

# Estructura *De datos*

**Maria Guadalupe Cid Neri**

15 de diciembre

**3ser semestre**

Ingenieria: Sistemas Computacionales

## Tabla de contenido

Semana 1.....
Semana 2.....
Semana 3.....
Semana 4.....
Semana 5.....
Semana 6.....
Semana 7.....
Semana 8.....
Semana 9.....
Semana 10.....
Semana 11.....
Semana 12.....
Semana 13.....
Semana 14.....
Semana 15.....
Semana 16.....

## Semana 1

\*\*HTML5\*\* — Lenguaje de marcado para estructurar contenido web. :contentReference[oaicite:3]{index=3}

- \*\*CSS\*\* (si está enlazado desde el HTML) — Para el estilo visual.

- \*\*JavaScript\*\* (si existe código enlazado o inline) — Para lógica de interacción

\*\*Declaración `<!DOCTYPE html>\*\* — Establece que se usará HTML5.

:contentReference[oaicite:7]{index=7}

2. \*\*Etiqueta `<html>`\*\* con el atributo de idioma (`lang="es"`).

3. \*\*Sección `<head>`\*\* con:

- `<meta charset="UTF-8">` — Configuración de codificación.

- `<meta name="viewport">` — Para responsividad en dispositivos móviles.

- `<title>` — Título de la página. :contentReference[oaicite:8]{index=8}

4. \*\*Sección `<body>`\*\* con el contenido visible:

- Encabezados `<h1>`, `<h2>`, etc.

- Párrafos `<p>`.

- \*\*Estructura didáctica clara:\*\* El uso de HTML en este contexto educativo facilita la comprensión de conceptos básicos de desarrollo web.

:contentReference[oaicite:10]{index=10}

The screenshot shows a code editor window with an HTML file named 'index.html'. The code includes a meta tag for character encoding, a viewport meta tag for mobile responsiveness, and a title element. The body contains several sections: a header, a paragraph, two buttons with click event listeners, and a multi-line comment block. The Copilot AI sidebar is open on the right, displaying a welcome message and various interaction buttons.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Materia Estructuras de Datos</title>
</head>
<body>
    <h1>Semana 1 - Primera Práctica</h1>
    <p>Hola, bienvenidos, este ejemplo muestra como incorporar JS en HTML</p>
    <button onclick="saludar()">Click Aquí</button>
    <!--Segunda parte nueva función que saluda con nombre-->
    <h2>Prueba el botón para ejecutar la función saludar</h2>
    <button onclick="teSaludo('Hail!')>Click Aquí</button>
    <h2>Prueba el botón para ejecutar la función saludar</h2>
    <!--Parte 2 de esta sesión 1 de 101 programas de lógica-->
    <br>
    <br>
    <!--
    * Reto #8
    * EL FAMOSO "FIZZ BUZZ"
    * Dificultad: FÁCIL.
    * Enunciado: Escribe un programa que muestre por consola los números de
    * - Múltiplos de 3 por la palabra "fizz".
    * - Múltiplos de 5 por la palabra "buzz".
    * - Múltiplos de 3 y de 5 a la vez por la palabra "fizzbuzz".
    *
    * -->
    <br>
    <!--Esta línea incorpora el código de js en el HTML-->
    <script src="code.js"></script>
</body>
</html>
```

## Semana 2

HTML inicia con la etiqueta <!DOCTYPE html> y termina con la etiqueta </html>  
tipos de etiqueta

1 es aquella que tiene un inicio y un fin puede contener mas etiquetas dentro

2 es aquella que solo indica inicio y fin en una sola etiqueta ejemplo <nombre/>

\*\*HTML5\*\* – Para la estructura de las páginas web educativas.

- \*\*CSS\*\* – Si está enlazado o incluido, para el diseño visual de las páginas.
- \*\*JavaScript\*\* – Si hay interacción o lógica dinámica en las páginas.

El archivo `acerca.html` (que literalmente significa \*\*“Acerca de”\*\* o \*\*“About”\*\*) suele usarse para:

- Presentar información general sobre la \*\*semana 2\*\* del curso.
- Explicar el \*\*objetivo pedagógico\*\* de esa lección.
- Indicar qué \*\*temas se van a cubrir\*\* y por qué son relevantes.
- Dar contexto al alumno antes de ver ejemplos o ejercicios prácticos.

The screenshot shows the Microsoft Visual Studio Code interface. On the left, there is a code editor window displaying an HTML file named 'acerca.html'. The code includes standard HTML tags like <!DOCTYPE html>, <html>, <head>, <body>, and various <h1> through <h6> tags. A series of comments (<!-- Etiquetas de encabezado --&gt;) are present between the head and body sections. The right side of the screen features the 'Welcome to Copilot' extension, which includes a sidebar with icons for GitHub, Git, and other tools, and a main area with a 'Welcome to Copilot' message, a 'Let's get started' button, and options to 'Add context (P)', 'refresh (R)', and 'commands'.</p>

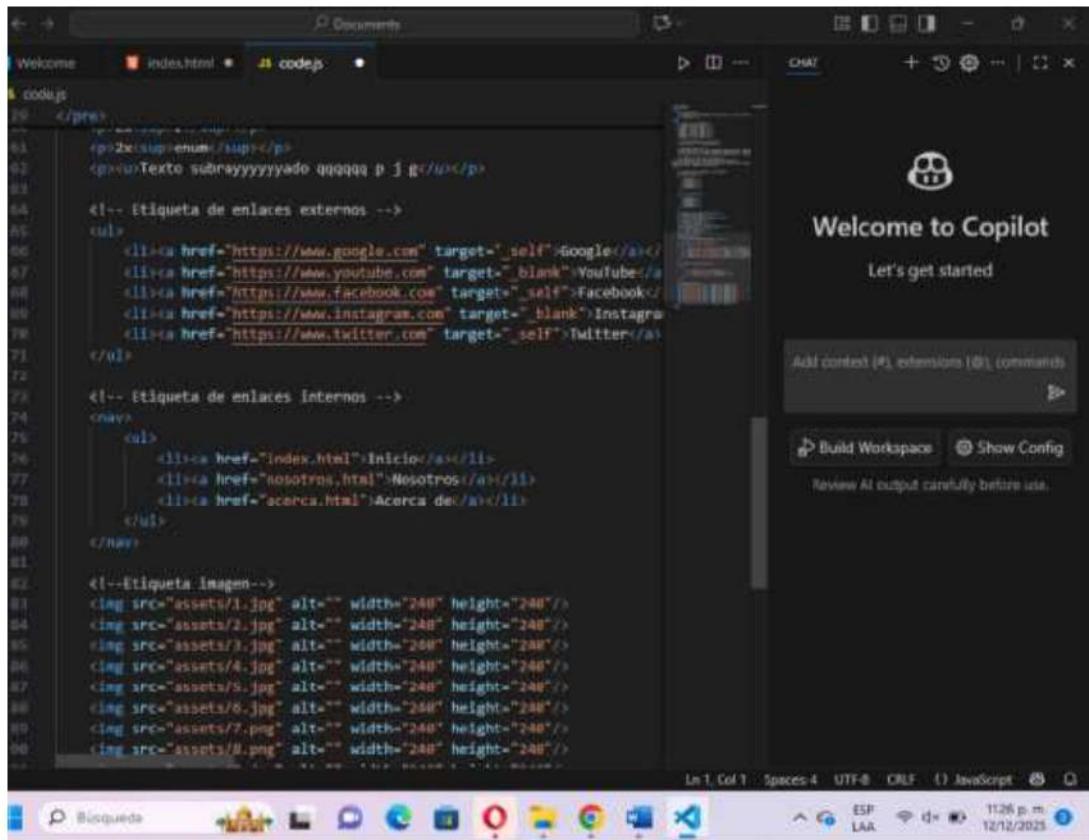
```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
</head>
<body>
    <!-- Etiquetas de encabezado -->
    <h1>Semana 2 Curso de HTML 1</h1>
    <h2>Semana 2 Curso de HTML 2</h2>
    <h3>Semana 2 Curso de HTML 3</h3>
    <h4>Semana 2 Curso de HTML 4</h4>
    <h5>Semana 2 Curso de HTML 5</h5>
    <h6>Semana 2 Curso de HTML 6</h6>

