



Emprendimiento digital con tecnologías Web

Módulo 5: Introducción al desarrollo web para emprendimiento digital

Aprendizaje Esperado

1. Estructurar contenido en una página web utilizando HTML5 para dar solución a una problemática.

1.- Introducción a HTML

1.1.- El editor de código: Descarga del editor Visual Studio Code e Instalación

Visual Studio Code es un editor de código fuente creado por la empresa Microsoft, desde una perspectiva multiplataforma. Actualmente se encuentra disponible para los sistemas operativos Windows, Linux y macOS [1].

Esta aplicación además permite ejecutar operaciones de desarrollo, tales como la depuración de código, ejecución de tareas y control de versiones. Su objetivo es proporcionar solo aquellas herramientas que un desarrollador necesita para un ciclo eficiente de creación de código y depuración. Los flujos más complejos, en tanto, se dejan en manos de entornos de desarrollo integrados (IDE) más complejos.

Una característica interesante de este software, es que su uso es liberado, tanto para uso privado o comercial. Para descargarlo, solo deben acceder al enlace incluido en las referencias.

Utilizar el potencial de un editor de texto para el desarrollo.

En términos generales, un editor de texto es un tipo de software diseñado para crear, modificar y almacenar archivos de texto. Se diferencia de los procesadores de texto tradicionales por la ausencia de opciones de formato avanzadas y de la posibilidad de mostrar elementos gráficos o multimedia. Casi todos los sistemas operativos masivos cuentan con un editor de texto, además de los editores de texto creados por terceros.

Entre las ventajas que considera una herramienta de esta naturaleza, se destacan las siguientes:

- **Multiplataforma:** cada sistema operativo cuenta con un conjunto de programas de escritorio que permiten editar archivos de texto, algunos incorporados y otros creados por entidades externas. Independiente de esta diversidad, situaciones tales como el manejo de saltos de líneas se encuentra estandarizado, mejorando la visualización y manipulación.
- **Integridad de los datos:** un editor de texto por lo general no agrega caracteres extraños provenientes de formatos o elementos gráficos, permitiendo generar archivos de texto plano. Para codificar sitios web, es fundamental que las etiquetas incorporadas sean incluidas como texto plano, ya que cualquier formato adicional puede alterar la integridad del archivo, generando errores en su visualización en un navegador.
- **Tamaño de archivos:** otra ventaja de los editores de texto, es la posibilidad de manipular archivos de gran tamaño, situación que no se aplica a procesadores de texto habituales. Esto incluye tanto la edición de los documentos como la búsqueda de términos.

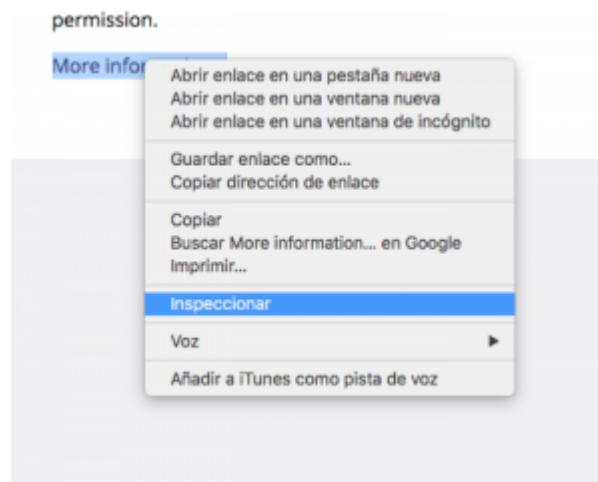
1.3.- Conociendo el inspector de elementos en un navegador

Un navegador web es una aplicación de software que comunica un dispositivo cliente (PC, notebook, celular, Tablet, etc.) con un servidor web, al cual se le solicita parte de la información que tiene almacenada. Si lo analizamos en términos simples, una página web no es más que un conjunto de archivos de texto plano y documentos multimedia que son interpretados por un navegador, brindándole forma.

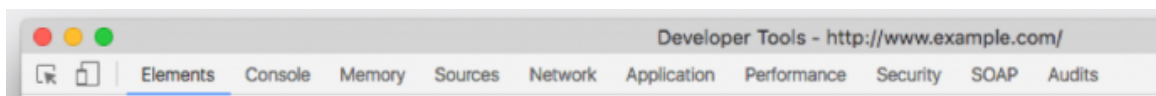
Los navegadores más usados en la actualidad son Microsoft Edge, Google Chrome, Mozilla Firefox y Opera [2]. Si bien todos ellos cumplen un objetivo específico y transversal, presentan características que los diferencian de sus pares, y que traen como consecuencia que los usuarios tengan un navegador de preferencia.

Una herramienta que es transversal en los navegadores conocidos es el “inspector de elementos”, la cual es permite analizar los contenidos de cualquier sitio web. A través de esta aplicación se puede ver el código fuente de la página, así como analizar cada uno de los componentes que la conforman. Es de mucha utilidad cuando se quiere ver cómo está hecho algo en un sitio, o bien para afinar estilos y depurar scripts que se ejecuten.

Cada inspector de los navegadores tiene diferencias menores, pero las acciones básicas a desarrollar son prácticamente las mismas. En las imágenes siguientes se usará como referencia el navegador Google Chrome. Para acceder al inspector de elementos debe seleccionar el objeto a analizar, y con el botón derecho del ratón seleccionar la opción “Inspeccionar”.



