HITO DEL 1ER TRIMESTRE DE LENGUAJES DE MARCAS

Pablo Martin Tomasi

CampusFP

Fecha limite: miercoles 20 de noviembre de 2024

Contenido

[1.Explica los orígenes de los lenguajes de marcas. 2](#_Toc182812419)

[2.Explicar los diferentes estándares y organismos oficiales en el desarrollo y diseño de lenguajes de marcas. 3](#_Toc182812420)

[3.Explica los orígenes y las diferencias entre CSS y SCSS. 4](#_Toc182812421)

[4.Principales etiquetas HTML5, así como una breve descripción de que son y cómo funcionan las etiquetas semánticas. 5](#_Toc182812422)

[5. Principales elementos CSS y su aplicación. 7](#_Toc182812423)

[6. Guía de creación de un sitio web con el paso a paso del desarrollo. 9](#_Toc182812424)

[7. Guia y uso de librerías de CSS para el desarrollo web 11](#_Toc182812425)

[8. Web grafía en cada HTML de las páginas consultadas. 14](#_Toc182812426)

[9. Enlace Github 15](#_Toc182812427)

# 1.Explica los orígenes de los lenguajes de marcas.

Los lenguajes de marcas surgieron, inicialmente, como lenguajes formados por el conjunto de códigos de formato que los procesadores de texto introducen en los documentos para dirigir el proceso de presentación mediante una linotipia o impresora programable.

# 2.Explicar los diferentes estándares y organismos oficiales en el desarrollo y diseño de lenguajes de marcas.

Los estándares en lenguajes de marcas son especificaciones técnicas diseñadas para garantizar uniformidad, interoperabilidad y accesibilidad en el desarrollo de tecnologías relacionadas con la web y otros sistemas.

W3C, su función principal es que es el organismo principal de desarrollar estándares para la web, incluyendo el lenguaje de marcas con el HTML, CML y SVG

# 3.Explica los orígenes y las diferencias entre CSS y SCSS.

* **CSS:**
  + Orígenes: Håkon Wium Lie propuso por primera vez el CSS el 10 de octubre de 1994 , y la primera Recomendación CSS del W3C (CSS1) se publicó en 1996. Está diseñada para permitir la separación de contenido y presentación, como colores, fuentes y diseño. La separación de contenido y presentación puede mejorar la usabilidad del contenido y dar más flexibilidad para controlar la especificación de presentación.
* **SCSS:**
  + Orígenes: SCSS son las siglas de Sassy Cascading Style Sheets (hojas de estilo en cascada Sassy) . La variante más avanzada de CSS es SCSS . Fue creada por Chris Eppstein y Natalie Weizenbaum y diseñada por Hampton Catlin . También se la conoce como Sassy CSS debido a sus características avanzadas. Es un lenguaje de preprocesador que se compila o interrumpe en el CSS. Tiene una extensión de archivo .scss .
* **Diferencias:**
  + SCSS incluye todas las características de CSS y otras características que no están disponibles en CSS, lo que lo convierte en una alternativa sólida para que los desarrolladores lo utilicen.
  + CSS es un lenguaje de estilo que se utiliza para diseñar y crear páginas web. Si bien SCSS es un tipo de archivo particular para SASS, utiliza el lenguaje Ruby, que ensambla las hojas de estilo CSS del navegador.
  + SCSS contiene funciones avanzadas y modificadas.
  + SCSS es más expresivo que CSS. SCSS utiliza menos líneas en su código que CSS, lo que facilita la carga del código.
  + Promueve la correcta anidación de reglas. El CSS normal no facilita la anidación. Dentro de otra clase, no podemos escribir una clase. Esto genera un problema de legibilidad a medida que el proyecto crece y el diseño no se ve bien.
  + Se pueden utilizar varias hojas de estilo en una sola página con algunos cambios simples en el código CSS. Esto tiene ventajas en términos de usabilidad y la capacidad de personalizar un sitio web para distintos dispositivos de destino.
  + Con SCSS podemos incluir diversas funciones en el código en forma de variables, anidación y selectores. Por el contrario, estas funciones no están presentes en el CSS.
  + La sintaxis SCSS utiliza sangrías que no están presentes en CSS.
  + SCSS nos ayuda a utilizar los operadores para realizar operaciones matemáticas. Dentro de nuestro código, podemos realizar cálculos simples para un mejor rendimiento.
  + El conocimiento de SCSS ayuda a personalizar Bootstrap 4.

