Desarrollo web con HTML, CSS y JavaScript

Ingniero Informático

Autor

Ismael Sallami Moreno

Universidad de Granada

2024

Sumário

1 Introducción

Vamos a ver unas conceptos muy simples a continuación para introducirnos al desarrollo web:

1.1 Frontend

Se trata de las tecnologías que se emplean en el lado del cliente, como es Chrome, Firefox, Safari y demás. Los lenguajes que se suelen utilizar son HTML, CSS y JavaScript.

1.2 Backend

Hace referencia a los servicios que se ejecutan por el lado del servidor. Por temas de seguridad no se puede ver el código. El lenguaje común es Java, Ruby, NodeJs, PHP, ...

2 HTML

Se trata del lenguaje de marcas de hipertexto. hace referencia al lenguaje de marcado utilizado en la creación de páginas web. Este estándar que sirve de referencia del software que interactúa con la elaboración de páginas web en sus diferentes versiones. Define una estructura básica y un código (denominado código HTML) para la presentación de contenido de una página web, que incluye texto, imágenes, videos, juegos, entre otros elementos.
Podemos relizaro en la plataforma que queramo, yo uso Visual Studio Code.
Comenzamos creando un archivo denominada **index.html** que va a contener lo que s eva a reflejar en la web, si abrimos ese archivo en nuestro navegador, expone lo que hayamos escrito.

Las **etiquetas** es todo lo que podemos agrupar o expander con tocarle, por ejemplo **<html>**.

La etiqueta **<head>** determina lo que no se va a ver exactamente en la página web, como son los metadatos y demás.

Ahora a continuación voy a ir proporcionando el código de programación con una breve explicación y si es necesario iré recalcando diversos detalles y explicando.

```
<!DOCTYPE html>
<!--
    Al poner este comando se nos pone la última versión de html
-->

<html>
    <head>
        <meta charset="utf-8" />
        <title>Aprendiendo con Ismael Sallami</title>
    </head>

    <body>

        <ul>
            <li>
                <a href="index.html" >Inicio </a>
            </li>
            <li>
                <a href="Contacto.html">Contacto</a>
            </li>

        </ul>
```

Figura 2.1: Index.html

Como podemos ver el contenido que posee head es dedicado a que se reconozca el texto y demás. La etiqueta **body** hace que lo que pongamos dentro sea lo que el usuario puede ver y trabajar con el. La etiqueta **ul** representa una lista desordenada donde cada elemento se introduce con **li**.
También podemos ver como encadenamos un módulo con otro al poner **href**.

Las etiquetas **h1** y **h2** hacen referencia a encabezados de menor o mayor tamaño. **P** es la etiqueta destinada a introducir texto, mientras que **br** es el salto de línea. **Ol** se trata de una lista ordenada y funciona de manera similar a ul.
En este caso tratamos con las imágenes con la etiqueta **img**, añadiéndole atributos como los pixeles con width, la descripción de la imagen con alt y la imagen con **src**, que podemos proporcionar tanto la url como la ruta relativa de la imagen dentro de nuestro proyecto.

```
24
25          <h1>Encabezado de esta página</h1>
26          <h2>Encabezado de segundo nivel</h2>
27
28          <p>Todo el texto que yo quiera(párrafo)</p>
29
30          <br/><!--No se abre y se cierra como las demás etiquetas si no que produce el efecto del salto de
31          línea-->
32
33          <hr> <!--Crea una léna separadora-->
34          <h1>Listas</h1>
35          <ul> <!--Lista desordenada, es decir, con · -->
36              <li>Golf</li>
37              <li>Porsche</li>
38              <li>Mercedes</li>
39          </ul>
40
41          <ol> <!--Lista dordenada, es decir, con números -->
42              <li>Golf</li>
43              <li>Porsche</li>
44              <li>Mercedes</li>
45          </ol>
46
```