El inspector se compone de varias pestañas, donde cada una determina distintas funcionalidades y características:



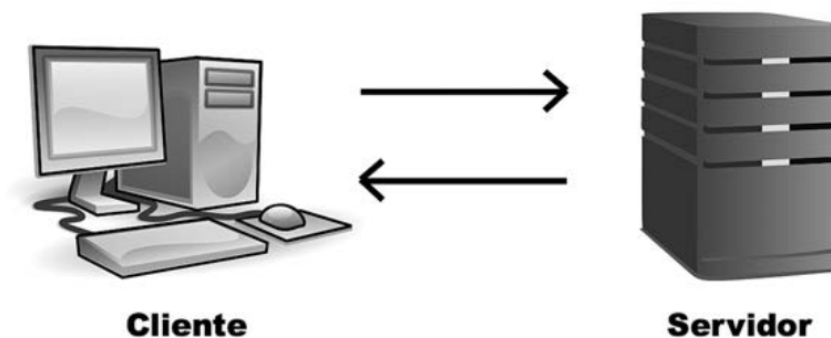
- **Elements:** se puede revisar el código fuente del sitio, junto a los estilos que se aplican sobre ella. Se pueden modificar localmente y añadir nuevas reglas.
- **Console:** en esta sección se muestran los mensajes de error y de advertencias que se producen en la página (imágenes que no cargan correctamente, errores de Javascript, entre otros).
- **Sources:** recursos de la página, expresados en forma de árbol. Se puede analizar su origen y modificarlos de manera local.
- **Network:** esta ventana permite analizar cada una de las peticiones que se hacen a un servidor web, considerando en ello el contenido de la petición, la respuesta que se obtuvo y los tiempos asociados.
- **Performance:** herramientas que permiten medir el desempeño de un sitio.
- **Memory:** mide la memoria que se consume durante el proceso de carga del sitio y durante la ejecución de las tareas respectivas.
- **Application:** datos de utilidad sobre los recursos que emplea una aplicación web.

- **Security:** información sobre los sitios que están enlazados al portal analizado, y el estado de su certificado de seguridad respectivo.
- **Audits:** contiene una serie de controles predeterminados que permiten validar errores típicos.

2. El Lenguaje HTML

2.1.- Introducción al lenguaje de etiquetas de hipertexto

Los sitios web son archivos que los usuarios descargan con sus navegadores desde equipos remotos. Cuando un usuario accede a una página, le indica al navegador la dirección del mismo y descarga los archivos, procesa el contenido y lo despliega. Dado que los archivos deben estar disponibles todo el tiempo, no pueden ser almacenados en equipos convencionales sino que deben estar dispuestos en equipos especializados. El equipo que hace la consulta se denomina “cliente”, mientras que el que la procesa y envía una respuesta es el “servidor” [3].



Un sitio web es la suma de documentos y recursos multimedia que el navegador descarga cuando el usuario los solicita. Los documentos que forman un sitio web se llaman “páginas” y el proceso de abrir nuevas páginas se conoce como “navegar”.

Si se desea crear un sitio web, es necesario crear un archivo por cada página que se incluirá. Junto con estos archivos, se deben incluir los archivos con las imágenes y cualquier otro recurso que se desplegará en estas páginas; considera que las imágenes y otros medios gráficos se almacenan en archivos aparte.

Los archivos de un sitio web son similares a los documentos que podemos encontrar en un computador común. Cada uno de ellos tiene un nombre asignado por el desarrollador y una extensión que determina el lenguaje utilizado para crear su contenido. Aunque es posible asignar cualquier nombre a estos archivos, el documento que genera la página inicial presenta algunos requisitos, como por ejemplo que el sitio inicial tenga como nombre "index.html"; sin embargo, esta última condición es configurable de acuerdo a las configuraciones aplicadas sobre el software que actúa como servidor web.

Cuando un sitio ha sido creado, todos los archivos que lo componen son almacenados en un servidor web. Cada servidor se identifica con un valor llamado IP (Internet Protocol). Esta IP es única para cada equipo y, por lo tanto, trabaja como una dirección que permite ubicarlo dentro de una red. Cuando el browser web tiene que acceder al servidor para descargar el documento solicitado por el usuario, primero busca el servidor a través de esta dirección IP y luego le pide que le envíe el documento.

En teoría, se puede acceder a cualquier servidor utilizando su dirección IP, pero estos valores son difíciles de recordar. Por lo mismo, Internet usa un sistema que identifica a cada servidor con un nombre específico. Estos nombres personalizados, llamados dominios, son identificadores sencillos que cualquier persona puede recordar, como google o yahoo, con una extensión que determina el propósito del sitio web al que hacen referencia, como .com (comercial) o .org (organización).

2.2.- Definición de HTML. Qué es y para que se usa.

La sigla HTML es una abreviatura que viene del inglés "HyperText Markup Language", y es un lenguaje compuesto por un conjunto de etiquetas definidas con un nombre rodeado de paréntesis "<" y ">". Los paréntesis angulares delimitan la etiqueta, y el nombre define el tipo de contenido que representa. Por ejemplo, la etiqueta <html> indica que el contenido es código HTML. Algunas de estas etiquetas son declaradas individualmente (por ejemplo,
) y otras son declaradas en pares, considerando una de apertura y otra de cierre, como <html></html> (en la etiqueta de cierre el nombre va precedido por una barra invertida). Las etiquetas individuales y las de apertura pueden incluir atributos para ofrecer información adicional acerca de sus contenidos (por ejemplo, <html lang="es">). Las etiquetas individuales y la combinación de etiquetas de apertura y cierre se llaman elementos. Los elementos compuestos por una sola etiqueta se usan para modificar el contenido que los rodea o incluir recursos externos, mientras que los

elementos que incluyen etiquetas de apertura y cierre se utilizan para delimitar el contenido del documento.

