**Nombre**: Michael Steven Jiménez Basante

**NRC:** 23407

**Materia:** PROGRAMACIÓN INTEGR COMPON WEB

**Docente:** [Vilmer David Criollo Chanchicocha](https://micampusvirtual.espe.edu.ec/user/view.php?id=27954&course=4058)

**Tema:** Estructuras de Control en JavaScript

**Taller 3**

1. **Descripción de la tarea:**

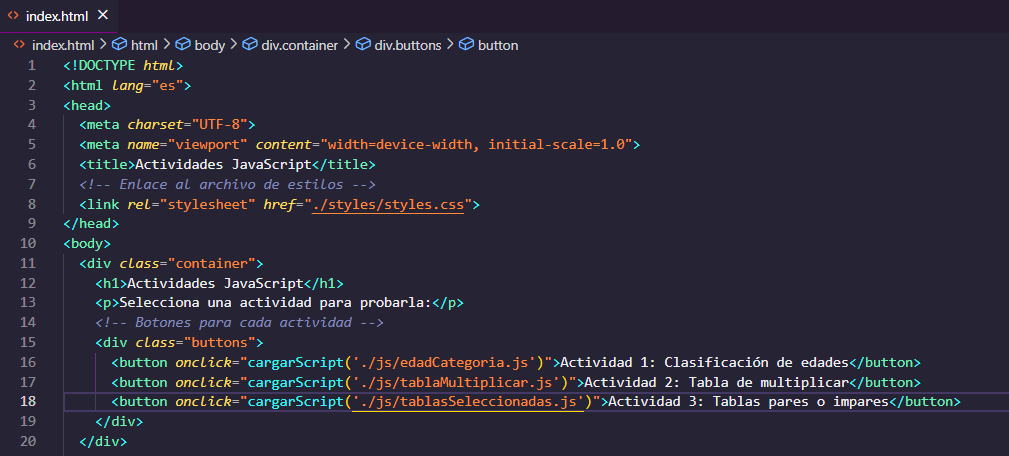
Desarrollar una interfaz web que permita ejecuitar 3 diferentes funciones programadas en un archivo de tipo .javascript aparte.

1. **Desarrollo:**
   1. **Enlace al GitHub:**

<https://github.com/maik151/ActividadEstructurasControl.git>

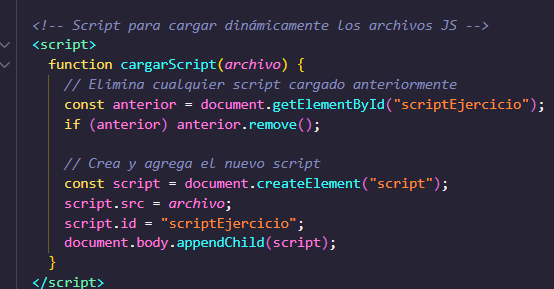
1. **Estructura del proyecto:**
   1. **Archivo *index.html:***

Para esta actividad se creo una estructura de botones, mediante al etiqueta *<button>* se inserta un objeto de tipo boton a la estructura de la pagina. Dentro de la etiqueta creamos un atributo *“onclick()”* el cual funciona como un evento de tipo “click”. el cual al presionar el boton definido, llama a la funcion “*cargarScript()”.*



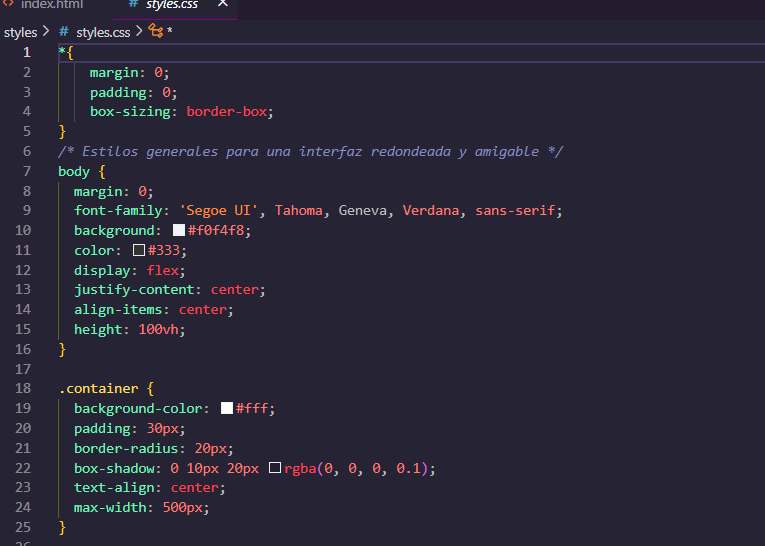
La funcion *“cargarscript()”* se define dentro de una etiqueta *<script>* (la cual nos permite ejecutar codigo de tipo javascript dentro de HTML). Dentro de la funcion de *“cargarscript()”*  definimos una funcion que toma como parametro un objeto de tipo archivo.js; luego se declara una constante la cual toma el valor de algun script de un archivo anterior que ya se haya ejecutado (toma el valor usando un selector por id). Luego se delcara un condicional *if* donde si existe un script anterior, se elimine usando el metodo de *“remove()”.*