Figura 2.2: Index.html

```
47          <hr/>
48
49          <h1>Imágenes</h1>
50
51          <img width="350px"    alt="Esto es un Porsche" src="https://th.bing.com/th/id/R.41be98a8483de17a84c53be1c68ad73a?rik=hNKtoObLXyHhlw&riu=http%3a%2f%2fcdn.carbuzz.com%2fgallery-images%2f1
52
53
54          <P>Esta es otra imagen pero a diferencia de la anterior que ha sido con la url de dicha imagen, vamos a introducir la ruta relativa dentro del proyecto de una imagen que tengamos.</P>
55
56          <img width="350px" alt="Omg que posche" src="images/P.jpg" />
57      </body>
58  </html>
```

Figura 2.3: Index.html

Ahora vamos a ver el otro módulo:

Podemos usar tablas con **table** y demás que viene ahí explicado en la imagen anterior.

También formularios, etiquetas de selección y poder enviar variables resultado para almacenarlas en una base de datos para su posterior uso.

```html
<!DOCTYPE html>
<!--
    Al poner este comando se nos pone la última versión de html
-->

<html>
    <head>
        <meta charset="utf-8" />
        <title>Aprendiendo con Ismael Sallami</title>
    </head>

    <body>
        <h1>Página de contacto</h1>
        <ul>
            <li>
                <a href="index.html" >Inicio </a>
            </li>
            <li>
                <a href="Contacto.html">Contacto</a>
            </li>

            <!--Como definir tablas? Para ello debemos de usar la etiqueta table, y para indicar
            las filas es con tr y las columnas con td, ponemos border para indicar que la tabla debe
            de rellenar todas los cuadrafos con bordes-->

            <h1> Tablas </h1>
            <table border="1">
                <tr>
                    <td>Nombre</td>
                    <td>Marca</td>
                    <td>Modelo</td>
                </tr>
                <tr>
                    <td>Nissan 350Z</td>
                    <td>Nissan</td>
                    <td>350Z 290cv</td>
                </tr>
            </table>
```

Figura 2.4: contacto.html

```html
            <!--También podemos introducir formularios que sean capaces de recoger entradas de texto-->
            <h1>Formularios</h1>
            <p>Formularios de contacto</p>

            <form>
                <label>Nombre</label>
                <input type="text"/>
                <!--LABEL  es la etiqueta de como se va a llamr el campo donde se va a introducir el texto
                e input es la etiqueta encargada de recogerla-->

                <!--Podemos usar la etiqueta select que nos permite seleccionar opciones dentro de las opciones definidas con
                la etiqueta option, y se le suele poner un value para guardar dicha variable en una base de datos-->

                <br/>
                <select>
                    <option value="hombre" > Hombre</option>
                    <option value="mujer" > Mujer</option>
                </select>

                <!--Tambien podemos hacer un boton de texto, donde el value contiene el texto de dicho boton-->

                <input type="submit" value="Enviar"/>
            </form>

        </ul>
    </body>
</html>
```

Figura 2.5: contacto.html

3 CSS

3.0.1 Primer ejemplo

CSS (siglas en inglés de Cascading Style Sheets), en español «Hojas de estilo en cascada», es un lenguaje de diseño **gráfico** para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles (como Firefox OS). (Estilo, colores, forma y apariencia).

Luego vamos a ir comentándo el código dando breves explicaciones:

Tenemos varias formas de editar el estilo como se expone en la imagen, la más recomendable es la segunda con style.css.