2.3 Creando un archivo HTML

Es necesario incluir varios elementos para definir un documento. Los elementos son listados en secuencia de descendente, y pueden contener otros elementos en su interior. Por ejemplo, la se dijo que la etiqueta `<html></html>` indica el inicio y cierre de cualquier sitio; por lo tanto, el resto de los elementos que describen el contenido de ese documento se deben declarar entre las etiquetas `<html>` y `</html>`. A su vez, las etiquetas dentro del elemento `<html>` pueden incluir otras más. El ejemplo indicado a continuación muestra un documento HTML sencillo que incluye todos los elementos necesarios para definir una estructura básica y mostrar el mensaje "HOLA MUNDO!" en un navegador.

```
1  <!DOCTYPE html>
2  <html lang="es">
3    <head>
4      <title>Mi primer documento HTML</title>
5    </head>
6    <body>
7      <p>HOLA MUNDO!</p>
8    </body>
9  </html>
```

En el ejemplo anterior se muestra un código que podría resultar sencillo en principio, pero que posee una estructura compleja. En la primera línea se encuentra una etiqueta que señala el tipo de documento existente en las líneas posteriores (`<!DOCTYPE html>`), seguida por una etiqueta de apertura `<html lang="es">`. Entre las etiquetas `<html>` y `</html>` se incluyen otros elementos que representan la cabecera y el cuerpo del documento (`<head>` y `<body>`), los cuales a su vez encierran más elementos con sus respectivos contenidos (`<title>` y `<p>`), demostrando cómo se compone un documento HTML. Los elementos se listan uno a continuación de otro y también dentro de otros elementos, de modo que se construye una estructura de tipo árbol con el elemento `<html>` como raíz.

2.3.- Conceptos básicos asociados a un documento HTML.

Como parte del estudio del HTML, existen conceptos claves que es necesario conocer. Por lo mismo, se irán abordando en detalle en las clases siguientes; por lo pronto, se dispone una lista de conceptos y su definición.

HTML5

Es un lenguaje de etiquetas usado para presentar y estructurar el contenido en un sitio web. Es la quinta revisión del estándar HTML, el cual fue creado en 1990. La W3C (World Wide Web Consortium), entidad que determina las bases del lenguaje, recomendó su utilización para transformarse en el estándar a ser usado en proyectos venideros. Está relacionado también con la entrada en decadencia del anterior estándar HTML 4, que se combinaba con otros lenguajes para producir los sitios que se pueden ver en la actualidad. Con HTML5 existen otras posibilidades para explotar sitios usando menos recursos. Con HTML5 entra conjuntamente en desuso el formato XHTML, dado que ya no sería necesaria su implementación.

Se trata de un sistema para formatear el “layout” o disposición de nuestras páginas, así como hacer algunos ajustes a su aspecto. Con HTML5 los navegadores como Firefox, Chrome, Explorer, Safari y más pueden saber cómo mostrar una determinada página web, saber dónde están los elementos, dónde poner las imágenes, dónde ubicar el texto. En este sentido, el HTML5 no se diferencia demasiado de su predecesor. La diferencia principal, sin embargo, es el nivel de sofisticación del código que podremos construir usando HTML5. Las etiquetas que se indicarán más adelante son una combinación del estándar original con HTML5.

Hipervínculos

También llamados “enlaces”, son referencias a documentos dentro de las páginas de un sitio web. Incorporando estos enlaces, una página puede contener referencias a otras páginas. Si el usuario hace clic con el ratón en un enlace, el navegador sigue esa referencia y el documento indicado por la URL de la referencia se descarga y muestra en pantalla. Debido a estas conexiones entre páginas, los usuarios pueden navegar en el sitio web y acceder a todos sus documentos simplemente haciendo clic en sus enlaces. Los enlaces son lo que transforma a un grupo de archivos en un sitio web. Para crear un sitio web, debe programar los documentos correspondientes a cada página e incluir dentro de las mismas los enlaces que establecen una ruta que el usuario puede seguir para acceder a cada una de ellas.

URL absolutas y relativas

Las URL absolutas son aquellas que incluyen toda la información necesaria para acceder al recurso (ver Figura 1-3), mientras que las relativas son aquellas que solo declaran la parte de la ruta que el navegador tiene que agregar a la URL actual para acceder al recurso. Por ejemplo, si tenemos un hipervínculo dentro de un documento que referencia una imagen dentro del directorio "imagenes", podemos crear el enlace con la URL `http://www.ejemplo.com/imagenes/miimagen.png`, pero también tenemos la opción de declararla como "imagenes/miimagen.png" y el navegador se encargará de agregar a esta ruta la URL actual y descargar la imagen.

Las URL relativas no solo pueden determinar una ruta hacia abajo, sino también hacia arriba de la jerarquía. Por ejemplo, si tenemos un documento dentro del directorio recursos en el ejemplo de la Figura 1-2 y queremos acceder a un documento en el directorio raíz, podemos crear una URL relativa usando los caracteres `../` al comienzo de la ruta. Si el documento que queremos acceder es `noticias.html`, la URL relativa sería `../noticias.html`. Los dos puntos `..` le indican al navegador que el documento al que queremos acceder se encuentra dentro del directorio padre del actual directorio (recursos, en nuestro ejemplo).

CSS - Hojas de estilo en cascada

CSS (Cascading Style Sheets) es el lenguaje que se utiliza para definir los estilos de los elementos HTML, como el tamaño, el color, el fondo, el borde, etc. Aunque todos los navegadores asignan estilos por defecto a la mayoría de los elementos, estos estilos generalmente están lejos de lo que queremos para nuestros sitios web. Para declarar estilos personalizados, CSS utiliza propiedades y valores. Esta construcción se llama

declaración y su sintaxis incluye dos puntos después del nombre de la propiedad, y un punto y coma al final para cerrar la línea.

```
1  body {
2      width: 100%;
3      margin: 0px;
4      background-color: ■ #FF0000;
5  }
6
```

Las propiedades CSS se pueden agrupar usando llaves. Un grupo de una o más propiedades se llama regla y se identifica por un nombre llamado selector. En el ejemplo anterior, el valor “#FF0000” se asigna a la propiedad “background-color”. Si esta propiedad se aplica luego a un elemento HTML, el contenido de ese elemento se mostrará en color rojo (el valor #FF0000 representa el color rojo). En el mismo ejemplo se declara también una regla con tres propiedades: width, margin y background-color. Esta regla se identifica con el nombre body, lo que significa que las propiedades serán aplicadas al elemento <body>. Si incluimos esta regla en un documento, el contenido del documento se extenderán hacia los límites de la ventana del navegador y tendrán un fondo rojo.

