



UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE
UNIDAD DE EDUCACIÓN A DISTANCIA



Asignatura:

Programación Integrativa Componentes Web

Ing.

Vilmer David Criollo Chanchicocha

Tema:

Estructuras de Control en JavaScript

Nombre:

Carlos Ramiro Yáñez Yazán

NRC:

23407

Quito...20...mayo de 2025

Estructuras de Control en JavaScript

Objetivo General

- Desarrollar tres programas JavaScript independientes que interactúen con el usuario a través de la consola para clasificar edades, generar tablas de multiplicar y mostrar tablas de multiplicar pares o impares, aplicando estructuras de control condicionales

Objetivos Específicos

- Implementar un programa que solicite la edad del usuario y determine su categoría (Niño, Adolescente, Adulto, Adulto mayor) utilizando if anidados y validando entradas no válidas
- Crear un programa que pida al usuario un número y muestre su tabla de multiplicar del 1 al 12 utilizando un bucle for para realizar las operaciones y mostrar los resultados.
- Desarrollar un programa que pregunte al usuario si desea ver las tablas de multiplicar de números pares o impares entre 1 y 10, utilizando una estructura if para seleccionar el tipo de tablas y un bucle for para imprimir las tablas correspondientes.

Desarrollo

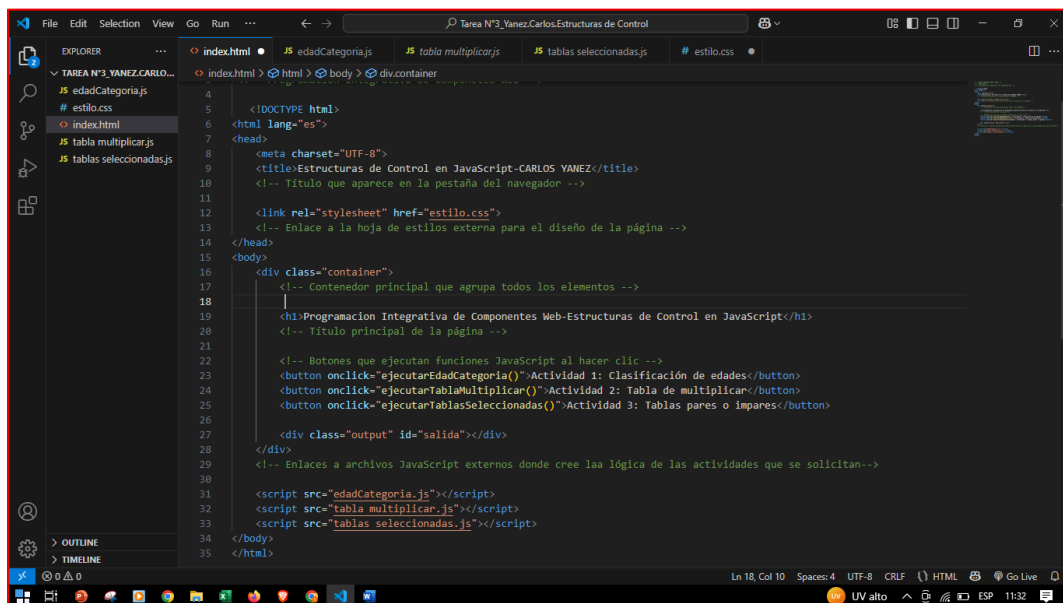
1. Enlace a GitHub

https://github.com/carlitos3454/TareaN-3_Yanez.Carlos_Estrcuturas-de-Control-en-JavaScript.git

2. Código Fuente

Para este punto se buscó desarrollar 5 archivos diferentes relacionados entre si para dar funcionalidad al sistema que se solicita en la actividad para las estructuras js.

Index.html



```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8">
5 <title>Estructuras de Control en JavaScript-CARLOS YANEZ</title>
6 <!-- Título que aparece en la pestaña del navegador -->
7
8 <link rel="stylesheet" href="estilo.css">
9 <!-- Enlace a la hoja de estilos externa para el diseño de la página -->
10 </head>
11 <body>
12 <div class="container">
13 <!-- Contenedor principal que agrupa todos los elementos -->
14
15 <h1>Programación Integrativa de Componentes Web-Estructuras de Control en JavaScript</h1>
16 <!-- Título principal de la página -->
17
18 <!-- Botones que ejecutan funciones JavaScript al hacer clic -->
19 <button onclick="ejecutarEdadCategoria()">Actividad 1: Clasificación de edades</button>
20 <button onclick="ejecutarTablaMultiplicar()">Actividad 2: Tabla de multiplicar</button>
21 <button onclick="ejecutarTablasSeleccionadas()">Actividad 3: Tablas pares o impares</button>
22
23 <div class="output" id="salida"></div>
24 </div>
25 <!-- Enlaces a archivos JavaScript externos donde cree la lógica de las actividades que se solicitan -->
26 <script src="edadCategoria.js"></script>
27 <script src="tablaMultiplicar.js"></script>
28 <script src="tablasSeleccionadas.js"></script>
29 </body>
30 </html>
```

Una vez establecido el archivo principal se definen los archivos de js para las estructuras de control “Archivos JavaScript”.