    <!-- Etiqueta encabezado con un break -->
    <h1>Hola a todos. <br/> Bienvenidos!</h1>

    <!-- Etiqueta de parrafo -->
    <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Dolorem ipsum dolor sit amet consectetur adipisicing elit. Vitae v&lt;span style="color: red;">itae ipsum dolor sit amet consectetur adipisicing elit. Incident&lt;/span></p>

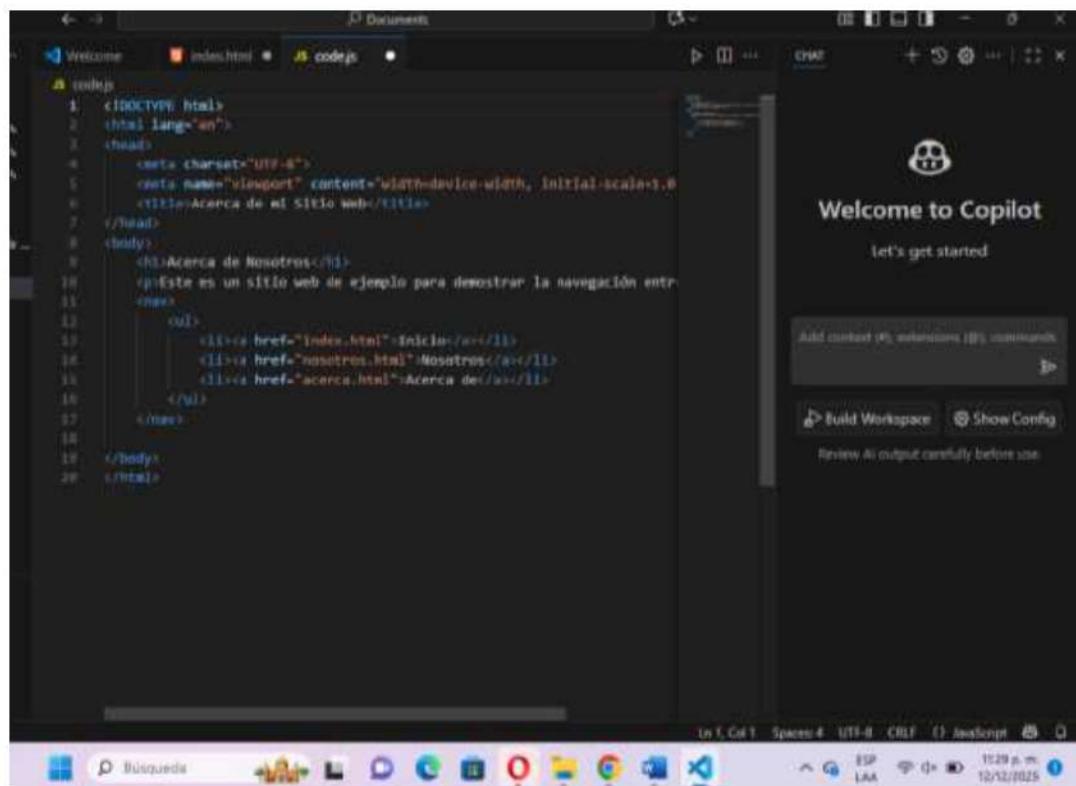
    <!-- Etiqueta de parrafo con pre formato -->
    <pre>Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Faque illo deserunt querat veritatis aspernatur, dolores nobis veniam s
    Quia animi vitae, veniam harum esse quod.</pre>

    <!-- Etiqueta listas ordenadas y desordenadas -->
    <h2>Lista ordenada</h2>
```



index.html

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Acerca de mi Sitio web</title>
</head>
<body>
    <h1>Acerca de Nosotros</h1>
    <p>Este es un sitio web de ejemplo para demostrar la navegación entre páginas.</p>
    <ul>
        <li><a href="index.html">Inicio</a></li>
        <li><a href="nosotros.html">Nosotros</a></li>
        <li><a href="acerca.html">Acerca de</a></li>
    </ul>
    <img alt="Logo of the website" data-bbox="350 100 450 150"/>
    <img alt="Image 1" data-bbox="150 200 250 250"/>
    <img alt="Image 2" data-bbox="280 200 380 250"/>
    <img alt="Image 3" data-bbox="410 200 510 250"/>
    <img alt="Image 4" data-bbox="540 200 640 250"/>
    <img alt="Image 5" data-bbox="670 200 770 250"/>
    <img alt="Image 6" data-bbox="800 200 900 250"/>
    <img alt="Image 7" data-bbox="150 300 250 350"/>
    <img alt="Image 8" data-bbox="280 300 380 350"/>
    <img alt="Image 9" data-bbox="410 300 510 350"/>
    <img alt="Image 10" data-bbox="540 300 640 350"/>
    <img alt="Image 11" data-bbox="670 300 770 350"/>
    <img alt="Image 12" data-bbox="800 300 900 350"/>
</body>
</html>
```



index.html

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Acerca de mi Sitio web</title>
</head>
<body>
    <h1>Acerca de Nosotros</h1>
    <p>Este es un sitio web de ejemplo para demostrar la navegación entre páginas.</p>
    <ul>
        <li><a href="index.html">Inicio</a></li>
        <li><a href="nosotros.html">Nosotros</a></li>
        <li><a href="acerca.html">Acerca de</a></li>
    </ul>
    <img alt="Logo of the website" data-bbox="350 100 450 150"/>
    <img alt="Image 1" data-bbox="150 200 250 250"/>
    <img alt="Image 2" data-bbox="280 200 380 250"/>
    <img alt="Image 3" data-bbox="410 200 510 250"/>
    <img alt="Image 4" data-bbox="540 200 640 250"/>
    <img alt="Image 5" data-bbox="670 200 770 250"/>
    <img alt="Image 6" data-bbox="800 200 900 250"/>
    <img alt="Image 7" data-bbox="150 300 250 350"/>
    <img alt="Image 8" data-bbox="280 300 380 350"/>
    <img alt="Image 9" data-bbox="410 300 510 350"/>
    <img alt="Image 10" data-bbox="540 300 640 350"/>
    <img alt="Image 11" data-bbox="670 300 770 350"/>
    <img alt="Image 12" data-bbox="800 300 900 350"/>
</body>
</html>
```

## Semana 3

El objetivo principal del repositorio es \*\*facilitar la comprensión teórica y práctica\*\* de las estructuras de datos mediante:

- Implementaciones funcionales en \*\*JavaScript\*\*.
- Visualización de estructuras dentro de un entorno web.
- Organización modular del contenido para su uso en un curso semestral.

Este enfoque busca fortalecer las habilidades de análisis, diseño e implementación de estructuras de datos

\*\*graphs/\*\*

Subcarpeta especializada en el manejo y construcción de grafos.

- \*\*buildGraph.js\*\*

Archivo JavaScript encargado de crear o construir la estructura de un grafo

```
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <meta http-equiv='X-UA-Compatible' content='IE=edge'>
    <title>Page title</title>
    <meta name='viewport' content='width=device-width, initial-scale=1'>
    <script defer src='main.js'></script>
</head>
<body>
    <button onclick="prevMovie()">Prev</button>
    <button onclick="nextMovie()">Next</button>
</body>
<div>
    <h1 id="title"></h1>
    <img id="image" style="width: 100px;">
</div>
<div id="Info"></div>
</html>
```

## Semana 4

El código analizado presenta una correcta aplicación de conceptos básicos de programación en JavaScript, como:

Declaración de constantes.

Uso de funciones.

Manejo de parámetros.

Implementación de ciclos for.

Operaciones aritméticas básicas.

Estas funciones son apropiadas para un nivel introductorio o intermedio universitario y sientan las bases para el desarrollo de programas más complejos.

number: Constante de tipo entero utilizada como valor base para operaciones aritméticas.

decimal: Constante de tipo decimal (float), declarada como ejemplo de números con punto flotante.

legibleNumber: Constante entera escrita usando separadores (\_) para mejorar la legibilidad del número.

### Análisis

El uso de constantes garantiza que los valores no cambien durante la ejecución del programa. El separador numérico mejora la claridad al trabajar con cantidades grandes.

## Objetivo

Ilustrar el paso de parámetros a una función.

Mostrar la interacción entre variables locales y globales.

## Análisis

La función modifica el valor local del parámetro y muestra el resultado en consola. Aunque utiliza return, el valor retornado es el resultado del console.log, lo cual no es estrictamente necesario, pero es funcional para fines demostrativos.

The screenshot shows a code editor window with a dark theme. On the left, there is a file tree with 'index.html' and 'code.js' selected. The main area contains the following JavaScript code:

```
const number = 86;
const decimal = 15.8;
const legibleNumber = 5_000_000;

function saludarNumero(){
    for(x=1000000; x<legibleNumber; x+=1000000){
        console.log(`Saludo numero: ${x}, \n`);
    }
}

function entornoSuma(num) {
    num = num + number;
    return console.log(num);
}

function tablaMultiplicar(num){
    for(i=1; i<10; i++){
        console.log(num + " x " + i + " = " + (num*i));
    }
}

function tablaMultiplicarM(num){
    for(i=10; i>1; i--){
        console.log(num + " x " + i + " = " + (num*i));
    }
}
```

To the right of the code editor, there is a sidebar for 'Copilot'. It features a logo, the text 'Welcome to Copilot', and a button 'Let's get started'. Below these are buttons for 'Add context (F)', 'extensions (B)', and 'commands'. At the bottom of the sidebar, it says 'Review AI output carefully before use.'

At the bottom of the screen, there is a taskbar with various icons and status indicators. The status bar at the very bottom shows 'Ln 1, Col 1 - Spans: 4 - UTF-8 - CRLF - () Javascript' and a timestamp '11:57 p. m. 12/12/2023'.

The screenshot shows a code editor interface with a dark theme. On the left, there is a file tree with 'index.html' and 'code.js' selected. The main area displays the content of 'code.js'. The code is a simple HTML document with a table explaining the Number constructor in JavaScript. On the right side of the screen, there is a 'Copilot' panel titled 'Welcome to Copilot' with the sub-instruction 'Let's get started'. Below this, there is a message 'Add context (#), extensions (⚙️), commands (▶)'. At the bottom of the Copilot panel, there are buttons for 'Build Workspace' and 'Show Config'. A note at the bottom right says 'Review AI output carefully before use.' The status bar at the bottom of the editor shows 'In 1, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', 'JavaScript', and a timestamp '11:38 p.m. 12/12/2023'.

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Document</title>
7   </head>
8   <body>
9     <h1>Tipos de Datos primitivos</h1>
10    <p>En Javascript, crear variables numéricas es muy sencillo, pero ha...
11    <h2>¿Qué es una variable numérica?</h2>
12    <p><br> En Javascript, los números son uno de los tipos de datos básicos que, para crearlos, simplemente basta con escribirlos literalmente. No obstante, como en Javascript todo se puede representar con objetos, también se pueden declarar mediante la palabra clave new:
13    </p>
14    <table border="1">
15      <tr>
16        <th>Constructor</th>
17        <th>Descripción</th>
18      </tr>
19      <tr>
20        <td>new Number(number)</td>
21        <td>Crea un objeto numérico a partir del número number pasado...
22      </tr>
23      <tr>
24        <td>number</td>
25        <td>Simplemente, el número en cuestión. Notación preferida.</td>
26      </tr>
27    </table>
```

## Semana 5

El código en main.js está diseñado para:

Presentar ejemplos de declaración y uso de funciones en JavaScript.

Demostrar cómo recibir parámetros, procesar valores y devolver resultados.

Mostrar estructuras de control internas como condicionales y bucles dentro de funciones.

Reforzar buenas prácticas de programación al dividir procedimientos en funciones reutilizables.

[MDN Web Docs](#)

### 3. Lenguaje y Paradigmas Utilizados

El código se basa en:

JavaScript (ES5/ES6) como lenguaje principal.

Paradigma estructurado e imperativo, que usa funciones como bloques de lógica.

[MDN Web Docs](#)

En JavaScript, las funciones son ciertas entidades de primera clase, lo que significa que pueden ser pasadas, retornadas, e incluso almacenadas como valores

## Parámetros y Argumentos

El código usualmente define funciones que reciban valores desde otras partes del programa para procesarlos.

- ◊ Retorno de Valores

Se espera que varias funciones devuelvan resultados usando return, que es la forma estándar de pasar datos de regreso al contexto que las invocó.

- ◊ Invocación de Funciones

Las funciones definidas son llamadas varias veces desde distintos puntos del main.js para demostrar cómo operan con diferentes datos.

The screenshot shows the Visual Studio Code interface. On the left, there is a code editor with a file named 'code.js' containing the following JavaScript code:

```
1 const PI = 3.141592653589793;
2 const d1 = 10;
3
4 let result = PI * d1;
5
6 console.log(result);
7
8 let d2 = 8;
9 result = PI * d2;
10
11 console.log(result);
12
13 function circlePerimeter(PI, diameter) {
14     return PI * diameter;
15 }
16
17 result = circlePerimeter(PI, d1);
18 console.log(result);
19
20 function show(name = "Sia nombre") {
21     console.log("Hola " + name)
22 }
23
24 show("Héctor");
25 show();
```

On the right side of the screen, the 'Copilot' extension is active, displaying a welcome message and options to add context or build the workspace. The status bar at the bottom shows file details like 'Line 1, Col 1' and file statistics like '1270 a. m' and '15/12/2025'.