JavaScript

A diferencia de HTML y CSS, JavaScript es un lenguaje de programación. Para ser justos, todos estos lenguajes pueden ser considerados lenguajes de programación, pero en la práctica existen algunas diferencias en la forma en la que entregan las instrucciones al navegador. HTML es como un grupo de indicadores que el navegador interpreta para organizar la información, CSS puede ser considerado como una lista de estilos que ayudan al navegador a preparar el documento para ser presentado en pantalla (aunque la última especificación lo convirtió en un lenguaje más dinámico), pero JavaScript es un lenguaje de programación, comparable con cualquier otro lenguaje de programación profesional como C++ o Java. JavaScript difiere de los demás lenguajes en que puede realizar tareas personalizadas, desde almacenar valores hasta calcular algoritmos complejos, incluida la capacidad de interactuar con los elementos del documento y procesar su contenido dinámicamente.

```
1  <script>
2      function cambiarColor() {
3          document.body.style.backgroundColor = "#0000FF";
4      }
5      document.addEventListener("click", cambiarColor);
6  </script>
```

Al igual que HTML y CSS, JavaScript se incluye en los navegadores y, por lo tanto, se encuentra disponible para todos los documentos creados. Para declarar código JavaScript dentro de un documento, HTML ofrece el elemento <script>. El siguiente ejemplo es una muestra de un código escrito en JavaScript; el código del ejemplo cambia el color de fondo del elemento <body> a azul cuando el usuario hace clic en el documento.

Lenguajes de servidor

Los códigos programados en HTML, CSS, y JavaScript son ejecutados por el navegador en el computador del usuario (el cliente). Esto significa que, después de que los archivos del sitio web se suben al servidor, permanecen inalterables hasta que se descargan en un equipo personal y sus códigos son ejecutados por el navegador o browser. Aunque esto permite la creación de sitios web útiles e interactivos, hay momentos en los cuales necesitamos procesar la información en el servidor antes de enviarla al usuario. El contenido producido por esta información se denomina contenido dinámico, y es generado por códigos ejecutados en el servidor y programados en lenguajes que fueron especialmente diseñados con este propósito (lenguajes de servidor). Cuando el navegador solicita un archivo que contiene este tipo de código, el servidor lo ejecuta y luego envía el resultado como respuesta al usuario. Estos códigos no solo se utilizan para generar contenido y documentos en tiempo real, sino también para procesar la información enviada por el navegador, almacenar datos del usuario en el servidor, controlar cuentas, etc.

Existen varios lenguajes disponibles para crear código ejecutable en los servidores. Los más populares son PHP, Ruby, y Python. En este curso haremos uso del lenguaje Java para programación web, bajo la lógica que promueven las "JavaServer Pages" o JSP.

3. Landing Page v/s Sitios web



Veamos entonces las diferencias entre una Landing Page y una Página Web.

Una **página web** es el centro o la base de toda la estrategia digital, pues es de la pertenencia del cliente y no depende de ninguna red social o google para mantenerse en pie.

Al mismo tiempo, en la página web puedes mantener contenido propio, un eCommerce, blog, actualizaciones y más, que generan un universo muy completo.

Las **landing page**, también llamadas páginas de aterrizaje, son páginas dentro de un sitio web, desarrollada con el único objetivo de convertir los visitantes en Leads o prospectos de ventas por medio de una oferta determinada. Generalmente tiene un diseño más sencillo con pocos enlaces e informaciones básicas sobre la oferta, además de un formulario para realizar la conversión.

Los usuarios que llegan a una página web, probablemente ya nos conocen, es más que posible que ya hayan puesto nuestro nombre o nuestra dirección en los buscadores.

Por otra parte, los usuarios que llegan a una landing page, seguramente, no nos conocen. Han llegado hasta nosotros, por decirlo así, de rebote. Seguramente encontraron en internet algo que les ha llamado la atención y que les ha llevado hasta nosotros, pero necesitan que nos presentemos.