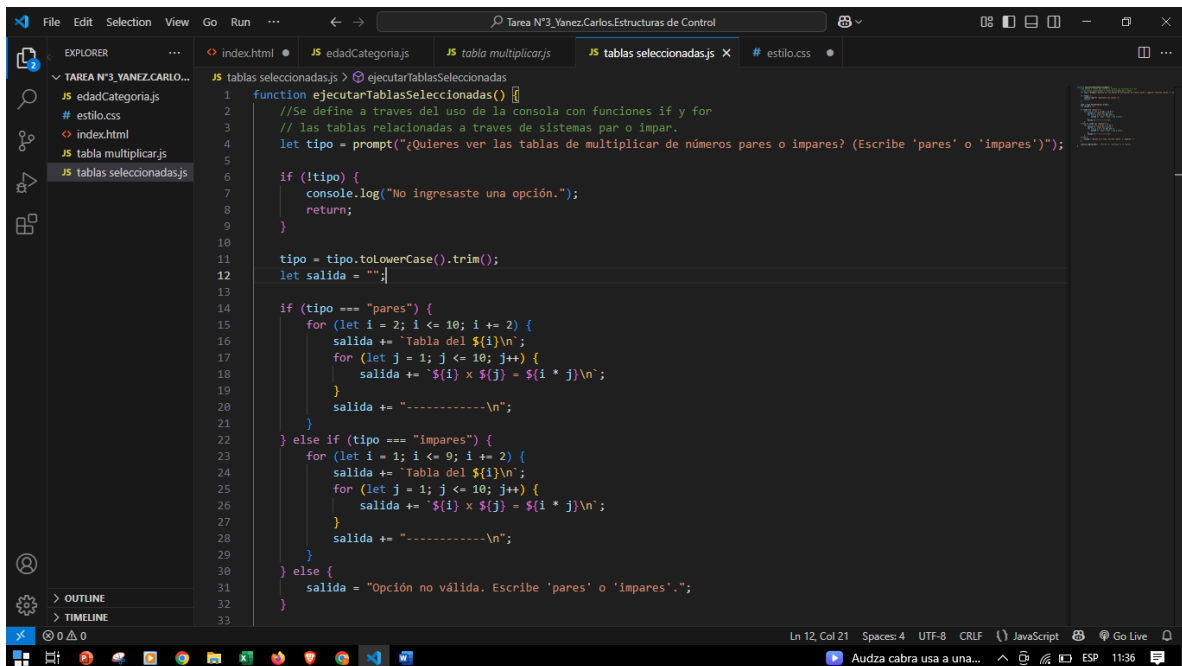
(Edad Categoria.js)

```
1 function ejecutarEdadCategoria() {
2   let edad = parseInt(prompt("Ingresa tu edad:"));
3   let mensaje = "";
4   //function js para definir una edad en torno a un valor establecido para cada criterio como niño, adolescente y adulto mayor
5
6   if (isNaN(edad)) {
7     mensaje = "Por favor ingresa un número válido.";
8   } else if (edad < 0) {
9     mensaje = "Edad no válida";
10  } else if (edad <= 12) {
11    mensaje = "Niño";
12  } else if (edad <= 17) {
13    mensaje = "Adolescente";
14  } else if (edad <= 59) {
15    mensaje = "Adulto";
16  } else {
17    mensaje = "Adulto mayor";
18  }
19  console.log(mensaje); // Aquí se muestra el mensaje en la consola
20 }
21
```

(Tabla de Multiplicar.js)

```
1 function ejecutarTablaMultiplicar() {
2   // Defini el uso de if y ciclos for para establecer la estutruectura de las tablas de multiplicar.
3
4   let numeroTexto = prompt("Escribe un número para ver su tabla de multiplicar:");
5
6   if (numeroTexto === null || numeroTexto.trim() === "") {
7     console.log("No ingresaste un número.");
8     return;
9   }
10
11   let numero = parseInt(numeroTexto);
12
13   if (isNaN(numero)) {
14     console.log("Entrada no válida. Ingresar un número.");
15     return;
16   }
17
18   let salida = `Tabla del ${numero}\n`;
19   for (let i = 1; i <= 12; i++) {
20     salida += `${numero} x ${i} = ${numero * i}\n`;
21   }
22   console.log(salida); // Mostrar toda la tabla en la consola
23 }
24
```