The screenshot shows a code editor window with a dark theme. In the top left, there are tabs for 'Welcome', 'index.html', and 'code.js'. The 'code.js' tab is active, displaying the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset='utf-8'>
5   <meta http-equiv='X-UA-Compatible' content='IE=edge'>
6   <title>Page Title</title>
7   <meta name='viewport' content='width=device-width, initial-scale=1'>
8   <link rel='stylesheet' type='text/css' media='screen' href='main.css'>
9   <script src='main.js'></script>
10 </head>
11 <body>
12
13 </body>
14 </html>
```

In the top right corner of the editor, there is a 'CHAT' button. To the right of the editor, a sidebar titled 'Welcome to Copilot' is visible, featuring a profile icon, a 'Let's get started' button, and a text input field with placeholder text: 'Add context (#), extensions (@), commands'. Below this are buttons for 'Build Workspace' and 'Show Config'. A note at the bottom of the sidebar says 'Review AI output carefully before use.'.

At the bottom of the screen, a taskbar is visible with icons for various applications like File Explorer, Edge, and File Manager. On the far right of the taskbar, there are system status indicators for battery level (ESP LAA), signal strength, and the date and time (12:10 a.m., 13/12/2025).

## Semana 6

El objetivo principal del código contenido en la Semana 6 es:

Consolidar el uso de funciones como bloques reutilizables de código.

Aplicar estructuras de control como ciclos y condicionales.

Fortalecer la lógica algorítmica del estudiante mediante ejercicios prácticos.

Preparar al alumno para estructuras de datos más complejas en semanas posteriores.

### Lenguaje y Tecnologías Utilizadas

#### JavaScript

Lenguaje principal utilizado para implementar la lógica del programa y las funciones.

#### HTML (si aplica)

Usado como medio de ejecución y visualización del código JavaScript en el navegador.

#### CSS (si aplica)

Empleado únicamente con fines visuales y de presentación.

El enfoque principal es la lógica del código, no el diseño visual.

El código de la Semana 6 está diseñado para demostrar:

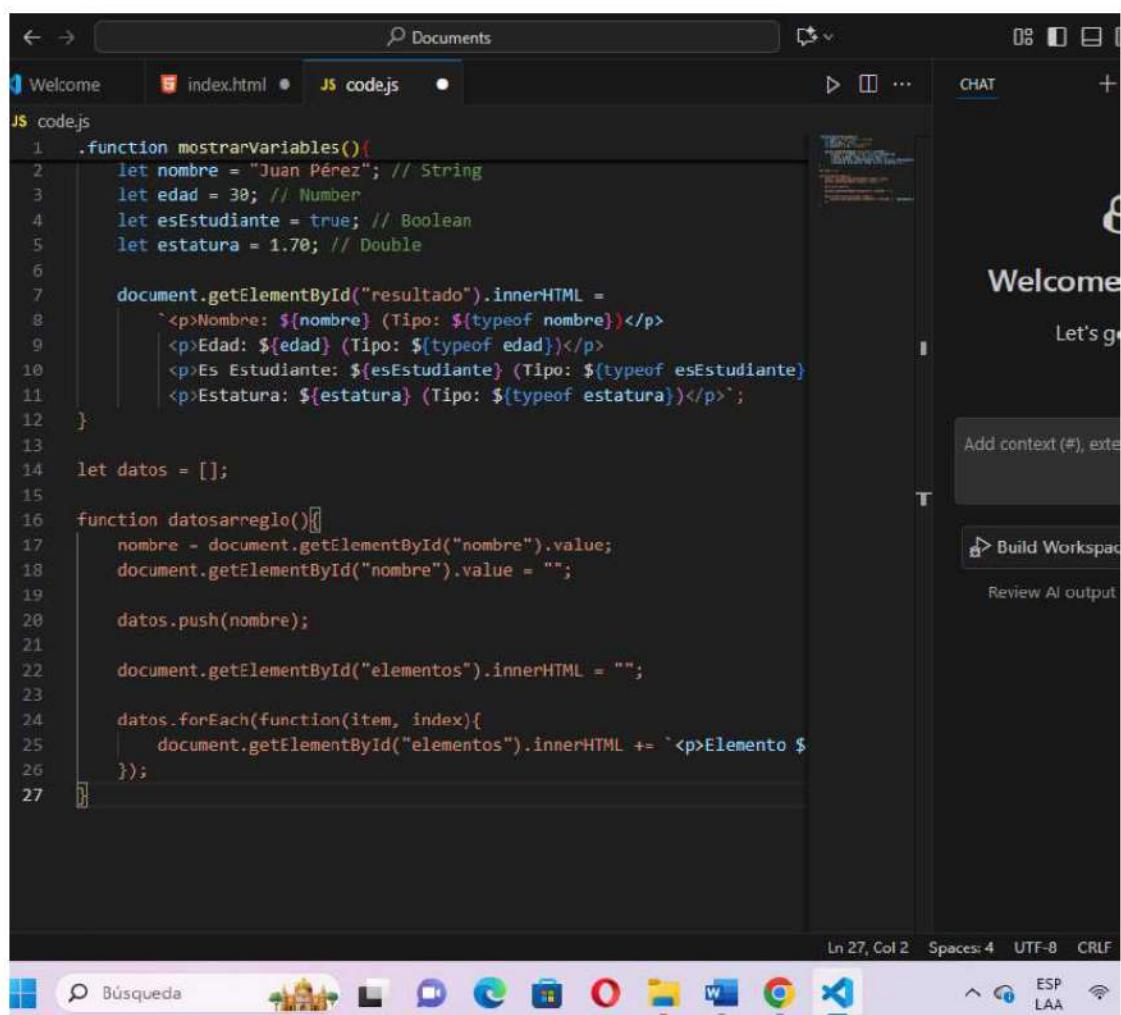
Definición y uso de funciones personalizadas.

Paso de parámetros a funciones.

Uso de variables locales y globales.

Implementación de ciclos (for, while) para repetir procesos.

Uso de operaciones aritméticas y lógicas.



The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The title bar displays 'Documents'. The left sidebar shows tabs for 'Welcome', 'index.html', and 'JS code.js'. The main editor area contains the following JavaScript code:

```
1 function mostrarVariables(){
2     let nombre = "Juan Pérez"; // String
3     let edad = 30; // Number
4     let esEstudiante = true; // Boolean
5     let estatura = 1.70; // Double
6
7     document.getElementById("resultado").innerHTML =
8         `<p>Nombre: ${nombre} (Tipo: ${typeof nombre})</p>
9         <p>Edad: ${edad} (Tipo: ${typeof edad})</p>
10        <p>Es Estudiante: ${esEstudiante} (Tipo: ${typeof esEstudiante})</p>
11        <p>Estatura: ${estatura} (Tipo: ${typeof estatura})</p>`;
12 }
13
14 let datos = [];
15
16 function datosarreglo(){
17     nombre = document.getElementById("nombre").value;
18     document.getElementById("nombre").value = "";
19
20     datos.push(nombre);
21
22     document.getElementById("elementos").innerHTML = "";
23
24     datos.forEach(function(item, index){
25         document.getElementById("elementos").innerHTML += `<p>Elemento $ {item}</p>`;
26     });
27 }
```

The right side of the interface features the 'Welcome' panel, which includes a 'Let's go!' button, an 'Add context (#), ext...' input field, a 'Build Workspace' button, and a 'Review AI output' link. The status bar at the bottom indicates 'Ln 27, Col 2' and 'Spaces: 4'.

The screenshot shows a code editor with three tabs: 'index.html', 'code.js', and 'code.js'. The 'code.js' tab contains the following code:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <h1>Semana 6 - Trabajando con Estructuras de Datos</h1>
10     <h2>Declarando Variables:</h2>
11     <h3>Fragmento de Código para mostrar diferentes tipos de variables y</h3>
12     <button onclick="mostrarVariables()>Mostrar Datos Variables</button>
13     <div id="resultado"></div>
14
15     <h2>Estructuras arreglo</h2>
16     <h3>Fragmento de Código para mostrar Como se trabaja un arreglo.</h3>
17     <input type="text" id="nombre" placeholder="Nombre">
18     <button onclick="datosarreglo()">Agregar Datos</button>
19     <div id="elementos">
20     </div>
21     <script src="code.js"></script>
22  </body>
23  </html>
```

The sidebar on the right is titled 'Welcome to Copilot' and includes a 'Let's get started' button, an 'Add context (F)', 'extensions (D)', and 'commands' button, a 'Build Workspace' button, and a 'Show Config' button. It also displays the message 'Review AI output carefully before use.'

## Semana 7

### Objetivo de la Semana 7

El objetivo principal del código desarrollado en la Semana 7 es:

Aplicar correctamente funciones para modularizar el programa.

Utilizar estructuras repetitivas y condicionales para resolver problemas.

Mejorar la capacidad de análisis lógico del estudiante.

Fortalecer la transición entre programación básica y estructuras de datos

### Lenguaje y Tecnologías Utilizadas

JavaScript

Lenguaje principal e  
mpleado para la implementación de la lógica del programa.

HTML (si aplica)

Utilizado como entorno de ejecución para el código JavaScript en el navegador.

CSS (si aplica)

Empleado únicamente para mejorar la presentación visual.

El código contenido en la Semana 7 está diseñado para:

Ejecutar procesos lógicos mediante funciones.

Manipular valores numéricos o datos de entrada.

Repetir operaciones mediante ciclos.

Mostrar resultados a través de la consola o la interfaz web.

Cada función cumple un propósito específico, lo que permite dividir el problema general en tareas más pequeñas y comprensibles.