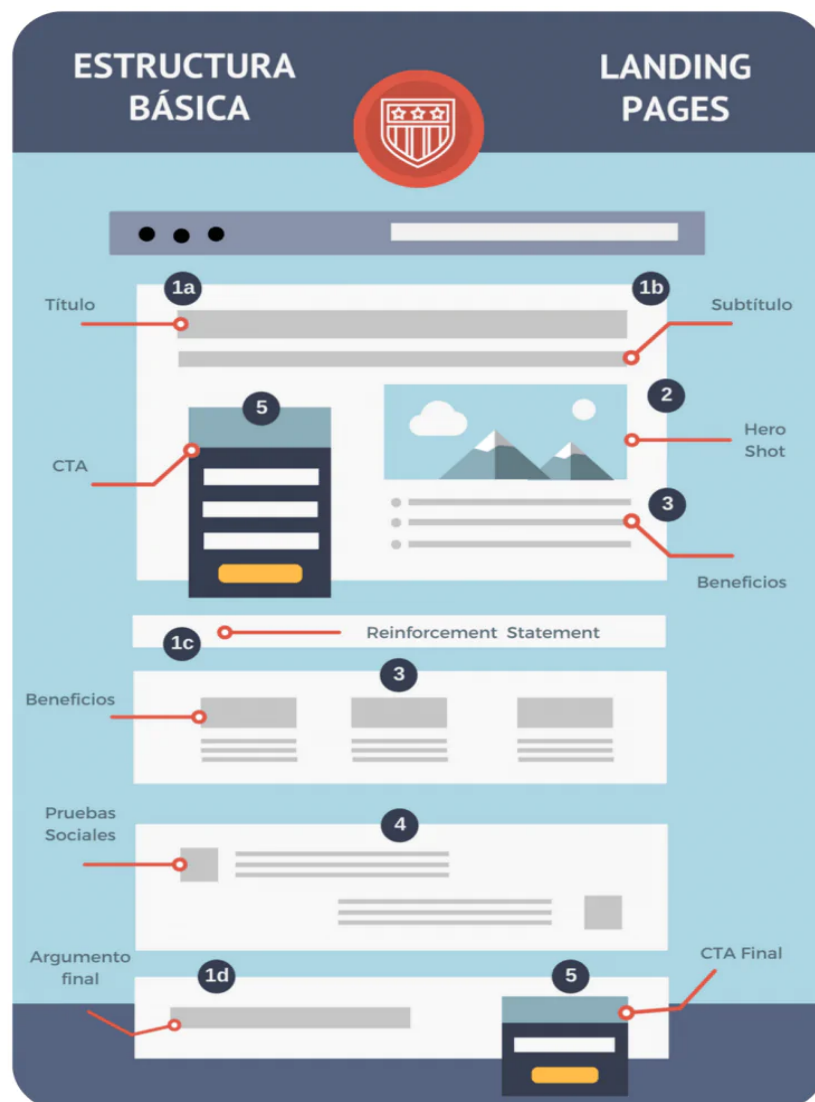
3.1 Estructura de una Landing Page

Entonces, ¿Para qué sirve una landing page?

Una landing page crea una experiencia única para un cliente y un producto concreto. Mantiene al visitante enfocado en la propuesta de valor del producto y su objetivo es maximizar la conversión de dicha página (conversión = % de clientes que compran respecto a los que visitan la página de producto).

¿Qué características hacen una landing page eficiente?

No hay una fórmula perfecta para elaborar una Landing Page. Pero podemos reconocer algunos elementos que se deben tener en cuenta.



Fuente: iebschool.com

Título de la Página (1a)

El título de la página es lo primero que va a ver el visitante de la web. Tiene que ser corto, conciso y, sobre todo, prometedor. Debe ser capaz de comunicar el problema que resolverá el producto al visitante. En la siguiente imagen sería el texto marcado en amarillo.

Subtítulo (1b)

En el subtítulo haremos una descripción un poco más larga hablando de cómo funciona nuestro producto para solucionar el problema del usuario. En la siguiente imagen es el texto marcado en azul.

Refuerzo (1c)

El refuerzo será un texto o copy corto que iremos posicionando más adelante en la página para recordar el beneficio de uso de nuestro producto. Hay landing pages que pueden ser extremadamente largas (como las páginas de infoproductos o cursos) y a veces es necesario hacer un recordatorio.

Hero Shot / Hero Image (2)

Seguro que te has dado cuenta de que muchas landing pages de venta que has visitado cuentan con una gran imagen o vídeo en la parte más superior de la página web.

Está demostrado por un estudio de Behavioural Analytics que estas imágenes de héroe o hero image captan la atención de la mayor parte de los visitantes. Aprovechemos este hecho para posicionar un pequeño vídeo que hable, en este orden, de:

- Problema del cliente
- Solución potencial
- Producto (características principales)

Beneficios o bullet points (3)

Los bullet points, como su propio nombre indica, son puntos bala. Esto quiere decir que son afirmaciones que van a contar justo lo que necesita saber tu cliente o público objetivo de tu producto para generar FOMO (Fear of missing out, o miedo de perderse algo).

Generalmente los bullet points o beneficios suelen ser funcionalidades de tu producto o servicio, pero recuerda que tienes que venderlos como solución a un problema determinado. Un cliente no te compra un smartphone por tener 512GB de almacenamiento, sino porque puede almacenar todas sus fotos.

4.- Estructura básica de un documento HTML. Secciones, etiquetas y atributos.

Debido a que los navegadores son capaces de procesar diferentes tipos de archivos, lo primero que debemos hacer en la construcción de un documento HTML es indicar su tipo. Para asegurarnos de que el contenido de nuestros documentos sea interpretado correctamente como código HTML, debemos agregar la declaración `<!DOCTYPE>` al comienzo del archivo. Esta declaración, similar en formato a las etiquetas HTML, se requiere al comienzo de cada documento para ayudar al navegador a decidir cómo debe generar la página web.

Los elementos HTML conforman una estructura de tipo árbol con el elemento `<html>` como su raíz. Esta estructura presenta múltiples niveles de organización, con algunos elementos a cargo de definir secciones generales del documento y otros encargados de representar secciones menores o contenido. Los siguientes son los elementos disponibles para definir la columna vertebral de la estructura y facilitar la información que el navegador necesita para mostrar la página en la pantalla.

Etiqueta, Doctype, Head, Body

<code><html></code>	Este elemento delimita el código HTML. Puede incluir el atributo <code>lang</code> para definir el idioma del contenido del documento.
<code><head></code>	Este elemento se usa para definir la información necesaria para configurar la página web, como el título, el tipo de codificación de caracteres y los archivos externos requeridos por el documento.
<code><body></code>	Este elemento delimita el contenido del documento (la parte visible de la página).

El código HTML insertado entre las etiquetas `<html>` se tiene que dividir en dos secciones principales: la cabecera y el cuerpo. Por supuesto, la cabecera va primero y, al igual que el resto de los elementos estructurales, está compuesta por etiquetas de apertura y cierre.

Entre las etiquetas `<head>` debemos definir el título de la página web, declarar el tipo de codificación de caracteres, facilitar información general acerca del documento, e incorporar los archivos externos con estilos y códigos

necesarios para generar la página. Excepto por el título e iconos, el resto de la información insertada en medio de estas etiquetas no es visible para el usuario. Entre las etiquetas que son parte de la cabecera están descritas en la tabla siguiente.