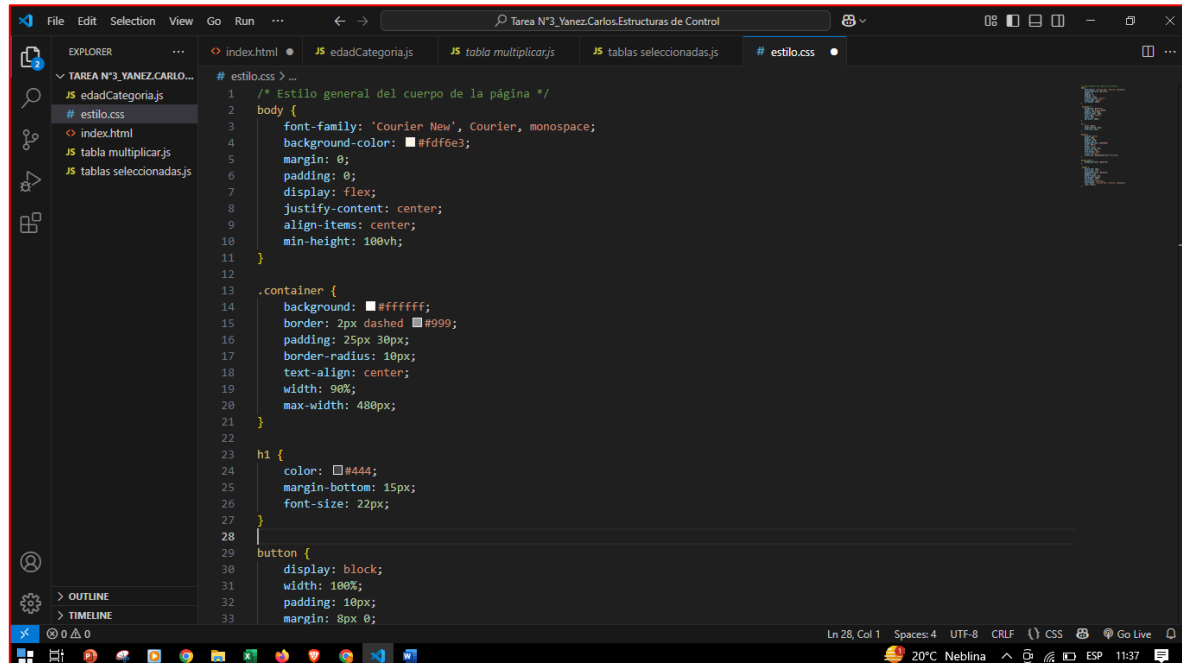
(Tablas seleccionadas.js) - pares o impares



```
1 function ejecutarTablasSeleccionadas() {
2     //Se define a traves del uso de la consola con funciones if y for
3     // las tablas relacionadas a traves de sistemas par o impar.
4     let tipo = prompt("¿Quieres ver las tablas de multiplicar de números pares o impares? (Escribe 'pares' o 'impares')");
5
6     if (!tipo) {
7         console.log("No ingresaste una opción.");
8         return;
9     }
10
11     tipo = tipo.toLowerCase().trim();
12     let salida = "";
13
14     if (tipo === "pares") {
15         for (let i = 2; i <= 10; i += 2) {
16             salida += `Tabla del ${i}\n`;
17             for (let j = 1; j <= 10; j++) {
18                 salida += `${i} x ${j} = ${i * j}\n`;
19             }
20             salida += "-----\n";
21         }
22     } else if (tipo === "impares") {
23         for (let i = 1; i <= 9; i += 2) {
24             salida += `Tabla del ${i}\n`;
25             for (let j = 1; j <= 10; j++) {
26                 salida += `${i} x ${j} = ${i * j}\n`;
27             }
28             salida += "-----\n";
29         }
30     } else {
31         salida = "Opción no válida. Escribe 'pares' o 'impares.'";
32     }
33 }
```

Por último, se añadió una hoja de estilo para definir una presentación mas persuasiva para el usuario.

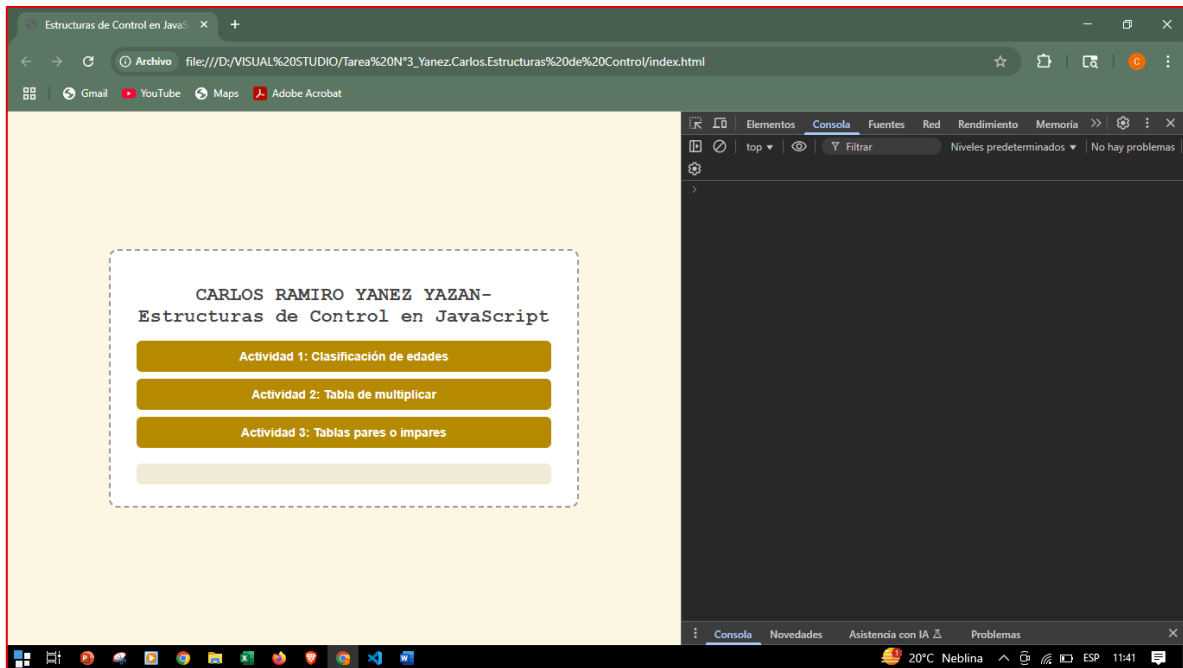
(Hoja de Estilo.css)



```
1 /* Estilo general del cuerpo de la página */
2 body {
3     font-family: 'Courier New', Courier, monospace;
4     background-color: #fdf6e3;
5     margin: 0;
6     padding: 0;
7     display: flex;
8     justify-content: center;
9     align-items: center;
10    min-height: 100vh;
11 }
12
13 .container {
14     background: #ffffff;
15     border: 2px dashed #999;
16     padding: 25px 30px;
17     border-radius: 10px;
18     text-align: center;
19     width: 90%;
20     max-width: 480px;
21 }
22
23 h1 {
24     color: #444;
25     margin-bottom: 15px;
26     font-size: 22px;
27 }
28
29 button {
30     display: block;
31     width: 100%;
32     padding: 10px;
33     margin: 8px 0;
```