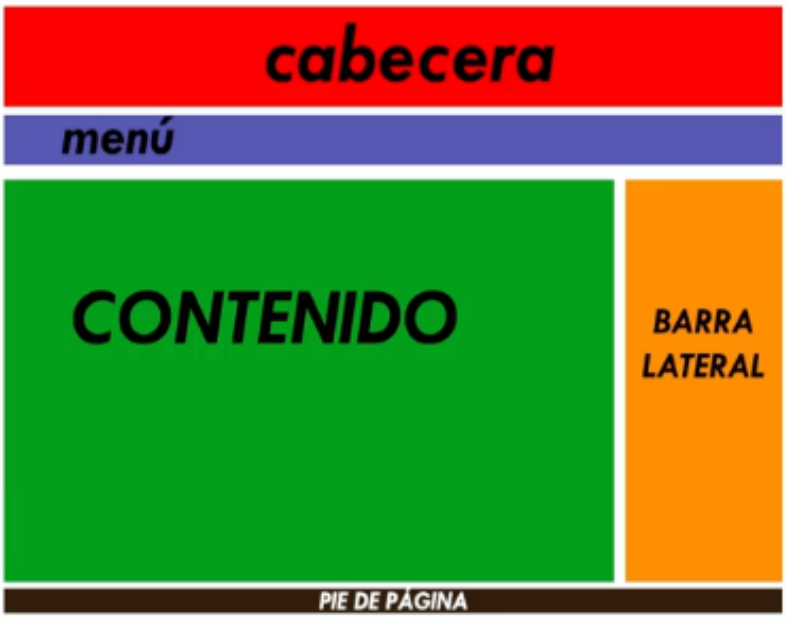


Figura 3.1: Estructura del sitio web que vamos a estar creando en un primer momento

Todo esta explicado con la idea principal y básica en el código.

Todo esta explicado brevemente en el código.

3.0.2 Segundo ejemplo FlexBox

Más cajas de por medio creadas, todas siguen la misma sintaxis a diferencia de estas dos:

Las dos últimas son distintas con el objetivo de darles un formato distinto e independiente en eel style.css.

Todo esta explicado brevemente en el código de manera comentada.//

```html
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <title>CSS con Ismael Sallami</title>

        <!--Hay varias formas de editar CSS, una de ellas puede ser con la etiqueta style o creando
        otra
        hoja por ej: estilos.css para ello debemos de vincularlo

        <style>
            body{
                background: lightskyblue;
            }

            #container{

            }

            .article{

            }
        </style>
        -->

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

    </head>
```

Figura 3.2: Primer ejemplo index.html

```
    <body>

        <div id="container">
            <!--Div es una caja y el id hace que luego en el CSS le pueda dar estilo y demás buscando
            con ese identificador-->

            <header>
                <!--CABECERA de mis sitio web-->
                <h1>
                    Ismael Sallami te ayuda a aprender CSS
                </h1>
            </header>


                <!--Definimos un menú de navegación con la etiqueta nav que es de HTML 5-->

                <nav>
                    <ul>
                        <li>
                            <a href="index.html">
                                Inicio
                            </a>
                        </li>
                        <li>
                            <a href="index.html">
                                Página 1
                            </a>
                        </li>
                        <li>
                            <a href="index.html">
                                Página 2
                            </a>
                        </li>
                        <li>
                            <a href="index.html">
                                Blog
                            </a>
                        </li>
                        <li>
                            <a href="index.html">
                                Contacto
                            </a>
                        </li>
                    </ul>
                </nav>

                <div class="clearfix"></div>