<title>	Este elemento define el título de la página.
<base>	Este elemento define la URL usada por el navegador para establecer la ubicación real de las URL relativas. El elemento debe incluir el atributo href para declarar la URL base. Cuando se declara este elemento, en lugar de la URL actual, el navegador usa la URL asignada al atributo href para completar las URL relativas.
<meta>	Este elemento representa metadatos asociados con el documento, como la descripción del documento, palabras claves, el tipo de codificación de caracteres, etc. El elemento puede incluir los atributos name para describir el tipo de metadata , content para especificar el valor, y charset para declarar el tipo de codificación de caracteres a utilizar para procesar el contenido.
<link>	Este elemento especifica la relación entre el documento y un recurso externo (generalmente usado para cargar archivos CSS). El elemento puede incluir los atributos href para declarar la ubicación del recurso, rel para definir el tipo de relación, media para especificar el medio al que el recurso está asociado (pantalla, impresora, etc.), y type y sizes para declarar el tipo de recurso y su tamaño (usado a menudo para cargar iconos).
<style>	Este elemento se usa para declarar estilos CSS dentro del documento
<script>	Este elemento se usa para cargar o declarar código JavaScript

Como parte de la sección visible de una página (etiqueta <body>), se incluyen todas las estructuras que dan vida al contenido del sitio, y que permiten desplegar la información. Debido a que esta información está compuesta por diferentes elementos visuales, como títulos, textos, imágenes y videos, entre otros, HTML define varios elementos para representarla. A continuación se

indica un listado con etiquetas presentes en esta sección, y su función.

Body h1-h6

<h1>	Este elemento representa un título. El título se declara entre las etiquetas de apertura y cierre. HTML también incluye elementos adicionales para representar subtítulos, hasta seis niveles (<h2>, <h3>, <h4>, <h5>, y <h6>).
<p>	Este elemento representa un párrafo. Por defecto, los navegadores le asignan un margen en la parte superior para separar un párrafo de otro.
<pre>	Este elemento representa un texto con formato predefinido, como código de programación o un poema que requiere que los espacios asignados a cada carácter y los saltos de línea se muestren como se han declarado originalmente.
	Este elemento puede contener un párrafo, una frase o una palabra. No aplica ningún estilo al texto pero se usa para asignar estilos personalizados.

	Este elemento se usa para insertar saltos de línea.
<wbr>	Este elemento sugiere la posibilidad de un salto de línea para ayudar al navegador a decidir dónde cortar el texto cuando no hay suficiente espacio para mostrarlo entero.
	Este elemento se usa para indicar énfasis. El texto se muestra por defecto con letra cursiva.
	Este elemento se utiliza para indicar importancia. El texto se muestra por defecto en negrita.
<i>	Este elemento representa una voz alternativa o un estado de humor, como un pensamiento, un término técnico, etc. El texto se muestra por defecto con letra cursiva.
<u>	Este elemento representa texto no articulado. Por defecto se muestra subrayado.
	Este elemento se usa para indicar importancia. Debería ser implementado solo cuando ningún otro elemento es apropiado para la situación. El texto se muestra por defecto en negrita.

<mark>	Este elemento resalta texto que es relevante en las circunstancias actuales (por ejemplo, términos que busca el usuario).
<small>	Este elemento representa letra pequeña, como declaraciones legales, descargos, etc.
<cite>	Este elemento representa el autor o título de una obra, como un libro, una película, etc.
<address>	Este elemento representa información de contacto. Se implementa con frecuencia dentro de los pies de página para definir la dirección de la empresa o el sitio web.
<time>	Este elemento representa una fecha en formato legible para el usuario. Incluye el atributo datetime para especificar un valor en formato de ordenador y el atributo pubdate , el cual indica que el valor asignado al atributo datetime representa la fecha de publicación.
<code>	Este elemento representa código de programación. Se usa en conjunto con el elemento <pre> para presentar código de programación en el formato original.
<data>	Este elemento representa datos genéricos. Puede incluir el atributo value para especificar el valor en formato de computador (por ejemplo, <data value="32">Treinta y Dos</data>).
<a>	Este elemento crea un enlace. El texto o la imagen que representa el enlace se incluye entre las etiquetas de apertura y cierre. El elemento incluye el atributo href para especificar la URL del enlace.
<download>	Este es un atributo booleano que, cuando se incluye, indica que en lugar de leer el archivo el navegador debería descargarlo.
<ping>	Este atributo declara la ruta del archivo que se debe abrir en el servidor cuando el usuario hace clic en el enlace. El valor puede ser una o más URL separadas por un espacio.

Además de las etiquetas mencionadas en las tablas anteriores, dentro del estándar se consideran categorías que determinan diferentes

comportamientos y usos del lenguaje.

Editor de código

A menudo la información se debe representar como una lista de ítems. Por ejemplo, muchos sitios web incluyen listados de libros, películas, o términos y descripciones. Para crear estos listados, HTML ofrece los siguientes elementos.

	Este elemento crea una lista de ítems sin orden. Está compuesto por etiquetas de apertura y cierre para agrupar los ítems (y) y trabaja junto con el elemento para definir cada uno de los ítems de la lista.
	Este elemento crea una lista ordenada de ítems. Está compuesto por etiquetas de apertura y cierre para agrupar los ítems (y) y trabaja junto con el elemento para definir los ítems de la lista. Este elemento puede incluir los atributos reversed para invertir el orden de los indicadores, start para determinar el valor desde el cual los indicadores tienen que comenzar a contar y type para determinar el tipo de indicador que queremos usar. Los valores disponibles para el atributo type son 1 (números), a (letras minúsculas), A (letras mayúsculas), i (números romanos en minúsculas) e I (números romanos en mayúsculas).
<dl>	Este elemento crea una lista de términos y descripciones. El elemento trabaja junto con los elementos <dt> y <dd> para definir los ítems de la lista. El elemento <dl> define la lista, el elemento <dt> define los términos y el elemento <dd> define las descripciones.

Los siguientes elementos se han diseñado con propósitos diferentes, pero también se utilizan frecuentemente para construir listas de ítems.