# 4.Principales etiquetas HTML5, así como una breve descripción de que son y cómo funcionan las etiquetas semánticas.

**<header>**

Define la cabecera de una sección o página. Suele contener el título, logo y el menú de navegación. Indica el contenido introductorio o de encabezado en una sección.

**<footer>**

Representa el pie de página de una sección o documento, que normalmente incluye información de contacto, derechos de autor o enlaces adicionales.

**<nav>**

Nos indica los apartados que pueden tener nuestro sitio web. Para poder utilizarlo bien es recomendable utilizar bootstrap que es un biblioteca de CSS.

**<article>**

Define contenido autónomo y completo, como una entrada de blog o una noticia. Ideal para contenido que tiene sentido por sí mismo.

**<section>**

Define una sección temática de contenido dentro de una página. Se usa para organizar el contenido en bloques lógicos.

**<aside>**

Contiene contenido relacionado con el principal, como una barra lateral o enlaces adicionales. Es útil para mostrar información complementaria o relevante.

**<main>**

Define el contenido principal de la página, excluyendo encabezados, pies de página y barras de navegación. Ayuda a los navegadores a identificar el contenido central.

**<figure>**

Encapsula contenido independiente, como imágenes, gráficos o diagramas, junto con su leyenda. Normalmente se usa con <figcaption>.

**<figcaption>**

Proporciona una leyenda o descripción para el contenido dentro de <figure>, lo cual ayuda a entender mejor el contexto visual.

**<time>**

Representa una fecha u hora específica. Los navegadores y motores de búsqueda lo interpretan para eventos temporales.

**<details>**

Define un contenido expandible que el usuario puede abrir y cerrar, como una sección de preguntas frecuentes.

**<summary>**

Se usa dentro de <details> para proporcionar un encabezado visible. Al hacer clic en <summary>, se despliega el contenido de <details>.

**<mark>**

Resalta texto que es relevante en un contexto específico, como términos buscados en un documento.

**<dialog>**

Define un cuadro de diálogo o ventana emergente que puede aparecer en la pantalla para alertas o confirmaciones.

**<address>**

Indica la información de contacto de una persona o entidad. Generalmente se usa en el pie de página.

**<progress>**

Representa el progreso de una tarea en curso, como una barra de carga o de porcentaje completado.

**<output>**

Define el resultado de un cálculo o acción, como el resultado de una operación en un formulario.

**<template>**

Almacena fragmentos de HTML que no se renderizan en la página hasta que se activa mediante JavaScript. Ideal para generar contenido dinámico.

**<abbr>**

Representa una abreviatura o acrónimo, y puede incluir el atributo title para mostrar el significado completo.

**<bdi>**

Sirve para aislar una sección de texto con una dirección de lectura diferente (de derecha a izquierda), preservando el formato de los elementos adyacentes.

**<strong>**

Define texto con énfasis fuerte, que generalmente se presenta en negrita. Se usa para indicar importancia o relevancia del contenido.

**<em>**

Define texto con énfasis, normalmente representado en cursiva. Indica que el contenido tiene un énfasis especial o debe destacarse.

# 5. Principales elementos CSS y su aplicación.

list-style  
Personaliza el estilo de las listas, incluyendo el tipo de marcador (por ejemplo, círculo, cuadrado) o una imagen.

text-transform  
Controla el formato de un texto, que permite transformar en mayúsculas, minúsculas o capitalizar cada palabra

visibility  
Nos permite mostrar u ocultar los elementos en un sitio web.

float  
Nos permite poder colocar un elemento en la izquierda o en la derecha de su contenedor, permitiendo que los otros elementos floten a su alrededor.

clear  
Se utiliza junto a float para evitar que un elemento flote alrededor de otros, asegurando que los elementos se disponen debajo de los flotantes.

