😊 En HTML4 las tablas se usaban para maquetar. Cuando CSS creció y se hizo más fuerte, nacieron los detractores de las tablas.

ETIQUETAS BÁSICAS PARA TABLAS EN HTML

Fila 1 - Columna 1	Fila 1 - Columna 2	Fila 1 - Columna 3	Fila 1 - Columna 4	Fila 1 - Columna 5
Fila 2 - Columna 1	Fila 2 - Columna 2	Fila 2 - Columna 3	Fila 2 - Columna 4	Fila 2 - Columna 5
Fila 3 - Columna 1	Fila 3 - Columna 2	Fila 3 - Columna 3	Fila 3 - Columna 4	Fila 3 - Columna 5

<table>

</table>



Acepta 3 atributos de “diseño”:

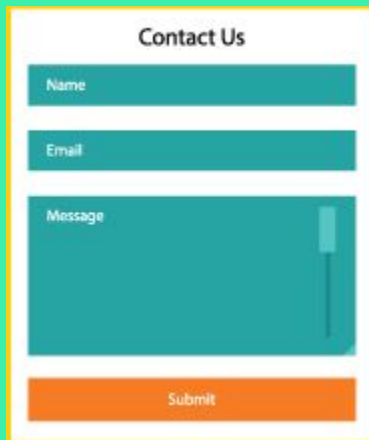
- **Border:** bordes de la tabla.
- **Cellpadding:** especifica el espacio, en píxeles, entre la pared de la celda y su contenido.
- **Cellspacing:** indica la distancia entre las celdas y el margen exterior de la tabla

```
<table>
<tr><!-- inicio de fila-->
  <td>Fila 1 - Columna 1</td>
  <td>Fila 1 - Columna 2</td>
  <td>Fila 1 - Columna 3</td>
</tr><!-- cierre de fila -->
<tr><!-- inicio de otra fila-->
  <td>Fila 2 - Columna 1</td>
  <td>Fila 2 - Columna 2</td>
  <td>Fila 2 - Columna 3</td>
</tr><!-- cierre de la segunda fila -->
</table>
```

FORMULARIOS

FORMULARIOS

👉 Son etiquetas donde **el usuario ingresará o seleccionará valores**, que serán enviados a un archivo encargado de procesar la información.



The image shows a contact form titled "Contact Us". It contains three input fields: "Name", "Email", and "Message". The "Name" and "Email" fields are single-line text inputs, while the "Message" field is a multi-line text area. Below these fields is a "Submit" button. The form is styled with a light blue header, light blue input fields, and a light blue submit button.

ETIQUETA **<FORM>**

Para insertar un formulario se usa la etiqueta **<form>**, que dentro lleva todos los controles que vayan al mismo destino. Un formulario requiere 3 atributos para funcionar:

- **Action:** documento que se encarga de recibir los datos y procesarlos.
- **Method:** la forma en que será enviada la información. Existen dos métodos de envío, que son GET y POST.
- **Enctype:** cómo se codificarán los contenidos.

ACTION

En este atributo se indicará **cuál es el archivo que recibe y procesa los datos**. Debe ser de un lenguaje de los llamados “del lado del servidor” (PHP / ASP / JSP). Si no se indica un valor, el **Action** será por defecto el mismo archivo donde está el formulario.

METHOD

Forma en la que se recopilan y envían los datos. Existen dos métodos comunes en el HTML:

- **GET:** la información viajará por la barra de direcciones a continuación del nombre del archivo.
- **POST:** la información viajará junto a los encabezados del HTML (será “invisible”).

Si el method no se indica, por defecto será GET.

ENCTYPE

Cuando el valor del atributo **method** es **post**, el mismo es el tipo MIME del contenido, que es usado para enviar el formulario al servidor. Los posibles valores son:

- **application/x-www-form-urlencoded**: será el valor por defecto si un atributo no está especificado.
- **multipart/form-data**: usar este valor si se está usando el elemento **input** con el atributo **type** ajustado a **"file"**.
- **text/plain** (HTML5)

Normalmente se utiliza para permitir el envío de archivos a través de un formulario.

INGRESO DE TEXTO

Existen tres controles generales para el ingreso de texto:

- Cajas de texto de una sola línea (no acepta el uso de la tecla Enter).
- Cajas para el ingreso de contraseñas (el contenido no será visible).
- Cajas para contenido multilínea. Puede ser una o muchas líneas de texto.

Atributo “name”

Control de formulario: `<input>`: Text, Email, Password.

Control de formulario: `<textarea></textarea>`

BOTONES

Los botones disparan las acciones del formulario. Hay 3 tipos:

- El que envía los datos al archivo indicado como **Action**.
- El que vacía todo lo ingresado y resetea los campos.
- El que “no hace nada”, pensado para usarse con Javascript.