3. Proyecto Ejecutando

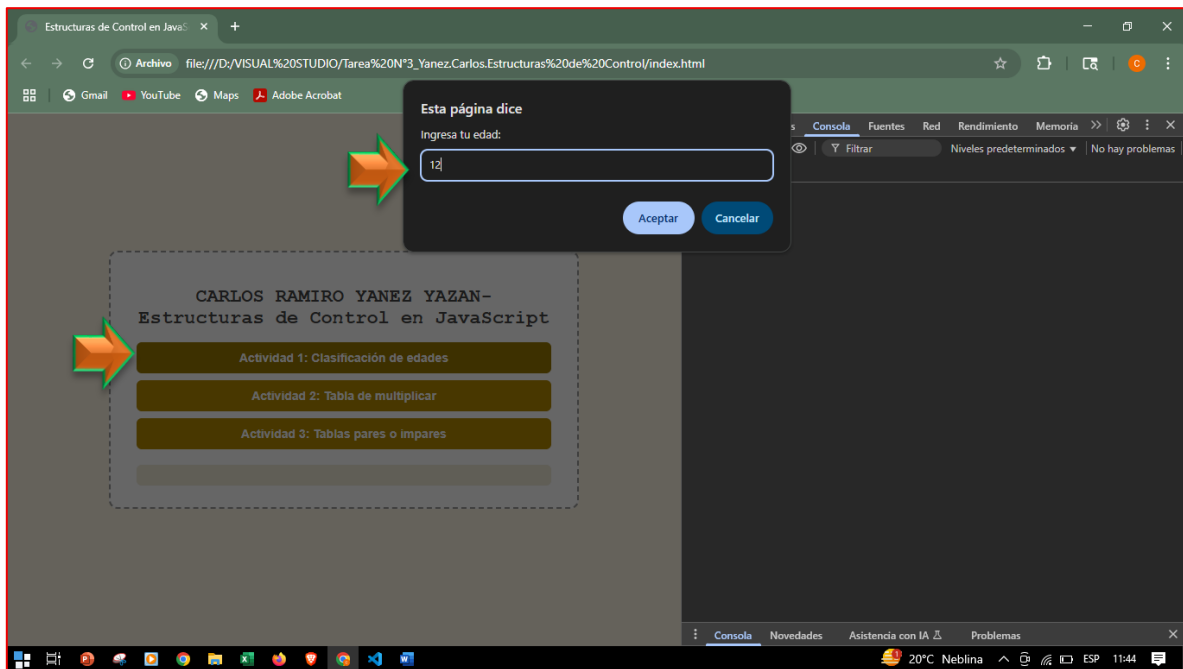
Para la ejecución del proyecto se define la página de la siguiente forma.



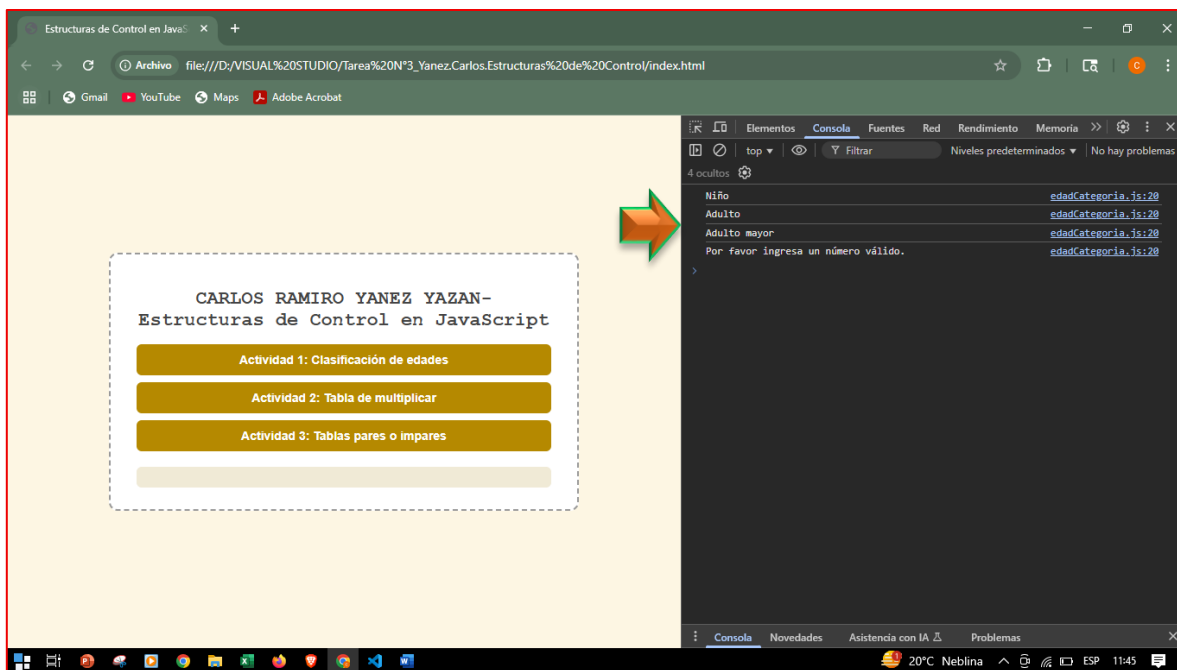
Para comprobar cada una de las funciones se procede a manejar a través de cada una de las actividades.