max-width y min-width  
Establece el ancho máximo que puede tener un elemento.  Establece el ancho mínimo que debe tener un elemento

min-height y max-height  
Establece el alto máximo que debe tener un elemento. Establece el alto mínimo que debe tener un elemento

object-fit  
Controla cómo se redimensiona las imágenes o videos dentro de su contenedor, manteniendo o ajustando la proporción.

align-items  
Utilizado en contenedores de tipo flex, define cómo se alinean los elementos hijos en el eje transversal (vertical por defecto en flex).

justify-content  
También en flex, controla la alineación de los elementos en el eje principal (horizontal por defecto).

grid-template-columns  
Define el número y tamaño de las columnas en un contenedor grid.

grid-template-rows  
Similar a grid-template-columns, pero define las filas en un contenedor grid.

filter  
Aplica efectos visuales como desenfoque, brillo o saturación a imágenes y elementos visuales.

aspect-ratio  
Establece una proporción de aspecto (ancho) para un elemento, útil en imágenes y videos para mantener una relación constante en diferentes tamaños.

white-space  
Controla cómo se gestiona el espacio en blanco dentro de un elemento.

background-attachment  
Controla si el fondo se desplaza con el contenido o se mantiene fijo cuando se hace scroll.

# 6. Guía de creación de un sitio web con el paso a paso del desarrollo.

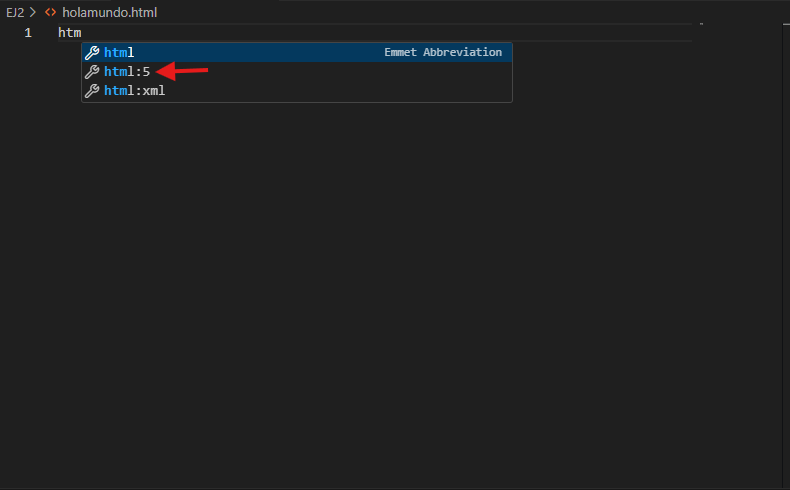
Para crear un sitio web es recomendable empezar instalándonos, una aplicación para poder ingresar nuestro código. Para esta guía vamos a utilizar Visual Studio Code.

Una vez instalado la aplicación, creamos una carpeta con el título de nuestro proyecto que tenemos en mente. Luego creamos en nuestra carpeta una nueva carpeta con el nombre de imágenes o img, para poderla ingresar un nuestro sitio web solo si quieres.

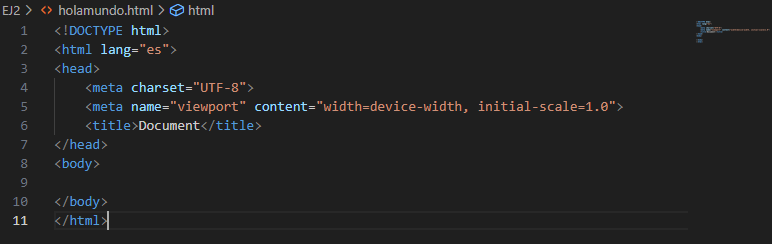
Una vez creada esas dos carpetas abrimos el Visual Studio Code, le hacemos clic a archivo que está arriba a la izquierda y luego le hacemos clic a abrir carpeta y seleccionamos la carpeta en la que queremos crear nuestro sitio web.