Todos los botones son etiquetas **<input>**, con distintos tipos de “*Type*”. El botón debe de estar dentro del **<form>** que afectará.



ATRIBUTO “VALUE”

Representa la etiqueta del botón, la cual es normalmente mostrada por los navegadores dentro de éste.

- Input de tipo “*submit*”: envía el formulario.
- Input de tipo “*reset*”: resetea el formulario.
- Input de tipo “*button*”: no tiene acciones por defecto.

```
<form>
  <input type="submit" value="Enviar formulario"/>
  <input type="reset" value="Limpiar formulario"/>
  <input type="button" value="Sin acciones"/>
</form>
```


CONTROLES DE SELECCIÓN

En estos casos, el usuario no puede ingresar libremente un texto, sino que el programador le da una lista predefinida. El dato que llega al elegir una opción se define desde el atributo *“value”*. Existen 3 grupos de controles de selección:

- **Botones de radio:** sólo se puede elegir una opción.
- **Casillas de chequeo:** de toda la lista de opciones, el usuario puede optar por una, todas o ninguna opción.
- **Menú desplegable:** sólo es posible seleccionar una opción.

ATRIBUTO “VALUE”



En este caso es el valor que se enviará al enviarse el formulario.

Botones de radio:

```
<form>
  <div>hombre</div>
  <input type="radio" name="sexo" value="hombre" />
  <div>mujer</div>
  <input type="radio" name="sexo" value="mujer" />
</form>
```

Casillas de chequeo:

```
<form>
  <div>Acepta términos y condiciones</div>
  <input type="checkbox" name="acepta" value="1" />
</form>
```



ETIQUETA <LABEL>

Define formalmente a cada elemento de un formulario. Esta etiqueta es de mucha ayuda para generar un formulario accesible.

Su principal atributo es “for”, que va a referenciar a “label” con su elemento del formulario. El valor del atributo “for” debe ser igual al valor del atributo “id” o “name” del elemento.

```
<form>
  <label for="nombre_apellido">Nombre:</label>
  <input type="text" name="nombre_apellido" />
</form>
```

MENÚ DESPLEGABLE



Es el llamado **combo-box, selector o menú**. De toda la lista, se puede elegir una opción (aunque tiene un atributo que permite cambiarlo). Lo ideal es que sean al menos dos elementos distintos para observar el select:

```
<form>
  <select name="talles">
    <option value="L">Large</option>
    <option value="M">Medium</option>
    <option value="S">Small</option>
  </select>
</form>
```



CONJUNTO DE CAMPO

Las etiquetas `<fieldset>` y `<legend>` se utilizan en conjunto. La primera, tiene como objetivo crear grupos de elementos del formulario que posean un mismo propósito; mientras que la segunda, define formalmente el propósito del elemento `fieldset`. Se estructuran de la siguiente manera:

```
<form>
  <fieldset>
    <legend>Talle de remera</legend>
    <!-- Aquí irán los elementos de formulario -->
  </fieldset>
</form>
```

Ejemplo
en vivo



¡VAMOS A PRACTICAR LO VISTO!



FORMULARIOS

Crea un formulario de contacto.



¡A PRACTICAR!

Crea un formulario de contacto como indica la imagen a continuación, usando los códigos vistos anteriormente. Tienes 15 minutos para realizar la actividad.

Información Personal:

Sexo:

☐ hombre ☐ mujer

Nombre:

Apellido:

Talle:

Enviar



BREAK

¡5/10 MINUTOS Y VOLVEMOS!