```

Figura 3.3: Primer ejemplo index.html

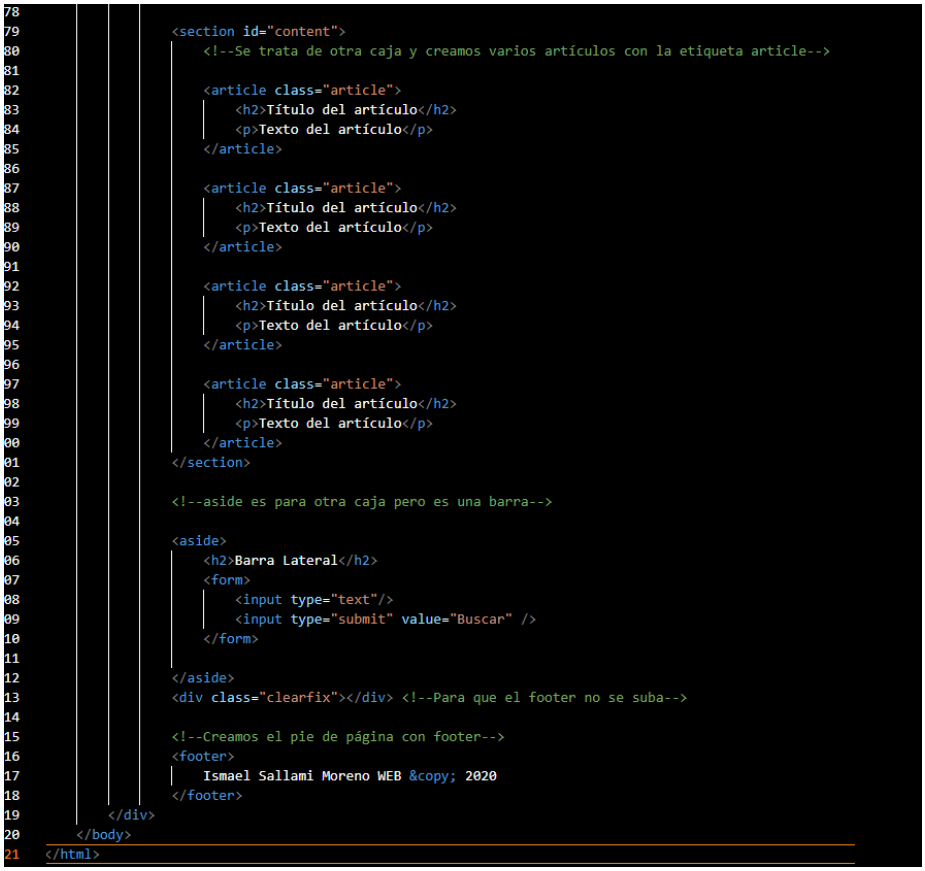


Figura 3.4: Primer ejemplo index.html

```css
/*el asterisco selecciona todos los elementos de la pag web*/
*{
 margin: 0px;
 padding: 0px;
 /*no hay márgenes en toda la web y padding tampoco, este es el margen dentro de una caja*/
}

 body{
    background: lightskyblue;
    font-family:Arial, Helvetica, sans-serif ;
    /*va por orden de preferencia y dependiendo de la máquina*/
 }

 #container{
    /*width: 70%;*/
    width: 1100px;
    margin: 0px auto;
    /*Centra el contenido*/
    border: 1px solid black;
 }

 header{
    /*porque solo tenemos una en la pág*/
    background: green;
    height: 100px;
    width: 100%;
    text-align: center;
    line-height: 100px; /*le pongo la misma altura que la caja porq asi lo que hace
    el texto es ponerse en el medio*/
    color: white;
    border-bottom: 3px dashed black;
    /*podemos poner bordes por cualquier lado este es por debajo y de manera discontinua
    de color negro*/



 }
 /*si creo mas elmentos header van a tener todos el mismo estilo, no se puede repetir
 el ID lo demás si*/
```

Figura 3.5: Primer ejemplo style.css

```css
nav ul li{
    /*los cojo todos*/

    float: left; /*se echen hacia la izq todos*/
    list-style: none;
    margin: 10px; /*no tengan estilo y se separen un poco*/
    line-height: 30px; /*centrar enlaces*/
}

nav{
    background: lightgrey;
    height: 50px;/*definir una altura para poder ver el fondo*/
    border-bottom:: 1px solid black
}

.clearfix{/*ponemos . para seleccionar una clase*/
    clear: both; /*cuando los floto hacia la izq todos los demás se quedan por debajo*/
}

#content{
    float: left;
    width: 80%;
    min-height: 500px;
    background: yellow;
    padding-left: 10px;
}

aside{
    float: left;
    width: calc(20% - 30px); /*usamos calc para que me lo ajuste bien y se refleje bien
    junto con los padding*/
    background: orange;
    min-height: 500px;
    padding: 10px; /*padding le da margenes interiores por todos lados, tmb puedo ponerlo
    por solo un lado con -lado que quiero*/
}
```