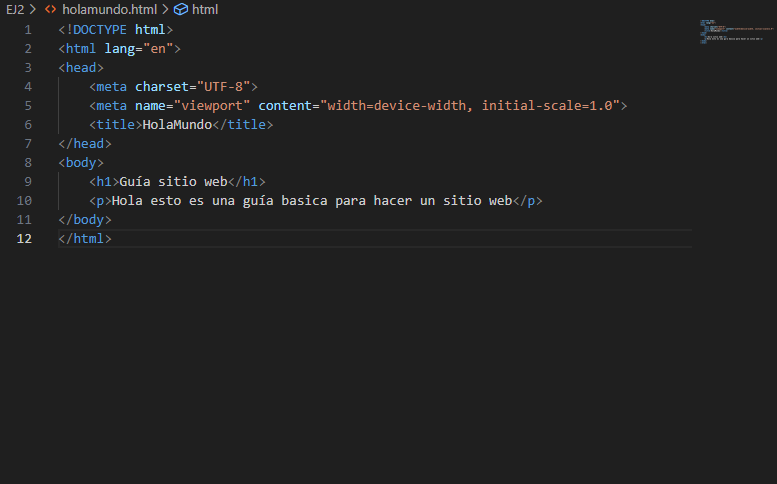
En la carpeta hacemos un nuevo archivo, y el título de ese archivo es ejemplo.html, evidentemente pones el nombre que deseas. Por ejemplo puedes poner holamundo.html.

Y en ese archivo escribes html:5, y le das clic a tab para poder hacer el html:



Una vez haces clic en html: 5, te va a salir la base, y también en lang que es el lenguaje cámbialo a es si quieres que no te salgan errores al escribir en español, o cámbialo a la lengua que quieras escribir. En defecto esta puesto en ingles(en)



Luego le puedes poner en <title> el nombre de tu sitio web</title>. Y luego en <body> </body> escribes tu código por ejemplo puedes hacer esto: 

# 7. Guia y uso de librerías de CSS para el desarrollo web

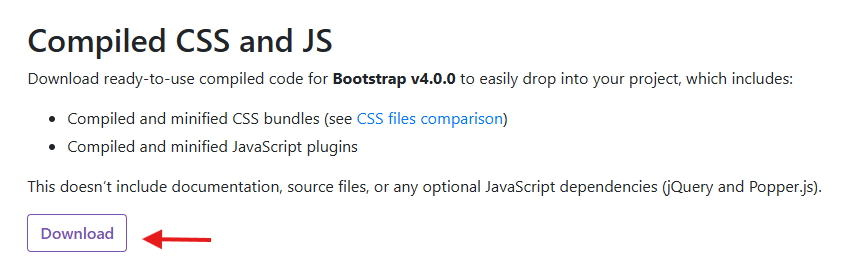
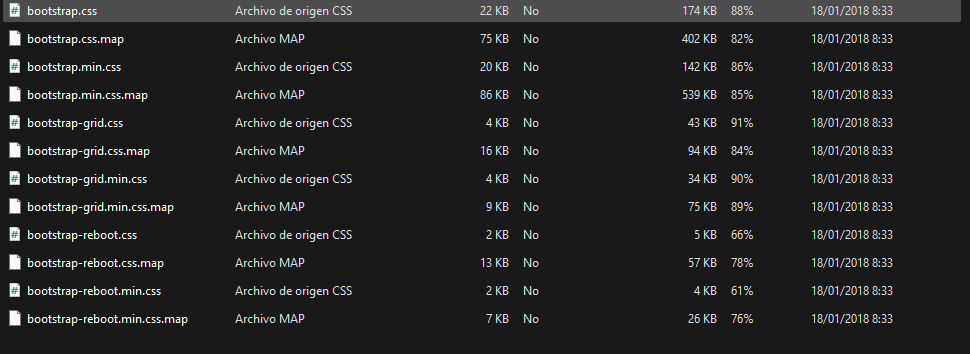
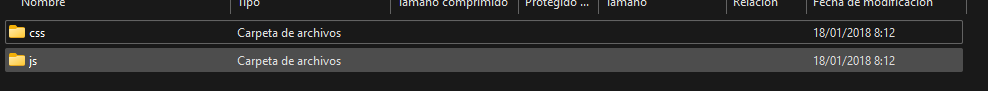
En el momento de crear tu sitio web, se puede usar una librería de CSS. Para empezar, tienes que crear una carpeta CSS. Y esa tu carpeta CSS puedes usar una librería, en esta guía vamos a enseñar la librería de boostrap.