PILAR
MUNICIPIO

conectados
con las *oportunidades*

CODERHOUSE

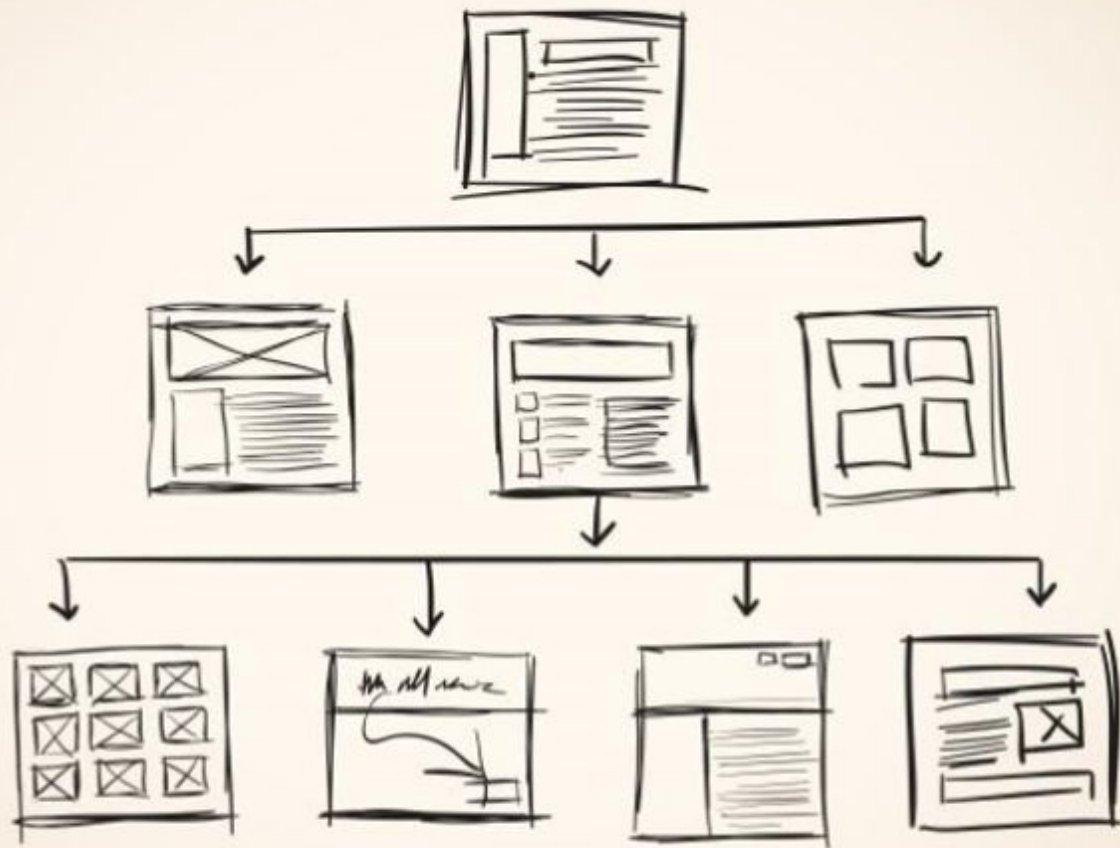
ENLACES

ARQUITECTURA DE UN SITIO

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. Por ejemplo:

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



ENLACES RELATIVOS, ABSOLUTOS E INTERNOS



ENLACES RELATIVOS

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.

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

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

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



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="http://www.coderhouse.com/frontend">Curso de Frontend</a>
```



ENLACES INTERNOS

Los **enlaces internos** permiten referenciar secciones de tu página, para lo cual se utiliza el **id**:

```
<a href="#pie">Ir al pie de página</a>  
...  
<footer id="pie"></footer>
```


También puedes usar como destino una sección específica una página distinta:

```
<a href="contacto.html#formulario">Formulario de contacto</a>
```

En el ejemplo anterior, el enlace apunta a la sección que tiene el id formulario dentro de la página “contacto.html”. No sólo es posible agregar enlaces a texto, también puedes hacerlo con otros elementos. Por lo general, se usan textos o imágenes. Veamos un ejemplo de enlaces con una imagen:

```
<a href="http://www.coderhouse.com/cursos.html#frontend">  
    
</a>
```

MULTIMEDIA EN HTML

IMÁGENES

👉 **Enriquecen el HTML:** las imágenes son elementos que, bien utilizados, mejoran la experiencia de los usuarios.

Unite a la comunidad de aprendizaje online en vivo más grande de Latinoamérica

Clases online en vivo dictadas por expertos de la industria, enfoque 100% práctico, mentorías personalizadas y acceso a una comunidad de +50,000 estudiantes.

[Ver todos los cursos](#)

Calidad 100% asegurada

A screenshot of a live class interface on the Coderhouse platform. The interface is dark-themed. At the top, it says 'CODERHOUSE' and 'Online en vivo'. Below that, it says 'Clases en vivo' and 'Diseño UX/UI'. The main area shows a video feed of a woman speaking, with three smaller video feeds of other participants above it. On the right side, there is a list of participants with their names and avatars. The interface is framed by a yellow border.

INSERTAR IMÁGENES

📌 Se insertan con la etiqueta ``, que pertenece al grupo de las etiquetas que se cierran a sí mismas (con la barra al final).

📍 Para funcionar requiere, como mínimo, indicar en dónde está el archivo a mostrar. Eso se hace con el atributo “src” (el source o fuente), que respeta todas **las reglas de ruteo** vistas en los links.

👥 Se comportan como **elementos de línea**, esto significa que se verán una al lado de la otra.



ALT

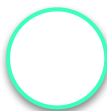


El “alt” es un texto que debe representar la foto que se está visualizando. Tiene que ser conciso y breve, pero dejar en claro de qué se trata la imagen.