Figura 3.6: Primer ejemplo style.css

```css
footer{
    background: black;
    color: white;
    text-align: center;
    height: 50px;
    line-height: 50px;
}

.article{
    color:black;
    margin-top: 15px;
    margin-bottom: 15px;
    border-bottom: 1px solid lightblue; /*esto pone un gris muy clarito #eee*/
    padding: 10px;
}

.article h2{ /*seleccioname todos los h2 que esten dentro de article*/
    font-size: 30px;
}

.article:first-child{ /*cojo el primer artículo y le doy bordes por encima*/
    border-top: 1px solid lightblue;
    padding-top: 10px
}
```

Figura 3.7: Primer ejemplo style.css

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" type= "text/css" href="styles.css">
    <title>Aprendiendo Flexbox con Ismael Sallami</title>
</head>
<body>
    <h1>Aprendiendo con Ismael </h1>

    <section class="flex-container">

        <div class="caja">
            Caja 1
        </div>

        <div class="caja">
            Caja 2
        </div>

        <div class="caja">
            Caja 3
        </div>

        <div class="caja">
            Caja 4
        </div>

        <div class="caja">
            Caja 1
        </div>
```

Figura 3.8: index.html

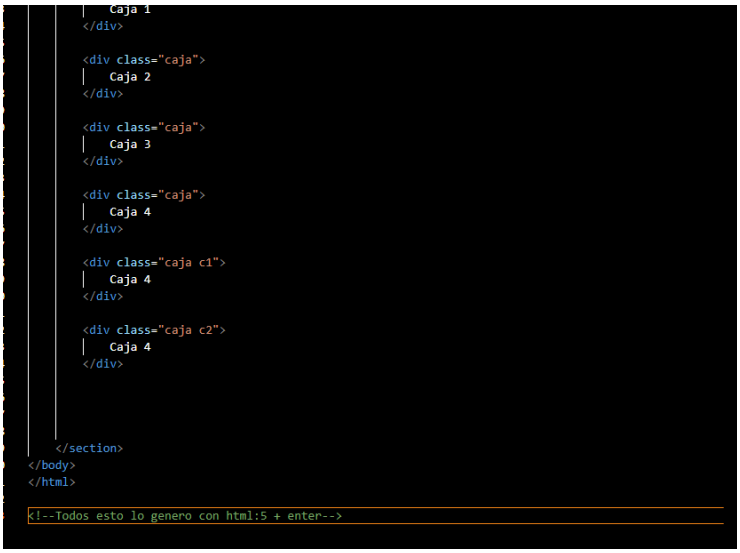
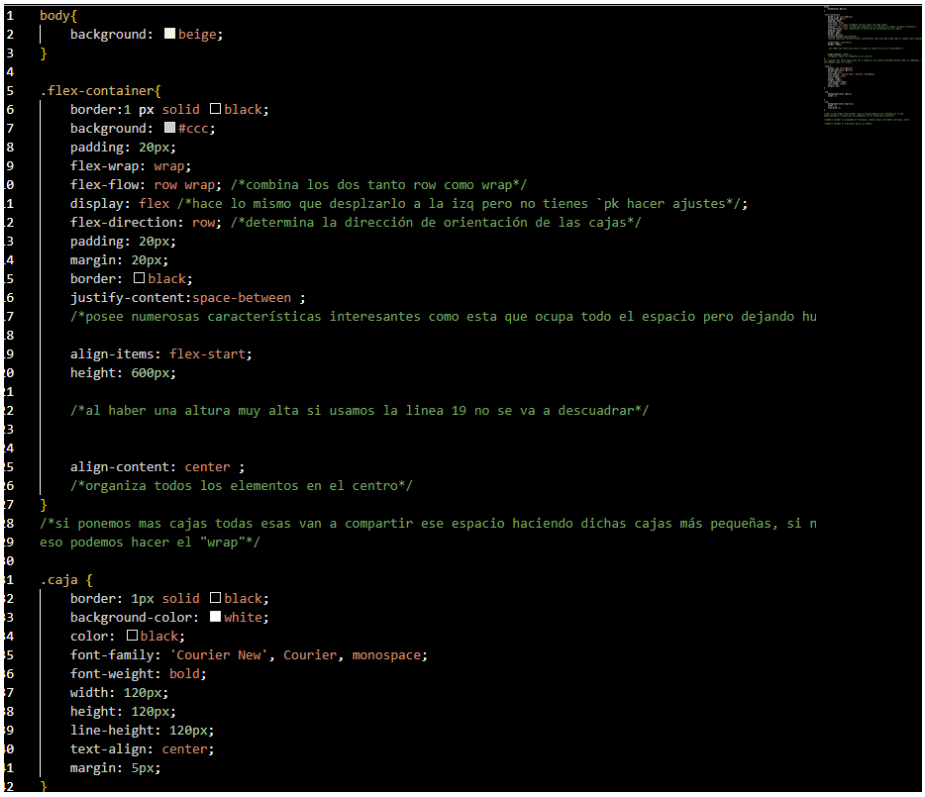


Figura 3.9: index.html