Luego creamos una constante donde creamos un elemento de tipo script, a dicho script le adjuntamos el archivo que se pasa como parametro en la funcion, y se le asigna el id de “scriptEjercicio”. Luego usando el metodo de *“appendChild(script)”* agregamos la nueva etiqueta script al ultimo de nuestra estructura HTML.



* 1. **Archivo *styles.css:***

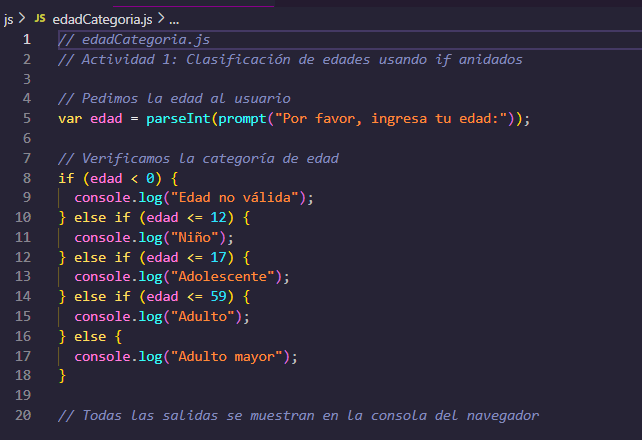
Dentro de este archivo definimos los estilos en base a las clases definidas en el HTML.



* 1. **Archivo *edadCategoria.js:***

Dentro de este archivo, definimos una variable con un scope global tipo ***var*** donde primero pasamos a un valor de entero (int) el valor obtenido por medio de un *“prompt()”,*  ese valor lo guardamos en una variable ***edad.***

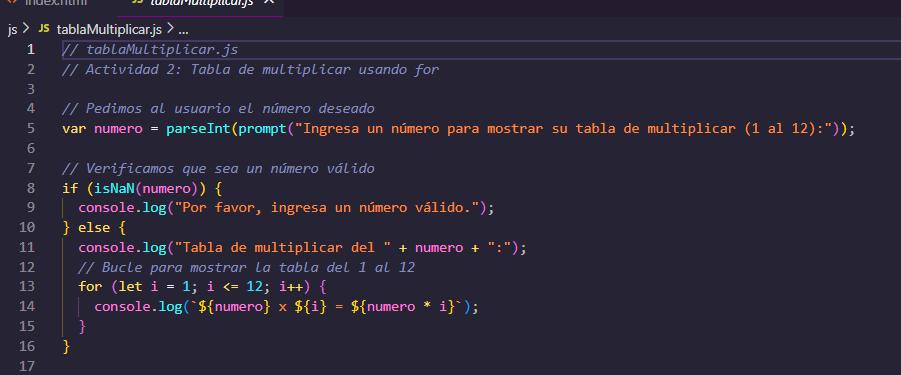
Dicha variable la pasamos por condicionales tipo IF para poder definir los rangos de las edades, segun eso, dependiendo de los intervalos, se muestra en consola a que grupo pertenece, si a niño. adolescente, adulto o adulto mayor.



* 1. **Archivo *tablaMultiplicar,js:***

Dentro de este archivo definimos igual una ***var***  de nombre ***numero,***  donde primeor validamos que el numero sea valido usando el metodo ***isNaN().*** Si la variable no es un numero entonces ejecuta en consola el mensaje de que se digite un numero valido, y si no, entonces se ejecuta un ciclo **FOR,** donde empezamos en 1, avanzamos hasta menor igual a 12 y avanzamos de 1 en 1.

El ciclo ejecuta un console.log donde usando una nomenclatura de backticks, creramos una plantilla usando la variable numero para por ejecutar las operaciones en la misma linea dle console.log (esto nos permite ahorrar lineas de codigo en tener que definir otra variable resultado, ejecutar la operacion y luego mostrarla, etc).