Actividad 1: Clasificación de Edades.

Se da clic en la primera opción y aparecerá el siguiente enunciado en la parte superior del navegador e ingresamos la edad.

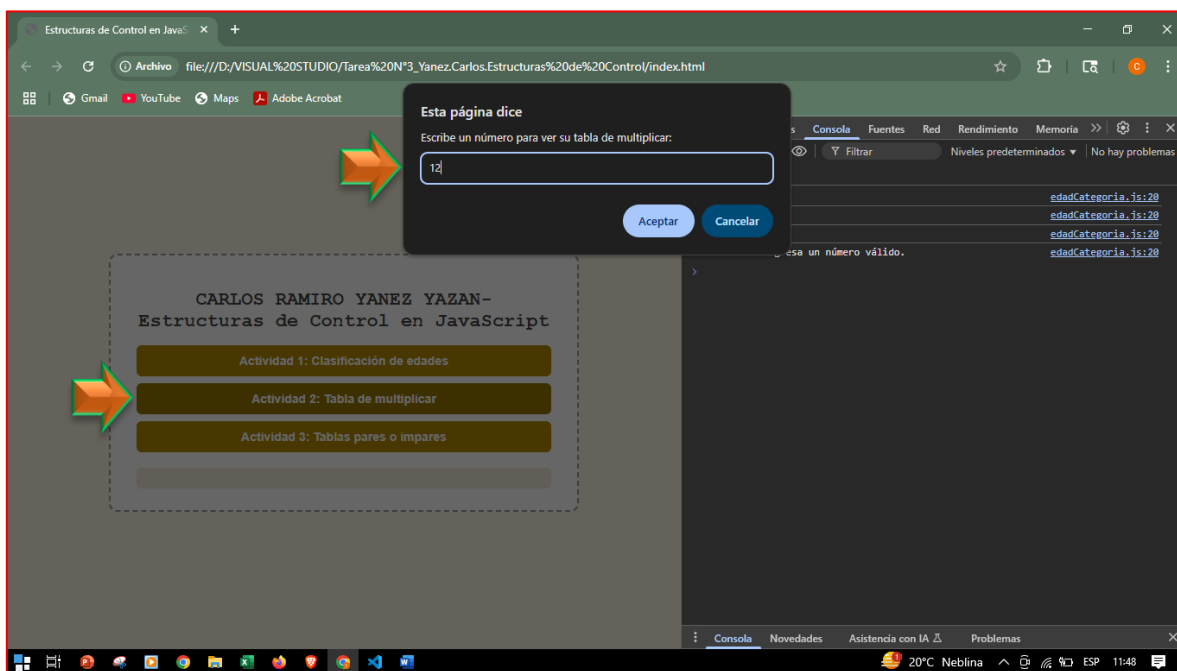


Una vez ingresado la edad el resultado debe reflejarse dentro de la consola, para esto se añadió 3 ejemplos mas de como debe visualizarse para **la edad de un niño, adulto, adulto mayor y de ser el caso que no se registre un valor.**