```css
body{
    background: beige;
}

.flex-container{
    border:1 px solid black;
    background: #ccc;
    padding: 20px;
    flex-wrap: wrap;
    flex-flow: row wrap; /*combina los dos tanto row como wrap*/
    display: flex /*hace lo mismo que desplzarlo a la izq pero no tienes `pk hacer ajustes*/;
    flex-direction: row; /*determina la dirección de orientación de las cajas*/
    padding: 20px;
    margin: 20px;
    border: black;
    justify-content:space-between ;
    /*posee numerosas características interesantes como esta que ocupa todo el espacio pero dejando hu

    align-items: flex-start;
    height: 600px;

    /*al haber una altura muy alta si usamos la linea 19 no se va a descuadrar*/


    align-content: center ;
    /*organiza todos los elementos en el centro*/
}
/*si ponemos mas cajas todas esas van a compartir ese espacio haciendo dichas cajas más pequeñas, si n
eso podemos hacer el "wrap"*/

.caja {
    border: 1px solid black;
    background-color: white;
    color: black;
    font-family: 'Courier New', Courier, monospace;
    font-weight: bold;
    width: 120px;
    height: 120px;
    line-height: 120px;
    text-align: center;
    margin: 5px;
}
```

Figura 3.10: style.css

```css
.c1{
    background-color: blue;
    order: 2;

}
.c2{
    background-color: yellow;
    order: 1;
    flex-grow: 4;
}

/*todo lo que tenga flex-content tiene la posibilidad de ser ordenado por lo que
puedo manipular y clasificar los elementos con el orden que yo quiera*/

/*también tenemos la propiedad de flex-grow, dándole mayor prioridad a uno que a otro*/