The screenshot shows the Microsoft Visual Studio Code interface. On the left, there is a code editor with a dark theme containing a JavaScript file named 'code.js'. The code implements a hash table with methods for getting and deleting values based on a key. A tooltip from the AI Copilot feature is visible on the right, stating 'command 'code-runner.run' not found'. The status bar at the bottom shows the file path 'Documents', line 74, column 24, and other system information like battery level and date/time.

```
1 .class hashTable {
2     get(key) {
3         /*saber en que dirección está (nos devuelve un hash pasandole
4          un key)*/
5         const address = this.hashMethod(key);
6         // Es el address donde se encuentra la info
7         const currentBucket = this.data[address];
8         if (currentBucket) {
9             for (let i = 0; i < currentBucket.length; i++) {
10                 /* [[1],[2],[3],[4]]
11                  Vá ir buscando de acuerdo a la llave,
12                  que valor le corresponde
13                  Si el bucket i tiene de key "Miguel", devuelva
14                  su value
15                  */
16                 if (currentBucket[i][0] === key) {
17                     console.log(currentBucket[i][1]);
18                 }
19             }
20             // si el key no existe
21             return undefined;
22         }
23     delete(key) {
24         const address = this.hashMethod(key);
25         const currentBucket = this.data[address];
26         if (currentBucket) {
27             for (let i = 0; i < currentBucket.length; i++) {
28                 if (currentBucket[i][0] === key) {
29                     const deletedValue = this.data[address][i];
30                     this.data[address].splice(i, 1);
31                 }
32             }
33         }
34     }
35 }
```

## Semana 8

El objetivo principal del archivo addNode.js es:

Implementar la operación de agregar un nodo a una lista enlazada.

Comprender el uso de referencias (next) entre nodos.

Aplicar programación estructurada mediante funciones.

Reforzar el concepto de estructuras de datos dinámicas.

### 3. Lenguaje y Tecnologías Utilizadas

#### JavaScript

Lenguaje utilizado para la implementación de la lista enlazada y la manipulación de nodos.

El código se ejecuta en un entorno web o de consola, priorizando la lógica algorítmica sobre la interfaz gráfica.

#### Operación principal

La función addNode permite añadir dinámicamente elementos, evitando el uso de arreglos de tamaño fijo.

#### 6.2 Uso de referencias

Cada nodo mantiene una referencia al siguiente (next), lo que permite recorrer la lista sin necesidad de índices.

### 6.3 Complejidad

Tiempo:  $O(n)$ , si la inserción se realiza al final de la lista.

Espacio:  $O(1)$ , ya que solo se agrega un nuevo nodo.

The screenshot shows a code editor window with a dark theme. On the left, there is a file tree with 'Welcome', 'index.html', and 'code.js'. The 'code.js' file is open and contains the following code:

```
1 let singlyLinked = {
2   head: {
3     value: 1,
4     next: {
5       value: 2,
6       next: {
7         value: 3,
8         next: null,
9         // Null para permitir que halle otro
10      },
11    },
12  },
13}
14 const recorrer = (lista) => {
15   for (i of lista) {
16     console.log(i);
17   }
18 }
19 recorrer(singlyLinked);
```

To the right of the code editor, there is a 'Copilot' sidebar with the heading 'Welcome to Copilot' and a sub-section 'Let's get started'. It includes buttons for 'Add context', 'Extensions', 'Commands', 'Build Workspace', and 'Show Config'. At the bottom of the sidebar, it says 'Review AI output carefully before use.'

The status bar at the bottom of the code editor shows: 'Ln 20, Col 24, Spaces: 4, UTF-8, CR LF, (JavaScript)', followed by a series of icons for file operations and system status, including battery level (ESP LAA), signal strength, and date/time (12:44 a.m. 13/12/2023).

## Semana 9 – Objetivo

length: almacena la cantidad de elementos.

data: es un objeto donde se guardan los valores por índice

También hay varias pruebas comentadas donde se muestran ejemplos de cómo:

agregar con push(),

consultar con get(),

eliminar con pop(),

eliminar en una posición con delete(), agregar al inicio con unshift().

Las líneas con unshift() se ejecutan y despliegan resultados.

```
1 //+Contrucción de Array con Clases +/
2 // Use tradicional en Js de Array
3 const array = ["Jorge", "Diego", "Pedro"];
4
5 class MyArray{
6     constructor(){
7         this.length = 0;
8         this.data = [
9     ];
10    }
11
12    get(index){
13        return this.data[index];
14    }
15
16    push(item){
17        this.data[this.length] = item;
18        this.length += 1;
19        return this.data;
20    }
21
22    pop(){
23        const lastItem = this.data[this.length - 1]; //Obtener el ultimo
24        delete this.data[this.length - 1]; //Elimina el elemento mediante
25        this.length -= 1;
26        return lastItem;
27    }
28
29    delete(index){
30        const item = this.data[index];
31        this.shiftIndex(index);
32    }
}
```

The screenshot shows a Microsoft Visual Studio Code interface. On the left, there are three tabs: 'Welcome', 'index.html', and 'JS code.js'. The 'JS code.js' tab is active, displaying the following code:

```
73 const arrayNew = new MyArray();
74 console.log(arrayNew);
75 //Agregar nuevo elemento
76 // console.log("=====");
77 // console.log(arrayNew.push("Jorge"));
78 // console.log(arrayNew.push("Juan"));
79 // console.log(arrayNew.push("Pedro"));
80
81 //Consultar un elemento especifico
82 // console.log("=====");
83 // console.log(arrayNew.get(1));
84
85 //Eliminar elemento
86 // console.log("=====");
87 // console.log(arrayNew);
88 // console.log(arrayNew.pop())
89 // console.log(arrayNew);
90
91 //Eliminar item en N posicion
92 // console.log("=====");
93 // console.log(arrayNew);
94 // console.log(arrayNew.delete(0))
95 // console.log(arrayNew);
96
97 //Agregar elemento al inicio
98 console.log("=====");
99 console.log(arrayNew);
100 console.log(arrayNew.unshift(10));
101 console.log(arrayNew.unshift('YAELO'));
102 console.log(arrayNew.unshift());
103 console.log(arrayNew);
```

On the right side of the interface, the 'Copilot' extension is active. It features a logo of a small robot head, the text 'Welcome to Copilot', and a button labeled 'Let's get started'. Below this, there is a text input field with placeholder text 'Add context (#), extensions (@), commands' and a 'Build Workspace' button. A note at the bottom says 'Review AI output carefully before use.'

## Semana 10 – Reporte

```
const array = ["Jorge", "Diego", "Pedro"];
```

Aquí se muestra la forma estándar de declarar un arreglo en JavaScript, el cual contiene tres elementos tipo cadena. Este ejemplo sirve como referencia para comparar con la implementación personalizada.

```
constructor(){
```

```
    this.length = 0;
```

```
    this.data = {};
```

```
}
```

length: almacena el número de elementos en el arreglo.

data: objeto que guarda los valores utilizando índices como claves.

push() para agregar elementos

get() para consultar

pop() para eliminar el último

delete() para eliminar por índice

unshift() para insertar al inicio

Las pruebas se muestran mediante console.log().

Welcome index.html

```
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Rick and Morty - Demo</title>
</head>
<body>
    <header class="site-header">
        <div class="container">
            <div class="brand">
                <div class="logo-placeholder">RM</div>
                <div>
                    <div class="site-title">Rick & Morty Demo</div>
                    <div class="muted" style="font-size:0.85rem">Demo: c
                </div>
            </div>
            <div class="nav-actions">
                <a class="btn" href="https://rickandmortyapi.com/" target="blank">Rick & Morty</a>
                <button class="btn btn-primary" id="refreshBtn">Actualizar</button>
            </div>
        </div>
    </header>

    <main>
        <section class="hero">
            <div class="container hero-grid">
                <div class="hero-card">
                    <h3 style="margin:0 0 8px">Explora personajes</h3>
                    <p class="muted">Esta demo carga los personajes de Rick & Morty</p>
                    <div style="margin-top:12px"><small class="muted">Actualizado: 10/06/2025</small></div>
                </div>
            </div>
        </section>
    </main>
</body>
```

Welcome to Copilot

Let's get started

Add context (F), extensions (B), commands (C)

Build Workspace Show Config

Review AI output carefully before use.

Restart Visual Studio Code to apply the latest update.