```

```

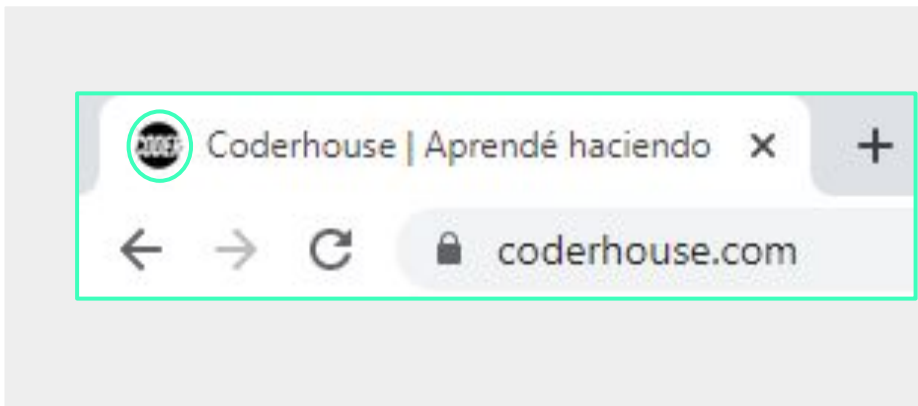
FAVICON



El favicon es un pequeño ícono que **identifica un sitio web** cuando está abierto en una pestaña o cuando es guardado como favorito.

HTML recomendado para agregar dentro de tu etiqueta <head>:

```
<link rel="shortcut icon" href="favicon.png">
```





ETIQUETA IFRAME



Es un elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML principal.

```
<iframe src="pagina_fuente.html" width=290  
height=250>Texto para cuando el navegador no  
conoce la etiqueta iframe</iframe>
```

ETIQUETA IFRAME



Insertar vídeo

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/
P03IJBtrFyc" title="YouTube video
player" frameborder="0"
allow="accelerometer; autoplay;
clipboard-write; encrypted-media;
gyroscope; picture-in-picture"
allowfullscreen></iframe>
```

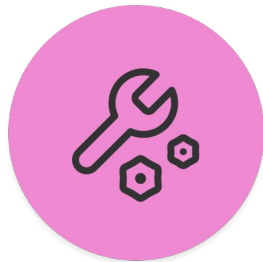
☐ Empezar en 0:12

COPIAR

Ejemplo
en vivo



¡VAMOS A PRACTICAR LO VISTO!



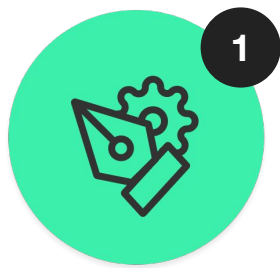
HTML

Crea un archivo HTML.



¡A PRACTICAR!

Crea un archivo HTML que contenga: un video, un audio (puedes descargar ambos de Drive), y un iframe que muestre el video de [Coderhouse](#). Tienes 15 minutos para realizar la actividad.



WIREFRAME Y ESTRUCTURA DEL PROYECTO

Genera el **wireframe de una vista para mobile** y **una vista para desktop** teniendo en cuenta el tema elegido.

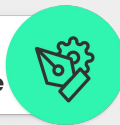
Luego, crea **los archivos html** de las secciones y utiliza las **etiquetas** vistas hasta ahora en clase.

WIREFRAME Y ESTRUCTURA DEL PROYECTO

Formato: pdf o jpg exportado de balsamiq + **carpeta root** con organizada en [Sublime Text](#)/[Visual Studio Code](#) Debe tener el nombre “Idea+Apellido”

Sugerencia: entregar en **un archivo comprimido** todo junto.

Desafío
entregable



>> Consigna: Genera el wireframe de una vista para mobile y una vista para desktop teniendo en cuenta el tema elegido. Luego, crea los archivos html de las secciones y utiliza las etiquetas semánticas vistas hasta ahora en clase.

WIREFRAME Y ESTRUCTURA DEL PROYECTO

>>Aspectos a incluir en el entregable:

1. **Wireframe** correspondiente a una vista para desktop y una vista mobile en balsamiq.
2. **5 archivos HTML** para cada sección del sitio web, aplica entre estos los enlaces relativos y escribe con etiquetas semánticas. Deberás incluir:
 - Una lista no ordenada con el menú principal del sitio web con los nombres de cada una de las secciones y replícalo en las mismas.
 - Un formulario en la página de contacto.
 - Un footer y replícalo en todos tus html.
 - Etiquetas multimedia como img e iframes.

>>**Ejemplo:** Podrás encontrar un ejemplo en la carpeta de clase

¿PREGUNTAS?





¡MUCHAS GRACIAS!

Resumen de lo visto en clase hoy:

- Sintaxis de HTML.
 - Diferentes etiquetas y el uso de cada una.
 - Acercamiento al concepto de web semántica, y las etiquetas HTML5 de estructura.
- 