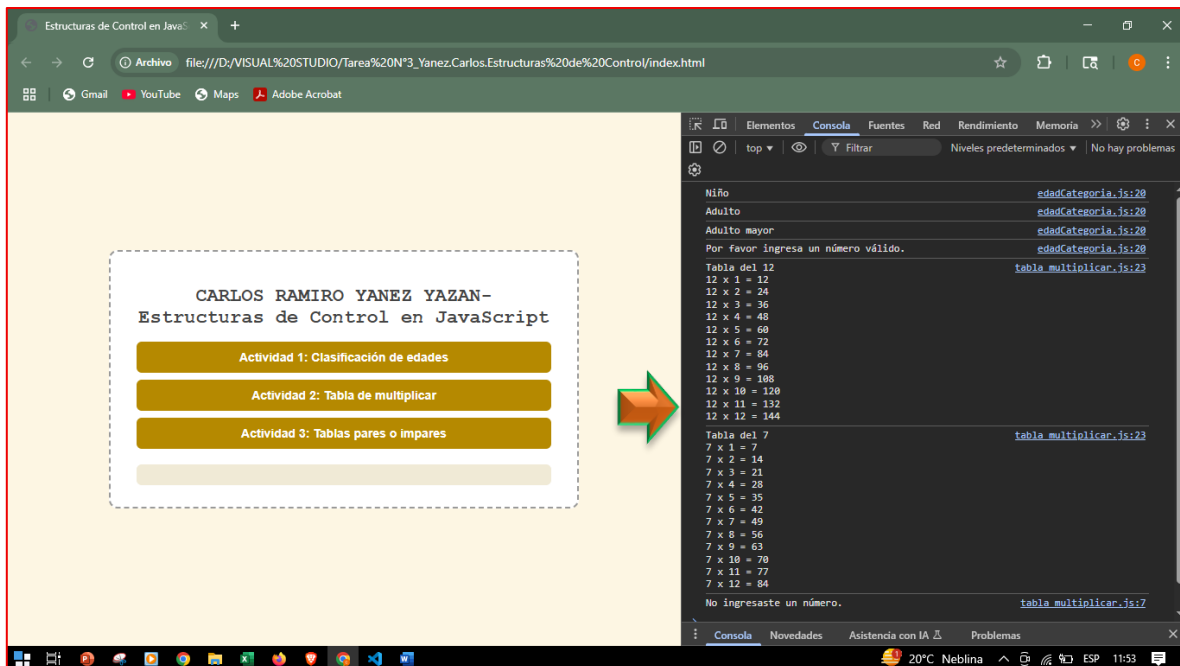


Actividad 2: Tablas de multiplicar.

Se da clic en la segunda opción y aparecerá el siguiente enunciado en la parte superior del navegador e ingresamos el valor de la tabla que queremos visualizar.

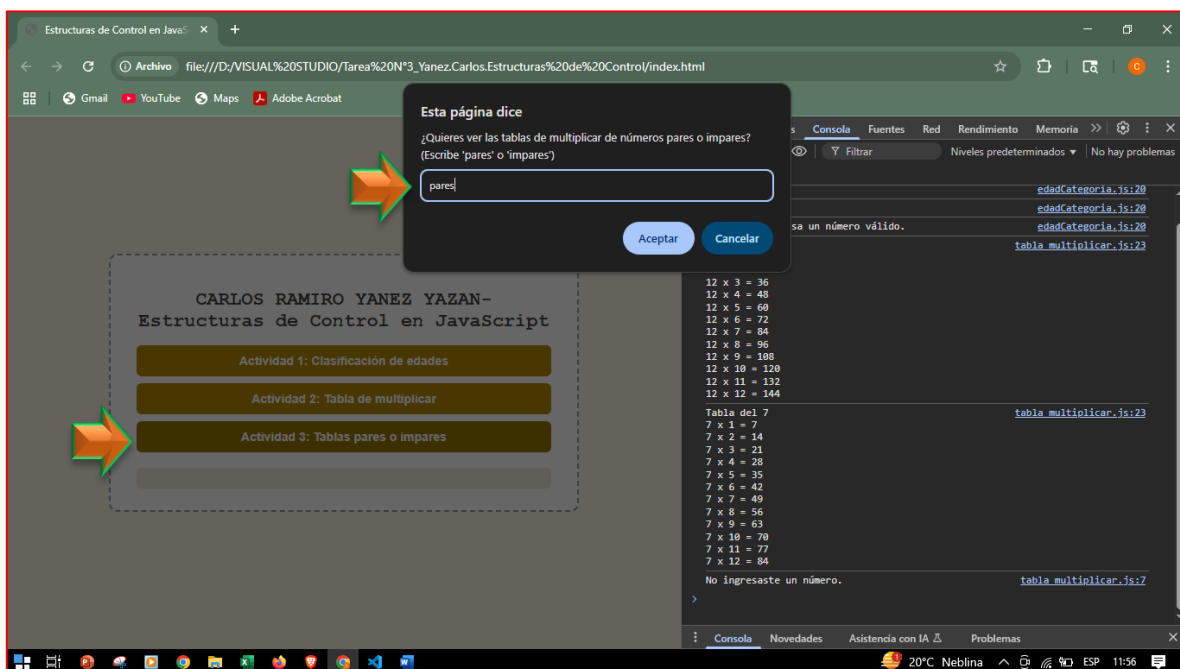


Una vez ingresado el valor el resultado debe reflejarse dentro de la consola, para esto se añadió 2 ejemplos más de como debe visualizarse para **la tabla del 12, 7 y de ser el caso que no se registre un valor.**



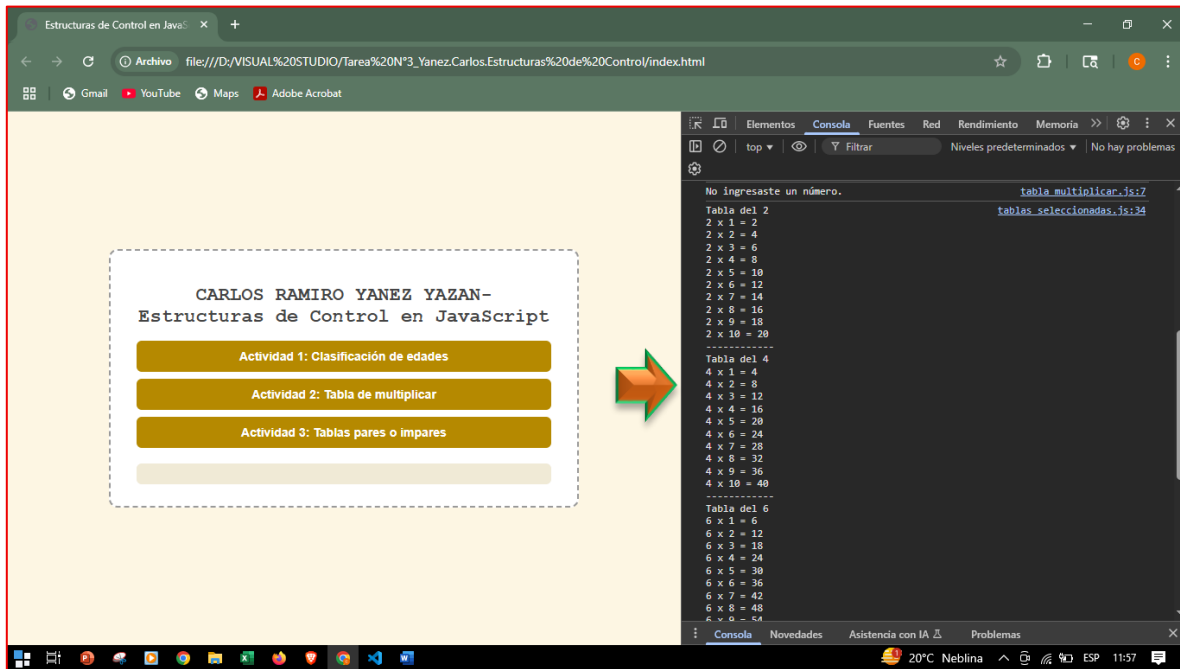
Actividad 3: Tablas pares o impares.