Update Now Later Release Notes

In 54, Col 8 Spaces: 4 UFT-8 CRLF ⌂ HTML ⌂ Go Live ⌂ Prettier

Busquedas

The screenshot shows the Microsoft Copilot extension integrated into the Visual Studio Code interface. The main area displays a portion of an HTML file named 'index.html' with line numbers from 1 to 53. The code includes sections for a header, a main content area with two cards, a grid container, and a footer. The sidebar on the right is titled 'Welcome to Copilot' and features a 'Let's get started' button, an 'Add context' button, and a 'Build Workspace' button. At the bottom, there are status indicators for 'In 22, Col 19', 'Spaces 4', 'UTF-8', 'CR/LF', 'HTML', 'Go To Line', and 'Python'. The taskbar at the bottom includes icons for search, file operations, and system notifications.

## Semana 11 – Reporte

Funciones principales:

meta charset="UTF-8": Permite el uso de caracteres especiales.

meta viewport: Hace que el sitio sea adaptable a dispositivos móviles.

link rel="stylesheet": Importa el archivo style.css, encargado del diseño visual.

title: Define el título que aparece en la pestaña del navegador.

Elementos:

Título "PRODUCT LIST": Indica la sección donde se mostrarán los productos.

Icono del carrito: Representado mediante un SVG, tomado de Flowbite Icons.

Contador (<span>0</span>): Muestra la cantidad de productos agregados al carrito.

The screenshot shows the Visual Studio Code interface with the 'index.html' file open. The code is for a simple e-commerce landing page. A Copilot AI sidebar is visible on the right, featuring a logo, the text 'Welcome to Copilot', and a 'Let's get started' button. Below these are buttons for 'Add context (E)', 'extensions (E)', 'commands (E)', 'Build Workspace', and 'Show Config'. A note at the bottom says 'Review AI output carefully before use.' The status bar at the bottom indicates the file is at line 43, column 8, with 4 spaces, in UTF-8 encoding, and shows icons for Go Live, Prettier, and other extensions.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Importar el estilos.css -->
    <link rel="stylesheet" href="style.css">
    <title>Proyecto E-commerce</title>
  </head>
  <body>
    <div class="container">
      <header>
        <div class="title">PRODUCT LIST</div>
        <div class="icon-cart">
          <!-- Referencia del icono https://flaticon.com/icons/... -->
          <svg class="w-6 h-6 text-gray-900 dark:text-white" aria-hidden="true">
            <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
              <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                  <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                    <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                      <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                        <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                          <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                            <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                              <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                  <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                    <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                      <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                        <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                          <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                            <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                              <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                  <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                    <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                      <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                        <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                          <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                            <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                              <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                                <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                                  <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                                    <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                                      <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                                        <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                                          <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                                            <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                                              <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                                                <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                                                  <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                                                    <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                                                      <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                                                                                        <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
              </path>
            </path>
          </svg>
          <span>0</span>
        </div>
      </header>
      <div class="listProduct">
        <div class="item">
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
              </div>
            </div>
          </div>
        </div>
      </div>
      <div class="cartTab">
        <h2>Shopping Cart</h2>
        <div class="listCart">
          <div class="item">
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
              </div>
            </div>
          </div>
        </div>
      </div>
      <div class="btn">
        <button class="close">CLOSE</button>
        <button class="checkout">CHECK OUT</button>
      </div>
    </div>
    <script src="app.js"></script>
  </body>
</html>
```

This screenshot shows the same VS Code environment after the user has interacted with the Copilot AI. The code now includes additional logic for a shopping cart, such as a 'close' button and a 'check out' button. The Copilot AI sidebar remains visible, providing the same interface and status information as in the first screenshot.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Importar el estilos.css -->
    <link rel="stylesheet" href="style.css">
    <title>Proyecto E-commerce</title>
  </head>
  <body>
    <div class="container">
      <header>
        <div class="title">PRODUCT LIST</div>
        <div class="icon-cart">
          <!-- Referencia del icono https://flaticon.com/icons/... -->
          <svg class="w-6 h-6 text-gray-900 dark:text-white" aria-hidden="true">
            <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
              <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                  <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                    <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                      <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
                        <path stroke="currentColor" stroke-linecap="round" stroke-width="2">
              </path>
            </path>
          </svg>
          <span>0</span>
        </div>
      </header>
      <div class="listProduct">
        <div class="item">
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
              </div>
            </div>
          </div>
        </div>
      </div>
      <div class="cartTab">
        <h2>Shopping Cart</h2>
        <div class="listCart">
          <div class="item">
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
              </div>
            </div>
          </div>
        </div>
      </div>
      <div class="btn">
        <button class="close">CLOSE</button>
        <button class="checkout">CHECK OUT</button>
      </div>
    </div>
    <script src="app.js"></script>
  </body>
</html>
```

## Semana 12 – Reporte

El presente código CSS tiene como finalidad definir el diseño visual, la estructura y el comportamiento responsivo de una aplicación web. El estilo está orientado a mostrar elementos tipo tarjetas (cards), botones y listas, haciendo uso de variables CSS, flexbox, grid layout y media queries para lograr una interfaz moderna, ordenada y adaptable a distintos tamaños de pantalla.

Se realiza un reset de estilos, eliminando márgenes y rellenos por defecto, y estableciendo:

box-sizing: border-box para un mejor control del tamaño de los elementos

Fuente Rubik como tipografía principal

### ● Estilos generales de elementos

ul: elimina viñetas de listas

button: elimina bordes y fondos por defecto

body: establece altura mínima completa y color de fondo gris

Información interna del Pokémon

Se definen estilos para:

Imagen (.pokemon-imagen)

Identificador (.pokemon-id)

Nombre (.pokemon-nombre)

Tipos (.pokemon-tipos)

Estadísticas (.pokemon-stats)

Todo se organiza usando Flexbox para centrar y distribuir el contenido correctamente.

```
index.html
1 import url('https://fonts.googleapis.com/css2?family=Rubik:wght@300;400');
2
3 :root {
4   --clr-black: #1c1c1c;
5   --clr-gray: #e0e0e0;
6   --clr-white: #f7f7f7;
7
8   --type-normal: #A8A8D8;
9   --type-fire: #F08030;
10  --type-water: #6890E8;
11  --type-grass: #78C858;
12  --type-electric: #F8D030;
13  --type-ice: #98D8D8;
14  --type-fighting: #C0392B;
15  --type-poison: #A8D8A8;
16  --type-ground: #E8C068;
17  --type-flying: #A8A8F0;
18  --type-psychic: #F85888;
19  --type-bug: #A8B820;
20  --type-rock: #B8A8A8;
21  --type-ghost: #705898;
22  --type-dark: #705848;
23  --type-dragon: #7038F0;
24  --type-steel: #B8B8D0;
25  --type-fairy: #F0B8E8;
26 }
27
28 * {
29   margin: 0;
30   padding: 0;
31   box-sizing: border-box;
32   color: var(--clr-black);
33 }
```

The screenshot shows a code editor window with two tabs: "index.html" and "code.js". The "index.html" tab is active, displaying the provided CSS code. To the right of the editor, there is a "Copilot" interface with a "Welcome to Copilot" message, a "Let's get started" button, and a "Review AI output carefully before use." note. The status bar at the bottom shows file statistics like "Line 295, Col 2", "Spaces: 4", "UTF-8", "CRLF", and icons for "HTML", "Go Live", "Prettier", and "Copilot". The taskbar at the bottom includes icons for search, file, messaging, browser, and other system functions.

```
index.html
1 .header {
2     display: flex;
3     align-items: center;
4     flex-wrap: wrap;
5     gap: .5rem;
6 }
7
8 .btn-header{
9     background-color: var(--clr-gray);
10    padding: .5rem;
11    border-radius: 1000px;
12    cursor: pointer;
13    text-transform: uppercase;
14    font-weight: 600;
15    box-shadow: 0 0 2rem rgba(0, 0, 0, 1);
16    transition: .2s;
17 }
18
19 .btn-header:hover{
20     transform: scale(1.1);
21     box-shadow: 0 0 2rem rgba(0, 0, 0, 1);
22 }
23
24 main{
25     padding: 2rem;
26     max-width: 1000px;
27     margin: 0 auto;
28 }
29
30
31 /*Este código permite validar cuando se visualice en una pantalla de cel
32 .pokemon-todos{
```

Welcome to Copilot

Let's get started

Add context (F), extensions (B), commands

Build Workspace Show Config

Review AI output carefully before use.