* 1. **Archivo *tablasSeleccionadas.js:***

De la misma forma declaramos una variable de scope global tipo ***var,*** de nombre ***tipo,***  donde esta va a tener el valor dado de un prompt, el cual se pasa a minusculas en el caso de que se haya escrito en mayusculas.Luego declaramos un if, donde usamos el operador de igualdad estricta (===), que compara tanto el valor como el tipo de dato, para verificar si la variable tipo es igual a "pares" o "impares". Esta doble condición se representa con el operador lógico || (OR).

Si se cumple alguna de estas dos condiciones, se ejecuta un ciclo for que comienza con un iterador i en 1, avanza de uno en uno (i++) hasta llegar a 10.

Dentro de ese mismo for, se declara otro if que evalúa si:

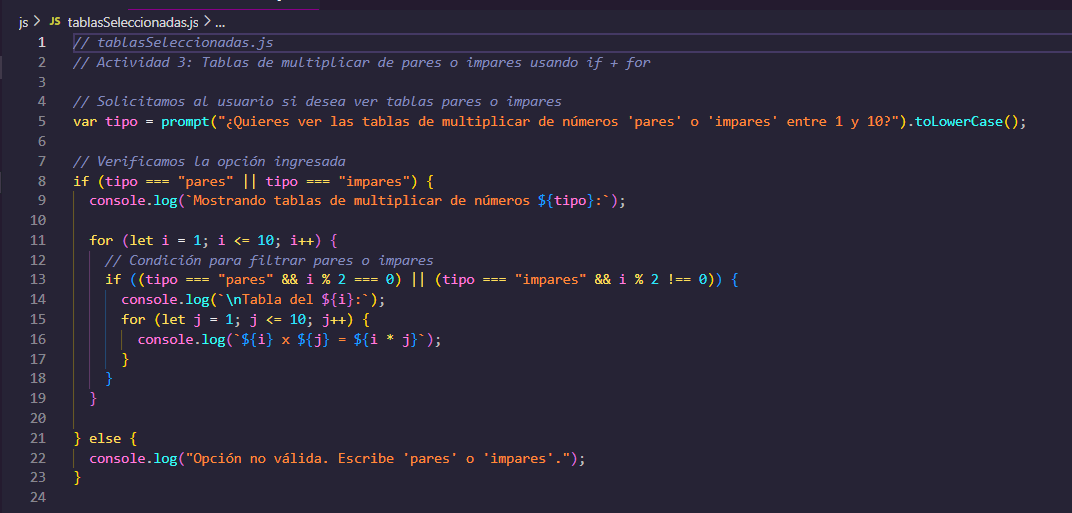
tipo es "pares" y i % 2 === 0 (es decir, si el número actual es par), o

tipo es "impares" y i % 2 !== 0 (es decir, si el número actual es impar).

Si la condición se cumple, se imprime en consola la tabla de multiplicar correspondiente al número actual.

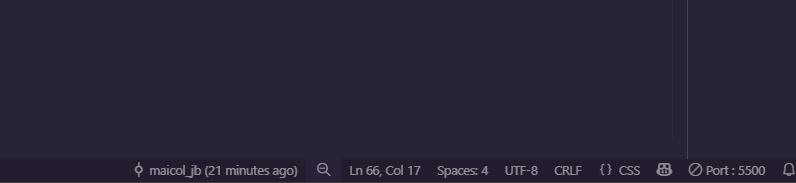
Para ello, se usa un segundo bucle for anidado que va desde 1 hasta 10, multiplicando el número actual (i) por cada uno de los valores del iterador j, e imprimiendo el resultado con formato:

i x j = resultado.