/*también tenemos el flex-basis que es un width*/
```

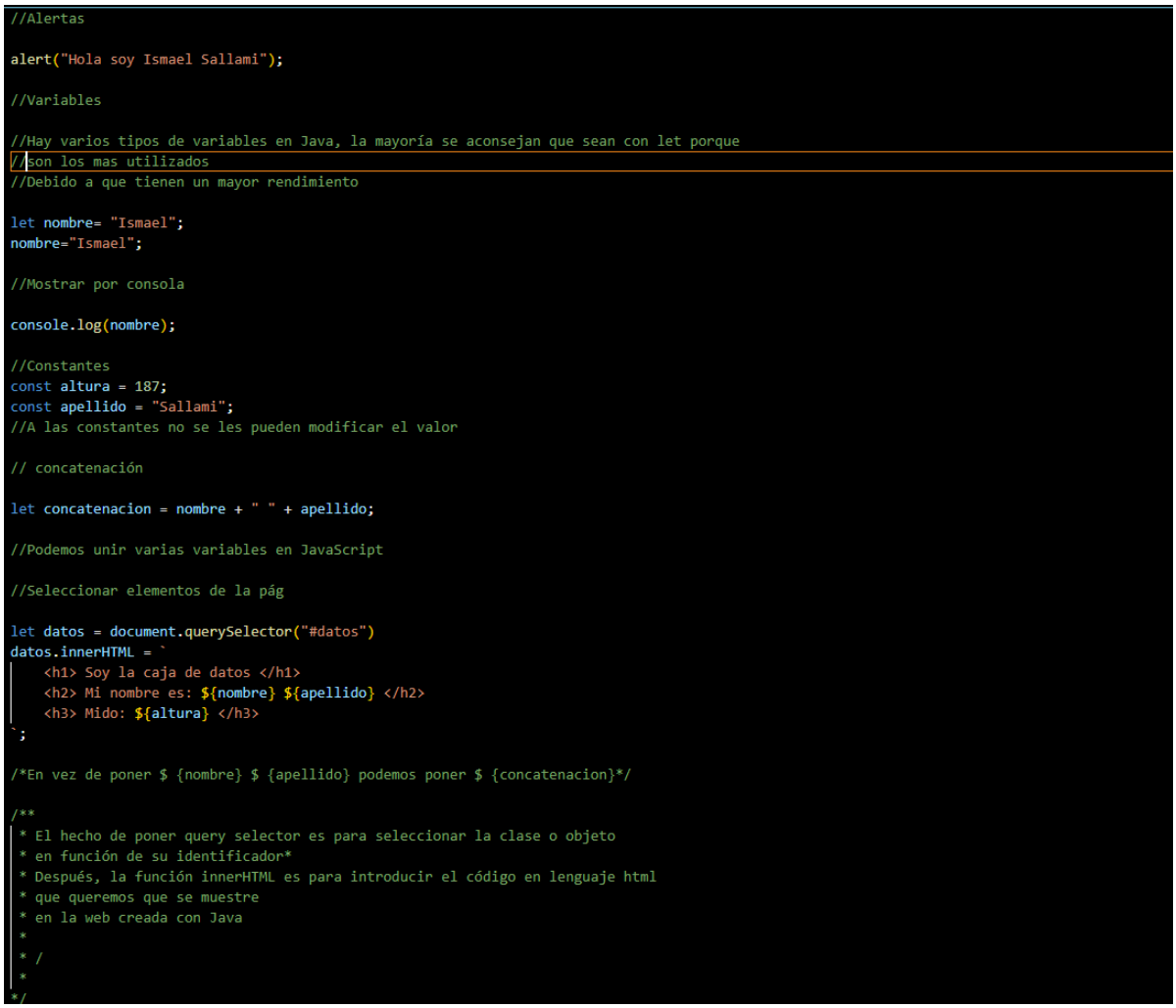
Figura 3.11: style.css

4 JavaScript

Todo lo que vamos a exponer viene explicado a la vez que se adjunta la foto del código, en el caso de que sea necesario aclarar algo lo haré a continuación.

4.1 main.js

4.2 index.html



```
//Alertas

alert("Hola soy Ismael Sallami");

//Variables

//Hay varios tipos de variables en Java, la mayoría se aconsejan que sean con let porque
//son los mas utilizados
//Debido a que tienen un mayor rendimiento

let nombre= "Ismael";
nombre="Ismael";

//Mostrar por consola

console.log(nombre);

//Constantes
const altura = 187;
const apellido = "Sallami";
//A las constantes no se les pueden modificar el valor

// concatenación

let concatenacion = nombre + " " + apellido;

//Podemos unir varias variables en JavaScript

//Seleccionar elementos de la pág

let datos = document.querySelector("#datos")
datos.innerHTML = `
    <h1> Soy la caja de datos </h1>
    <h2> Mi nombre es: ${nombre} ${apellido} </h2>
    <h3> Mido: ${altura} </h3>
`;

/*En vez de poner $ {nombre} $ {apellido} podemos poner $ {concatenacion}*/

/**
 * El hecho de poner query selector es para seleccionar la clase o objeto
 * en función de su identificador*
 * Después, la función innerHTML es para introducir el código en lenguaje html
 * que queremos que se muestre
 * en la web creada con Java
 *
 * /
 *
*/
```

Figura 4.1: main.js

```
*/
//Condiciones

let n=185;

if(n>=185){
    datos.innerHTML += "<h1> Eres una persona alta</h1>"
    //Concatena lo que antes había en innerHTML con dicha condición
}else{
    datos.innerHTML += "<h1>Eres una persona baja </h1>"
}