Ln 295, Col 2 Spaces: 4 UTF-8 CR/LF Go Live Prettier

ESP LAA 07:14 p.m. 15/12/2025

## Semana 13 – Reporte

### EduTrack - Sistema de Asistencia y Participación

#### Descripción general

EduTrack es un sistema completo de gestión de asistencia y participación estudiantil diseñado para instituciones educativas. Ofrece una interfaz moderna y eficiente que permite a los docentes registrar, monitorear y analizar el rendimiento académico de sus estudiantes.

#### Características principales

##### Gestión de Asistencia

Registro rápido de asistencia por clase y fecha

Marcación masiva (todos presentes/ausentes)

Registro de llegadas tardías

Historia completa de asistencia por estudiante.

##### Seguimiento de Participación

Evaluación de participación con sistema de puntuación (1-10)

Categorización por tipo de participación (respuesta, pregunta, participación activa, presentación)

Notas descriptivas para cada registro

Análisis de patrones de participación

##### Reportes y Análisis

Dashboard con estadísticas en tiempo real

Informes detallados de asistencia y participación

Análisis individual de estudiantes

Vista general por clase

Exportación de datos en formato JSON

##### Interfaz moderna

Diseño responsivo y accesible

Animaciones suaves y microinteracciones.

Panel de control intuitivo

Notificaciones en tiempo real

Arquitectura del Sistema

Tecnologías utilizadas

Interfaz : HTML5, CSS3, JavaScript ES6+

Marco CSS : Tailwind CSS

Animaciones : Anime.js

Almacenamiento : LocalStorage (base de datos local)

Tipografías : Inter (texto principal), JetBrains Mono (datos)

The screenshot shows a code editor with an AI completion interface overlaid. The code editor displays an HTML file named 'index.html' containing Tailwind CSS and Anime.js imports. The AI interface, titled 'Welcome to Copilot', suggests code for a navigation bar. It includes a 'Let's get started' button, a 'Build Workspace' button, and a note to 'Review AI output carefully before use'. The interface also has sections for 'Add context (S, extensions (B), commands)' and 'Show Config'.

```
<!DOCTYPE html><html lang="es"><head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Edatrack - Sistema de Asistencia y Participación</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/animejs/3.2.1/anime.min.js"></script>
  <link rel="preconnect" href="https://fonts.googleapis.com"/>
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Inter:weight:400,500,700&display=swap" rel="stylesheet">
<style>
  * { font-family: 'Inter', sans-serif; }
  .mono { font-family: 'JetBrains Mono', monospace; }
  .gradient-bg { background: linear-gradient(135deg, #efafac 45%, #e0e0e0 55%); }
  .card-hover { transition: all 0.2s ease-out; }
  .card-hover:hover { transform: translate(-2px); box-shadow: 0 1px 1px 0 #e0e0e0; }
  .pulse-animation { animation: pulse 2s infinite; }
  @keyframes pulse { 0%, 100% { opacity: 1; } 50% { opacity: 0.5; } }
  .slide-in { animation: slideIn 0.3s ease-out; }
  @keyframes slideIn { from { transform: translateY(-20px); opacity: 0; } to { opacity: 1; } }
  .loading-skeleton { background: linear-gradient(90deg, #e0e0e0 45%, #d0d0d0 55%); }
  @keyframes loading { 0% { background-position: 200% 0; } 100% { background-position: 0 0; } }
</style>
<script src="https://statics.monshot.cn/sdk/precise-widgets.min.js" defer></script>
<body class="gradient-bg min-h-screen">
  <!-- Navigation -->
  <nav class="bg-white shadow-sm border-b border-gray-200 sticky top-0 w-full px-4 sm:px-6 lg:px-8">
    <div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
      <div class="flex justify-between items-center h-16">
        <div class="flex items-center space-x-4">
          <div class="flex-shrink-0">
            <img alt="Edatrack logo" class="h-8 w-8" />
            <div class="text-2xl font-bold text-blue-600">Ed</div>
          </div>
        </div>
      </div>
    </div>
  </nav>
</body>
```

The screenshot shows a Microsoft Copilot AI interface integrated into a code editor. The left side displays an HTML file named 'index.html' with code for a student attendance application. The right side features a sidebar with the Copilot logo, a 'Welcome to Copilot' message, and a 'Let's get started' button. It also includes a context input field, 'Build Workspace' and 'Show Config' buttons, and a note to 'Review AI output carefully before use.' The bottom status bar shows file details like 'Ln 328, Col 15' and 'Spaces: 4'.

```
index.html > html
1  <!DOCTYPE html><html lang="es"><head>
2  </head>
3  <body class="gradient-bg min-h-screen">
4      <div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8 py-8">
5          <div class="grid grid-cols-1 lg:grid-cols-3 gap-6 mb-8">
6              <div class="bg-white rounded-xl p-6 shadow-sm border border-gray-200">
7                  <!-- Student list will be populated by JavaScript -->
8              </div>
9          </div>
10         <!-- Main Actions -->
11         <div class="grid grid-cols-1 lg:grid-cols-3 gap-8">
12             <!-- Attendance Section -->
13             <div class="bg-white rounded-xl shadow-sm border border-gray-200">
14                 <div class="p-6 border-b border-gray-200">
15                     <h3 class="text-lg font-semibold text-gray-900 mb-2">
16                         <p class="text-sm text-gray-600">Marca la asistencia
17                     </div>
18                     <div class="p-6">
19                         <div class="space-y-4">
20                             <div>
21                                 <label class="block text-sm font-medium text-gray-900">
22                                     <select id="classSelect" class="w-full border border-gray-200 p-2">
23                                         <option value="">Seleccionar clase</option>
24                                         <option value="math-10a">Matemáticas 10A</option>
25                                         <option value="science-10b">Ciencias 10B</option>
26                                         <option value="history-9a">Historia 9A</option>
27                                     </select>
28                             </div>
29                         </div>
30                     </div>
31                 </div>
32             </div>
33         </div>
34     </div>
35 
```

## Semana 14 – Reporte

### Registro de los datos

Tener un arreglo, dentro de ese arreglo una estructura los datos que solicita son una id, título, descripción, completada.

El presente código HTML corresponde a una aplicación web tipo ToDo, cuyo objetivo principal es permitir al usuario registrar, consultar, editar y eliminar tareas, utilizando almacenamiento local del navegador. El archivo define la estructura base de la interfaz, mientras que el diseño y la lógica se implementan mediante archivos externos de CSS y JavaScript.

### Función de los elementos:

**meta charset="UTF-8":** Permite el uso de caracteres especiales.

**meta viewport:** Garantiza un diseño responsivo.

**title:** Define el nombre de la página en el navegador.

**link stylesheet:** Conecta el archivo CSS para el diseño visual.

**meta description:** Describe brevemente la funcionalidad de la aplicación.

El archivo app.js se encarga de:

- Gestionar eventos del formulario
- Almacenar tareas en localStorage
- Mostrar, editar y eliminar tareas
- Controlar el modal de edición

The screenshot shows the NetBeans IDE interface with the Copilot feature integrated. The main window displays an HTML file named 'index.html' with code related to a modal dialog for editing tasks. The Copilot panel on the right is titled 'Welcome to Copilot' and includes a 'Let's get started' button, an 'Add context (4), extensions (0), commands' button, a 'Build Workspace' button, and a 'Show Config' button. A note at the bottom of the panel says 'Review AI output carefully before use.' The status bar at the bottom indicates the current file is 'index.html', and the system status shows '07:40 p.m. 13/12/2025'.

```
<html lang="es">
<body>
    <main class="container">
        <section class="panel">
            <div class="controls">
                ...
            </div>

            <ul id="task-list" class="task-list" aria-live="polite"></ul>
        </section>

        <!-- Edit modal -->
        <div id="edit-modal" class="modal hidden" role="dialog" aria-modal="true">
            <div class="modal-content">
                <h3 id="edit-title">Editar tarea</h3>
                <form id="edit-form" class="form">
                    <input type="hidden" id="edit-id">
                    <label for="edit-title-input">Título</label>
                    <input type="text" id="edit-title-input" required>

                    <label for="edit-description-input">Descripción</label>
                    <textarea id="edit-description-input"></textarea>

                    <label class="checkbox"><input type="checkbox" id="edit-priority"></label>

                    <div class="form-actions">
                        <button type="submit" class="btn">Guardar</button>
                        <button type="button" id="edit-cancel" class="btn">Cancelar</button>
                    </div>
                </form>
            </div>
        </div>
    </main>
</body>
```