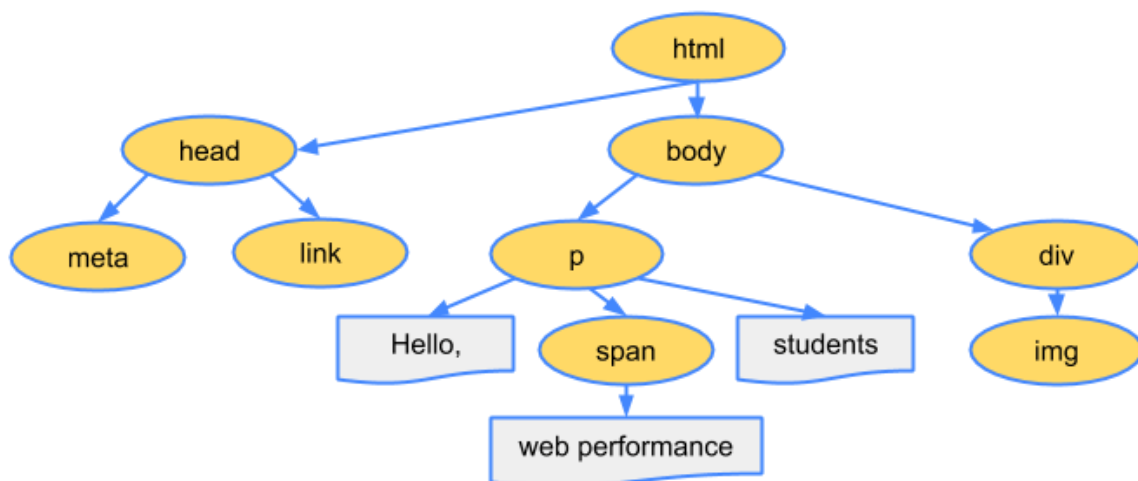
<blockquote>	Este elemento representa un bloque de texto que incluye una cita tomada de otro texto en el documento.
<details>	Este elemento crea una herramienta que se expande cuando se hace clic en ella para mostrar información adicional. La parte visible se define con el elemento

	<summary>, y se pueden usar elementos comunes como <p> para definir el contenido.
--	---

Navegador DOM

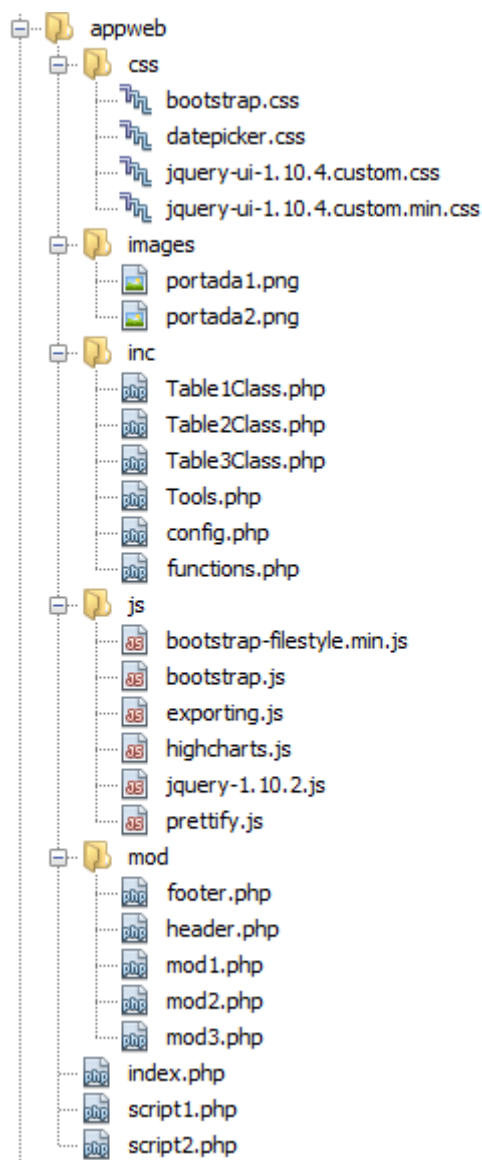
El modelo de objeto de documento (DOM) es una interfaz de programación para los documentos HTML y XML. Facilita una representación estructurada del documento y define de qué manera los programas pueden acceder, al fin de modificar, tanto su estructura, estilo y contenido. El DOM da una representación del documento como un grupo de nodos y objetos estructurados que tienen propiedades y métodos. Esencialmente, conecta las páginas web a scripts o lenguajes de programación.

Una página web es un documento. Éste documento puede exhibirse en la ventana de un navegador o también como código fuente HTML. Pero, en los dos casos, es el mismo documento. El modelo de objeto de documento (DOM) proporciona otras formas de presentar, guardar y manipular este mismo documento. El DOM es una representación completamente orientada al objeto de la página web y puede ser modificado con un lenguaje de script como JavaScript.



Organización de assets de un proyecto web

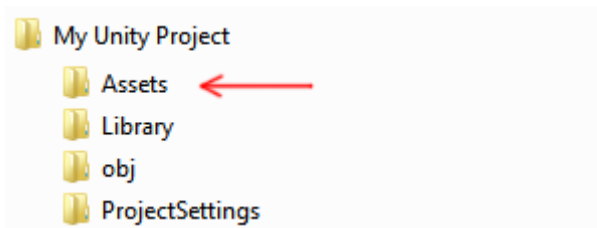
Aunque haya cambiado mucho la forma de diseñar los proyectos web, sin embargo, hay cierto tipo de tareas que se siguen necesitando. Por ejemplo, la exportación de los assets necesarios. O sea, generar los JPG o PNG que vayamos a incluir dentro de la aplicación. En el diseño que tengo en pantalla, de hecho, tengo una pequeña imagen que es la que quiero que se exporte para poderla utilizar dentro de la aplicación. Incluso esta tarea de generación de assets, vamos a ver que se ha mejorado. Para exportar los diferentes assets, yo lo único que necesito es seleccionar el que quiera que sea exportable y marcar, en el panel de Propiedades, Hacer exportable. Básicamente, lo que genera es un determinado preset.



La carpeta del proyecto

Los Assets creados afuera de Unity deben ser traídos a Unity al tener el archivo ya sea guardado directamente a la carpeta “Assets” de su proyecto, o copiado a esa carpeta. Para muchos formatos comunes, usted puede guardar su archivo fuente directamente a la carpeta Assets de su proyecto y Unity será capaz de leerlo. Unity va a notar cuando usted ha guardado nuevos cambios al archivo y va a re-importarlo si es necesario.