//Bucles

for(let year = 2000; year <= 2024; year++){
    //Similar al de c++
    datos.innerHTML  += "<h2>Estamos en el año: " + year + "</h2>";
    //en vez de sumar todo podemos poner `` y la variable con dolar y corchetes
}

//Array

let nombres = ["Ismael","Pepe"];

let divNombres = document.querySelector("#nombres");

//divNombres.innerHTML = nombres[0];

divNombres.innerHTML = "<h1>Listado de nombres</h1>";

nombres.forEach(nombre => {
    divNombres.innerHTML += "<li>"+nombre+"</li>";
});

/*Los foreach son similares  a los for pero con ligeras diferencias*/

//Esta es la manera en la que se recorren los de python

for(let nombre of nombres){
    divNombres.innerHTML += "<li>"+nombre+"</li>";
}
```

Figura 4.2: main.js

```
//Funciones

// const miInformacion= (nombre,altura) =>{
//     let misDatos =`
//         <h1> Soy la caja de datos </h1>
//         <h2> Mi nombre es: ${nombre} ${apellido} </h2>
//         <h3> Mido: ${altura} </h3>
//     `;

//     return misDatos;
// }

// console.log(miInformacion("Ismael", 182));

// //Esto no lo imprime de manera correcta, para ello debemos de realizar de la manera
// //correcta para poder verlo en nuestra pantalla

// const Imprimir = () => {
//     let datos = document.querySelector("#datos");
//     datos.innerHTML = miInformacion("Ismael Moreno",182);
// }

// Imprimir();

//Como podemos ver las funciones son similares a las de c++, varía en cuanto
//a la sintaxis pero de manera muy ligera


//Lo de arriba esta documentado para poder seguir probando otros métodos
```

Figura 4.3: main.js

```
//Objetos Json o los objetos literales

var coche = {
    modelo: 'Mercedes',
    maxima: 500,
    antiguedad: 2020,
    //Dentro también podemos crear funciones y métodos
    mostrarDatos(){
        console.log(this.modelo, this.maxima,this.antiguedad);
    },
    propiedad: "valor aleatorio"

};

document.open();
document.write("<h1>" + coche.modelo + "</h1>");
document.close();

coche.mostrarDatos();

//Tambien para ver lo que tiene podemos hacer, pero de esta manera muestra más datos:

console.log(coche)
```

Figura 4.4: main.js

```
//PROMESAS

//Se trata de un valor que puede estar disponible ahora, en el futuro o nunca.

//Con esto podemos capturar cuando la petición de un servicio es aceptado o rechazada,
//o multitud de cosas más

var saludar = new Promise ((resolve, reject) => {
    setTimeout(() => {
        let saludo = "Hola muy buenas a todos chavales!!!";
        //si saludo es false no devuelve nada
        if (saludo){
            resolve(saludo)
            //me muestre saludo si lo hay
        }
        else {
            reject('No hay saludo disponible')
        }
    }, 2000);
    //Se espera unos segundos a que algo se ejecute, en este caso ponemos 2 seg como 2000
});

saludar.then(resultado => {
    alert(resultado);
    //muestro resultado porque dentro tengo el saludo
})
.catch(err => {
    alert(err); //Err es lo que devuelve reject
});

//Básicamente en esta parte se muestra como tratar una alerta, es decir, que nos salga una ventana.
//transcurrido un pequeño tiempo, en este caso 2 seg
```

Figura 4.5: main.js

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!--Para que lo que escribamos ebn el archivo de java nos aparezca aquí, debemos de
    enlazarlo con el siguiente comando-->
    <script src="main.js" defer></script>
    <!--La etiqueta defer hace que se espere a todas las actualizaciones de nuestra pag de index.html
    y no de erroreds ya que sin esa etiqueta los cambios no se cargan al momento-->
    <title>Aprender JavaScript con Ismael Sallami</title>
</head>
<body>
    <h1>Aprendiendo con Ismael Sallami</h1>
    <div id="datos"></div>
    <div id="nombres"></div>
</body>
</html>
```

Figura 4.6: index.html