Esta librería de CSS, incluyen nuevos estilos para introducir en tu sitio web, algunos de los elementos que hay botones, tarjetas, navbars, etc.

Algunas ventajas para usar Boostrap:

* Facilidad de uso. Lo único que tienes que hacer es poner clases a elementos HTML.
* Responsive. Perfecto para crear páginas web adaptables a cualquier dispositivo.
* Personalizable. Puedes personalizar su descarga para usar los elementos que necesites.

Como descargar y instalar Boostrap, para poder usar Boostrap en un proyecto:

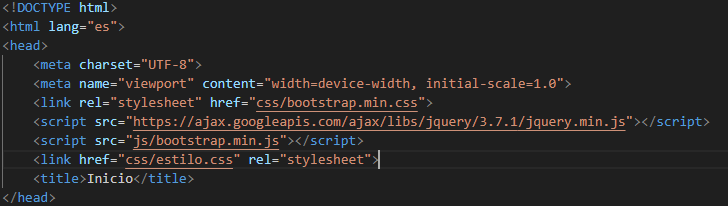
* Para poder intalar la librería de Boostrap entras en este enlace: <https://getbootstrap.com/docs/4.0/getting-started/download/>
* Una vez en el enlace, descargas boostrap
* Luego descomprimes la carpeta, y en la carpeta donde está tu proyecto, entras en la carpeta CSS, y pones todos los archivos para la librería de boostrap: 
* Lego en la carpeta de tu proyecto pones la carpeta de js
* Una vez has seguido estos pasos, entras en los archivos de tu sitio web y entre <head> y </head> pones esto:

<link rel="stylesheet" href="css/bootstrap.min.css">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>

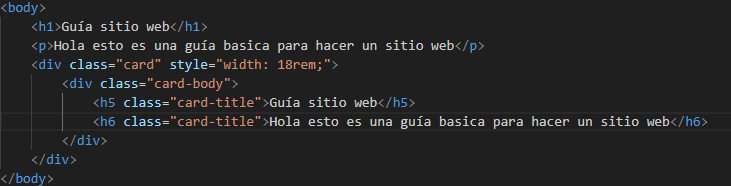
    <script src="js/bootstrap.min.js"></script>

    <link href="css/estilo.css" rel="stylesheet">



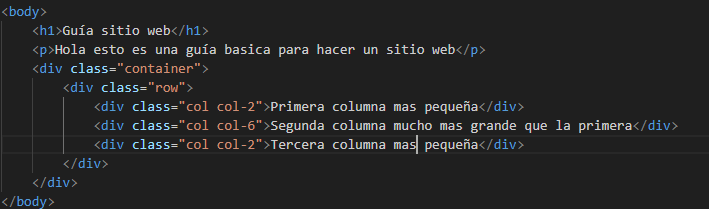
Una vez seguido estos pasos puedes usar algunos ejemplo:

* Tarjetas:



* Nav:



* Hacer columnas fácilmente: 

# 8. Web grafía en cada HTML de las páginas consultadas.

-*ChatGPT*. (s. f.). Chatgpt.com. Recuperado 18 de noviembre de 2024, de <https://chatgpt.com/c/6733bb7c-3524-8005-af17-4e9fbaa4661c>

-Cruz, A. L. (2023, agosto 2). *Tabla resumen de propiedades CSS y sus valores*. Eniun. <https://www.eniun.com/resumen-tabla-propiedades-css-valores/>

-*Difference between CSS and SCSS*. (s. f.). Www.javatpoint.com. Recuperado 18 de noviembre de 2024, de <https://www-javatpoint-com.translate.goog/css-vs-scss?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=rq>

-Wikipedia contributors. (s. f.). *Lenguaje de marcado*. Wikipedia, The Free Encyclopedia. <https://es.wikipedia.org/w/index.php?title=Lenguaje_de_marcado&oldid=163261575> .

# 9.Enlace Github

<https://github.com/PabloMartinTomasi/Lenguaje-de-marcas/tree/primer-trimestre/Hito_LM_1T_Pablo_Martin_Tomasi>