1. **Ejecucion del Proyecto**

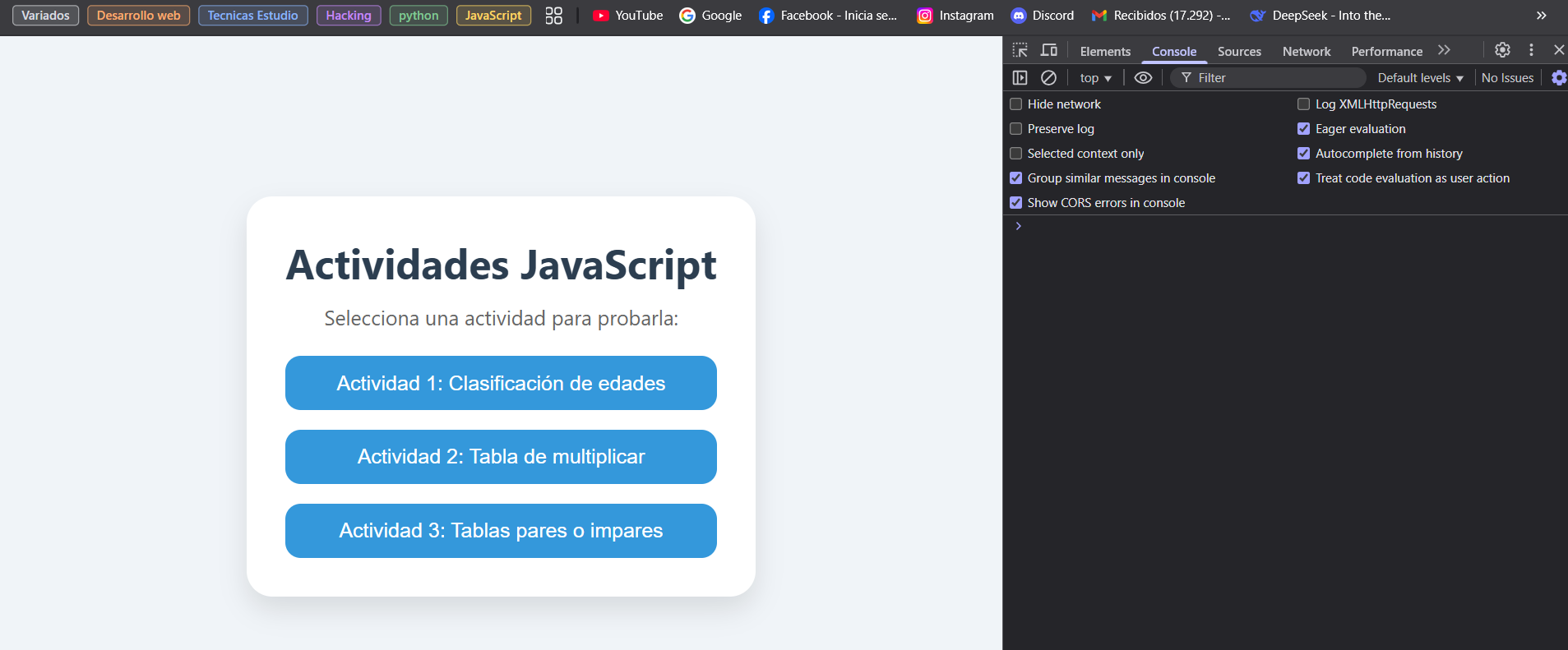
Para realizar la ejecucion del proyecto usamos la herramienta de *liveServer* la cual nos permite ejecutar el proyecto en un miniservidor que recepta las solicitudes http, estas solicitudes se hacen bajo el localhost y el puerto 5500.

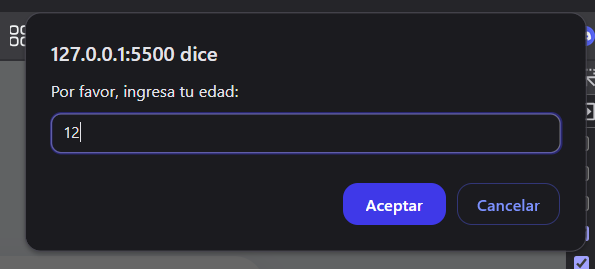


Al momento de ejecutar la herramienta tenemos que:

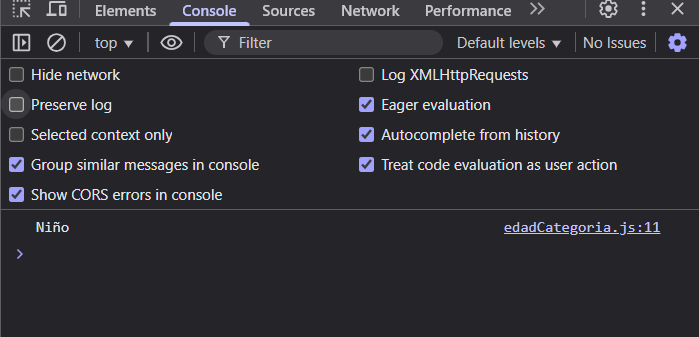


Podemos presion F12 o usar click derecho>inspeccionar para poder abrir la consola del navegador (consola donde corre codigo js)