Se da clic en la tercera opción y aparecerá el siguiente enunciado en la parte superior del navegador e ingresamos la palabra de la tabla que queremos visualizar sea par o impar.

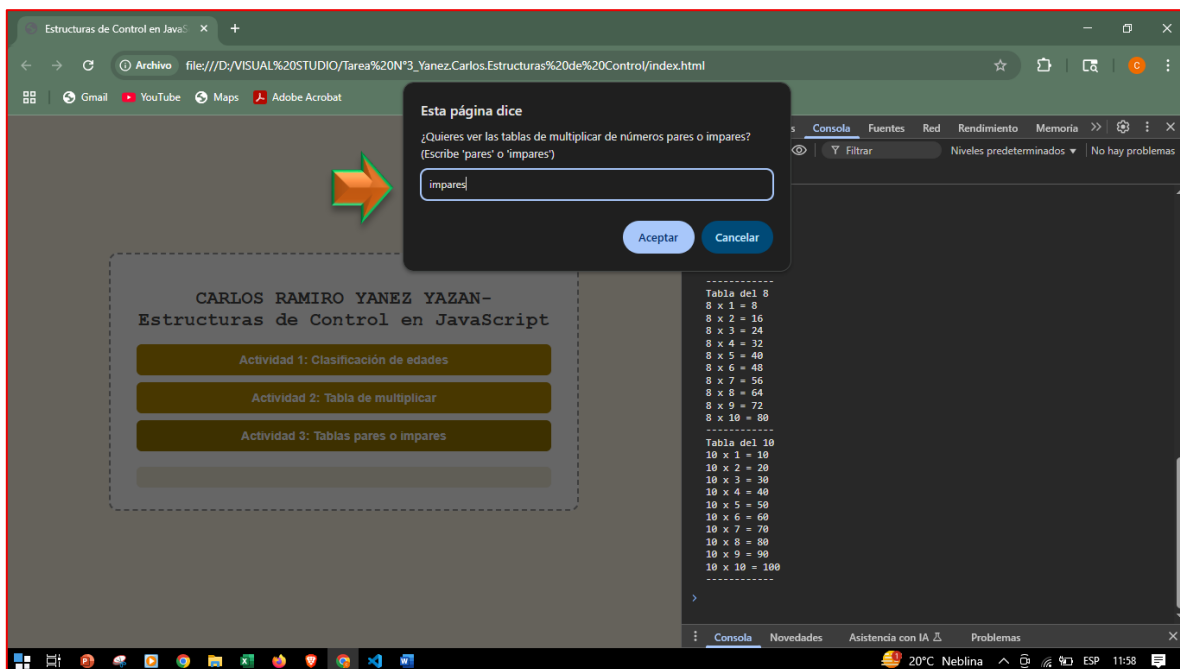


Una vez ingresado el valor el resultado debe reflejarse dentro de la consola, para esto se añadió 2 ejemplos más de cómo debe visualizarse sea par o impar además **de ser el caso que no se registre un valor.**