Cuando crees un proyecto de Unity, usted está creando una carpeta - llamada por su proyecto - que contiene las siguientes sub-carpetas:



La carpeta Assets es dónde usted debería guardar o copiar los archivos que usted quiere utilizar en su proyecto.

Los contenidos de la Project Window en Unity muestra los items en su carpeta de Assets. Por lo que si usted guarda o copia un archivo a su carpeta de Assets, este será importado y será visible en su Project Window.

Unity detectará archivos automáticamente a medida que son agregados a la carpeta Assets, o si son modificados. Cuando coloque cualquier asset en la carpeta Assets, verás al asset aparecer en su Project View.

Rutas locales y absolutas

Los enlaces de Internet están formados por una serie de rutas (también conocidos con el término inglés 'path'), donde se le indica la dirección a la que tiene que ir el navegador cuando pulsamos sobre ese link. Nos podemos encontrar dos tipos de rutas distintas:

- Ruta absoluta: Incluye el nombre del dominio.

Ejemplo: 'https://www.hostalia.com/contratacion/openxchange/'

- Ruta relativa: Sólo indica el orden de directorios.

Ejemplo: '/contratacion/openxchange/'

Formatos de imágenes

Las imágenes pueden ser consideradas el segundo medio más importante en la Web. HTML incluye los siguientes elementos para introducir imágenes en los documentos que conforman un portal.

	Este elemento inserta una imagen en el documento. El elemento requiere del atributo src para especificar la URL del archivo con la imagen que queremos incorporar.
<picture>	Este elemento inserta una imagen en el documento. Trabaja junto con el elemento <source> para ofrecer múltiples imágenes en diferentes resoluciones. Es útil para crear sitios web adaptables.
<figure>	Este elemento representa contenido asociado con el contenido principal, pero que se puede eliminar sin que se vea afectado, como fotos, vídeos, etc.
<figcaption>	Este elemento introduce un título para el elemento <figure> .

Accesibilidad web

La accesibilidad Web significa que personas con algún tipo de discapacidad van a poder hacer uso de la Web. Al hablar de accesibilidad Web se está haciendo referencia a un diseño Web que permitirá que estas personas puedan percibir, entender, navegar e interactuar con la Web, aportando a su vez contenidos. La accesibilidad Web también beneficia a otras personas, incluyendo personas de edad avanzada que han visto mermadas sus habilidades a consecuencia de la edad.

La accesibilidad Web considera muchos tipos de discapacidades, incluyendo problemas visuales, auditivos, físicos, cognitivos, neurológicos y del habla. El documento "Cómo utilizan la Web personas con discapacidad" muestra la forma en la que diferentes discapacidades pueden dificultar la utilización de la Web e incluye algunos escenarios de personas con discapacidad utilizando la Web.

La Web es un recurso muy importante para diferentes aspectos de la vida: educación, empleo, gobierno, comercio, sanidad, entretenimiento y muchos otros. Es muy importante que la Web sea accesible para así proporcionar un acceso equitativo e igualdad de oportunidades a las personas con discapacidad. Una página Web accesible puede ayudar a personas con discapacidad a que participen más activamente en la sociedad.

La accesibilidad Web se ha entendido siempre como responsabilidad de los desarrolladores Web. Sin embargo, el software Web tiene también un papel importante en la accesibilidad Web. Es importante que el software ayude a los desarrolladores a generar y evaluar sitios Web accesibles para que las personas con discapacidad puedan utilizarlos. Una de las funciones de la Iniciativa de Accesibilidad Web (WAI) es desarrollar pautas y técnicas que proporcionen soluciones accesibles para el software Web y para los desarrolladores Web. Las pautas de WAI son consideradas como estándares internacionales de accesibilidad Web.

Hacer un sitio Web accesible puede ser algo sencillo o complejo, depende de muchos factores como por ejemplo, el tipo de contenido, el tamaño y la complejidad del sitio, así como de las herramientas de desarrollo y el entorno. Muchas de las características accesibles de un sitio se implementan de forma sencilla si se planean desde el principio del desarrollo del sitio Web o al comienzo de su rediseño. La modificación de sitios Web inaccesible puede requerir un gran esfuerzo, sobre todo aquellos que no se "etiquetaron" correctamente con etiquetas estándares de XHTML, y sitios con cierto tipo de contenido, como multimedia. Existen, por cierto, sitios que validan que la estructura del sitio cumpla con los estándares solicitados.

Carga de archivos CCS, Local, CDN

El tipo y las consultas de medios son esenciales para saber cuándo se carga un CSS. Veamos algunos ejemplos.

```
<link href="style.css" rel="stylesheet">
<link href="style.css" rel="stylesheet" media="all">
<link href="print.css" rel="stylesheet" media="print">
<link href="other.css" rel="stylesheet" media="(min-width: 40em)">
<link href="portr.css" rel="stylesheet" media="orientation:portrait">
```

1. La primera declaración bloquea la presentación debido a que si no se especifica el atributo media el browser asume por defecto el valor all, es decir, todos los medios. En cualquier caso el CSS se cargará.

2. La segunda declaración define de manera explícita todos los medios, por lo cual es exactamente igual al ejemplo anterior y bloquea siempre la presentación.
3. La tercera declaración no bloquea la presentación debido a que solo se aplica durante la impresión de la página. Haría falta que el usuario intente imprimir la página para que dicho evento lanzara la carga del CSS.
4. La cuarta declaración bloqueará la presentación cuando se presente la condición especificada.
5. La última declaración podría bloquear la presentación dependiendo de la orientación del dispositivo al momento de cargar la página. Si la página se carga en portrait bloqueará la presentación.

Anexo: Referencias

[1] Sitio oficial de Visual Studio Code

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

[2] Mejores navegadores web

Referencia:

<https://es.digitaltrends.com/computadoras/mejores-navegadores-web/>

[3] J.D Gauchat “El Gran libro de HTML5, CCS3 y JavaScript” 3ª edición

[4] ¿Qué es una landing Page?

<https://wayik.com/que-es-mejor-pagina-web-o-landing-page/>