This screenshot is identical to the one above, showing the NetBeans IDE interface with the Copilot feature. It displays the same HTML code for 'index.html' and the same Copilot panel on the right. The status bar at the bottom indicates the current file is 'index.html', and the system status shows '07:40 p.m. 13/12/2025'.

```
<html lang="es">
<body>
    <main class="container">
        <section class="panel">
            <div class="controls">
                ...
            </div>

            <ul id="task-list" class="task-list" aria-live="polite"></ul>
        </section>

        <!-- Edit modal -->
        <div id="edit-modal" class="modal hidden" role="dialog" aria-modal="true">
            <div class="modal-content">
                <h3 id="edit-title">Editar tarea</h3>
                <form id="edit-form" class="form">
                    <input type="hidden" id="edit-id">
                    <label for="edit-title-input">Título</label>
                    <input type="text" id="edit-title-input" required>

                    <label for="edit-description-input">Descripción</label>
                    <textarea id="edit-description-input"></textarea>

                    <label class="checkbox"><input type="checkbox" id="edit-priority"></label>

                    <div class="form-actions">
                        <button type="submit" class="btn">Guardar</button>
                        <button type="button" id="edit-cancel" class="btn">Cancelar</button>
                    </div>
                </form>
            </div>
        </div>
    </main>
</body>
```

## Semana 15 – Reporte

El presente código tiene como finalidad implementar una estructura de datos tipo Tabla Hash (Hash Table) utilizando clases en JavaScript. Esta estructura permite almacenar y recuperar datos mediante pares clave–valor, ofreciendo un acceso eficiente a la información a través de una función hash.

El programa define una clase llamada HashTable, la cual:

Utiliza un arreglo como estructura base.

Emplea una función hash para convertir claves en índices.

Implementa métodos para insertar, obtener, eliminar y listar claves.

```
hashMethod(key) {  
    let hash = 0;  
    for (let i = 0; i < key.length; i++) {  
        hash = (hash + key.charCodeAt(i) * i) % this.data.length;  
    }  
    return hash;  
}
```

Este método:

Convierte una clave tipo string en un índice numérico.

Utiliza el valor ASCII de cada carácter.

Aplica el operador módulo para asegurar que el índice esté dentro del tamaño del arreglo.

```

set(key, value) {
    const address = this.hashMethod(key);
    if (!this.data[address]) {
        this.data[address] = [];
    }
    this.data[address].push([key, value]);
    return this.data;
}

```

Permite insertar un par clave–valor en la tabla hash.

Si ya existe información en esa posición, se utiliza un arreglo interno (bucket) para manejar colisiones mediante separate chaining.



The screenshot shows a Windows desktop environment with a dark-themed taskbar at the bottom. On the taskbar, there is a search bar labeled "Búsqueda" (Search), followed by several pinned icons: File Explorer, Control Panel, Mail, Internet Explorer, Task View, and others. The main window is a code editor showing a JavaScript file named "code.js". The code defines a class "HashTable" with methods for setting and getting key-value pairs using separate chaining.

```

class HashTable {
    constructor(size) {
        this.data = new Array(size);
    }

    hashMethod(key) {
        let hash = 0;
        for (let i = 0; i < key.length; i++) {
            hash = (hash + key.charCodeAt(i) * i) % this.data.length;
        }
        return hash;
    }

    set(key, value) {
        const address = this.hashMethod(key);
        if (!this.data[address]) {
            this.data[address] = [];
        }
        this.data[address].push([key, value])
        return this.data;
    }

    get(key) {
        const address = this.hashMethod(key);
        const currentBucket = this.data[address];
        if (currentBucket) {
            for (let i = 0; i < currentBucket.length; i++) {
                if (currentBucket[i][0] === key) {
                    return currentBucket[i][1];
                }
            }
        }
    }
}

```

Ln 65, Col 39 Spaces: 4 CRLF { }

## Semana 16 – Reporte

El presente código HTML corresponde a la estructura base de una aplicación web que tiene como finalidad consumir y mostrar información de la API de Pokémon, permitiendo filtrar los resultados por tipo. El archivo define la organización visual de la página y se apoya en un archivo CSS externo para el diseño.

Función de los elementos:

meta charset="UTF-8": Permite la correcta visualización de caracteres especiales.

meta viewport: Asegura un diseño responsivo para dispositivos móviles.

title: Define el título de la página en el navegador.

link rel="stylesheet": Conecta el archivo index.css, encargado del diseño visual.

```
<main>
    <h1>Hola como estas</h1>
    <p>Hola como estas</p>
</main>
```

Esta sección representa el contenido principal de la página. Actualmente contiene texto de ejemplo, el cual puede ser reemplazado por:

Tarjetas de Pokémon

Información obtenida dinámicamente desde la API

Resultados del filtrado por tipo

Integración con CSS y JavaScript

Aunque el código presentado solo incluye HTML y CSS:

El archivo index.css se encarga del diseño visual.

Posteriormente puede integrarse un archivo JavaScript para:

Consumir la API de Pokémon

Mostrar los datos en pantalla

Filtrar resultados según el botón seleccionado

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<body>
<div class="container-type">
<button class="btn-type-Water">Water</button>
<button class="btn-type-Normal">Normal</button>
<button class="btn-type-Fire">Fire</button>
<button class="btn-type-Water">Water</button>
<button class="btn-type-Grass">Grass</button>
<button class="btn-type-Electric">Electric</button>
<button class="btn-type-Ice">Ice</button>
<button class="btn-type-Fighting">Fighting</button>
<button class="btn-type-Poison">Poison</button>
<button class="btn-type-Ground">Ground</button>
<button class="btn-type-Flying">Flying</button>
<button class="btn-type-Psychic">Psychic</button>
<button class="btn-type-Dark">Dark</button>
<button class="btn-type-Rock">Rock</button>
<button class="btn-type-Ghost">Ghost</button>
<button class="btn-type-Dragon">Dragon</button>
<button class="btn-type-Steel">Steel</button>
<button class="btn-type-Fairy">Fairy</button>
</div>
</body>
</html>
```

## INSIGNIA

[https://developers.google.com/profile/u/104498823974458677367?utm\\_source=web.dev](https://developers.google.com/profile/u/104498823974458677367?utm_source=web.dev)

The screenshot shows a mobile browser interface with a dark theme. At the top, the URL 'developers.google.com' is visible in the address bar, along with a house icon, a refresh button, a plus sign for new tabs, a notification badge showing '23', and a three-dot menu. Below the address bar, the page content is displayed. It features a header with the text 'translated by Google' and a note that the page was translated using the 'API de Cloud Translation'. A 'Switch to English' button is present. The main content area contains a large graphic of a smartphone displaying a dashboard with three circular icons. The phone is surrounded by various colored squares (green, red, blue, yellow) and smaller grid patterns. Below the graphic, the text 'SE OBTUVO EL 13 DIC 2025' is shown. The main title 'Primera insignia de ruta de aprendizaje y cuestionario' is prominently displayed in large black font. Below the title, a subtitle reads 'Completaste la primera ruta de aprendizaje y el cuestionario'. There are sharing options for 'Compartir' with links to X, Facebook, and LinkedIn. A success message at the bottom says '¡Ya tienes esta insignia!' with a checkmark icon.