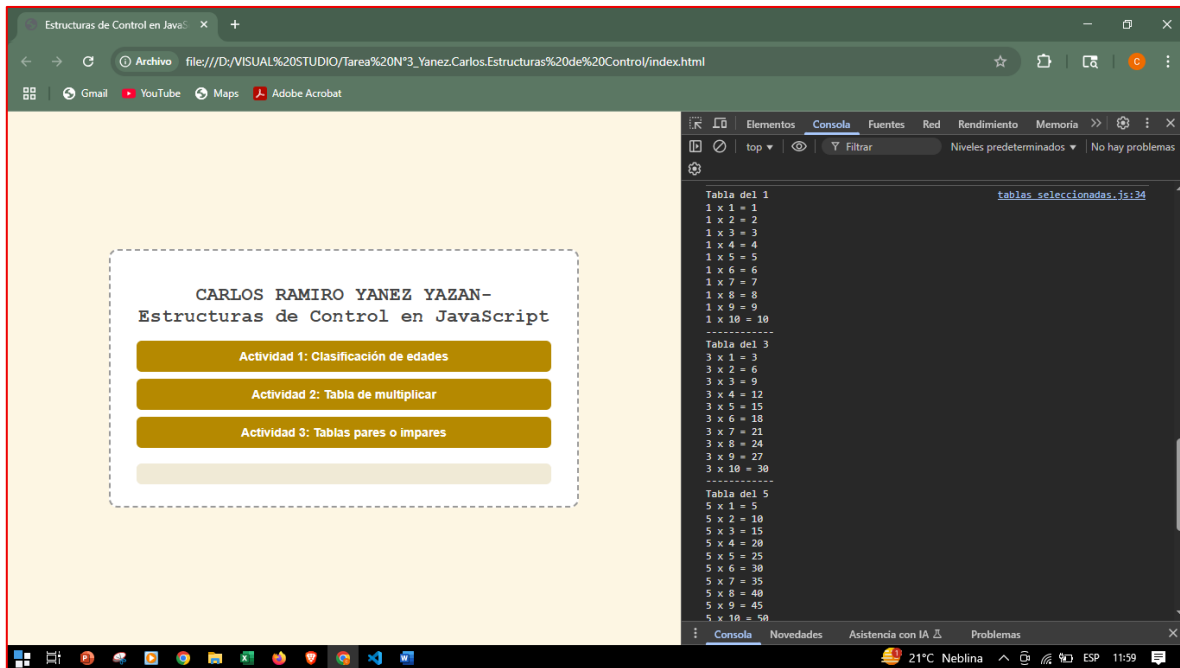
(Pares)



(Impares)



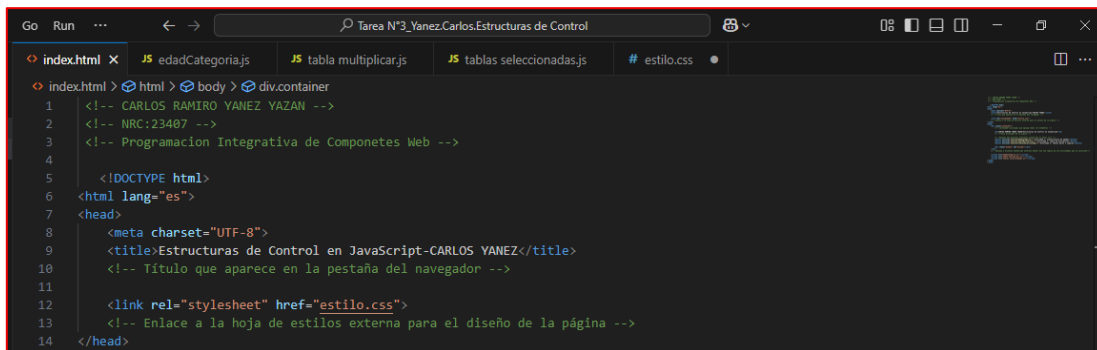
Los valores para la tabla impar se visualizan de la siguiente manera.



Una vez echo esto se logra visualizar todos los resultados obtenidos de cada una de las actividades establecidas cumpliendo con cada uno de los requerimientos.

4. Explicación breve del funcionamiento del archivo index.html y archivos js.

Estructura principal del archivo donde se define el idioma, titulo y la hoja de estilo externa que se vinculó.



Se define el contenedor el cual contendrá cada una de las funciones y elementos además de los botones que ejecutaran cada una de las funciones de JavaScript.

Adicional a esto, se valida los enlaces a archivos de JavaScript externos para la lógica de cada una de las actividades.

```

16 <div class="container">
17   <!-- Contenedor principal que agrupa todos los elementos -->
18
19   <h1>CARLOS RAMIRO YANEZ YAZAN-Estructuras de Control en JavaScript</h1>
20   <!-- Título principal de la página -->
21
22   <!-- Botones que ejecutan funciones JavaScript al hacer clic -->
23   <button onclick="ejecutarEdadCategoria()">Actividad 1: Clasificación de edades</button>
24   <button onclick="ejecutarTablaMultiplicar()">Actividad 2: Tabla de multiplicar</button>
25   <button onclick="ejecutarTablasSeleccionadas()">Actividad 3: Tablas pares o impares</button>
26
27   <div class="output" id="salida"></div>
28 </div>
29 <!-- Enlaces a archivos JavaScript externos donde cree la lógica de las actividades que se solicitan-->
30
31 <script src="edadCategoria.js"></script>
32 <script src="tabla multiplicar.js"></script>
33 <script src="tablas seleccionadas.js"></script>
34 </body>
35 </html>

```

Conclusiones

- La combinación de estructuras de control (if anidados, if, for) permite desarrollar programas con lógica condicional y repetitiva para resolver tareas específicas, como categorizar datos, generar tablas y filtrar resultados.
- Facilitar la interacción con el usuario a través de la solicitud de entrada (edad, número, tipo de tabla) hace que los programas sean más dinámicos y adaptables a diferentes necesidades.
- Al validar entradas se retoma una práctica fundamental para asegurar la robustez y el correcto funcionamiento de los programas, evitando errores causados por datos inesperados.

Recomendaciones

- Priorizar la claridad y legibilidad del código utilizando estructuras de control adecuadas y comentarios cuando sea necesario.
- Implementar mecanismos robustos de validación de entrada para anticipar y manejar posibles errores causados.
- Considerar el uso de funciones para encapsular tareas específicas lo que promueve la reutilización y simplifica la estructura general del programa.