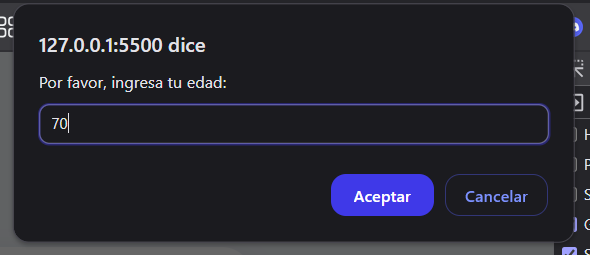


Ahora segun hagamos click en cada uno de los botones se ejecuta, la funcion definida. Para el caso de la primera funcion tenemos que:  
  


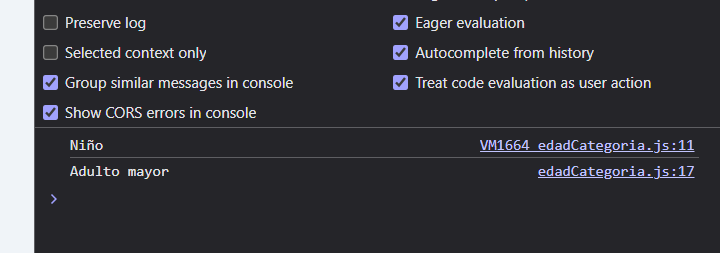
Se ejecuta el *prompt(),* donde introducimos un valor 12, y en la consola, nos arroja que pertenece a la categoria de niño:



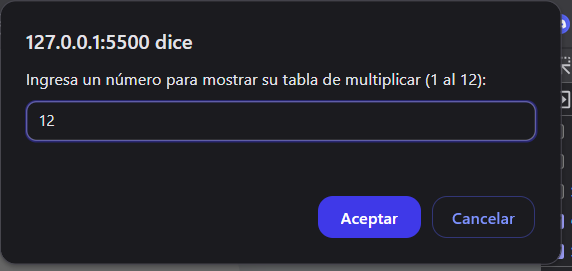
Si entroducimos otro valor la varaible se sobreescribe, en este caso usamos el valor 70

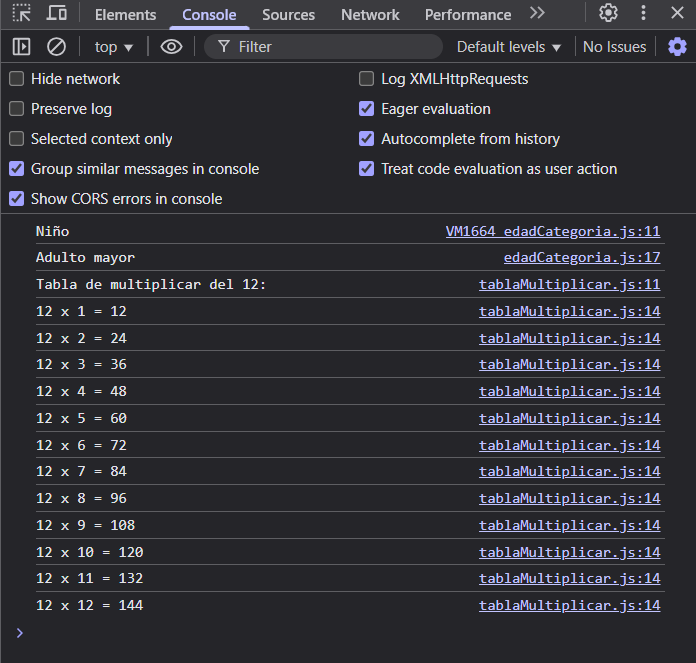


Nos arroja que pertenece a la categoria de adulto mayor:

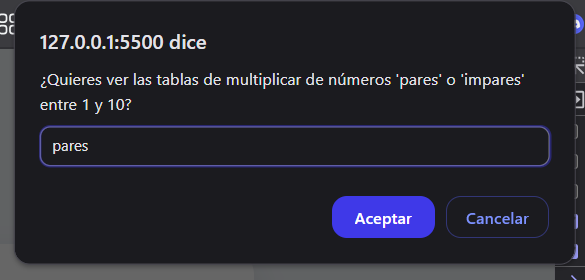


Para la segunda funcion se ejecuta otro ***prompt()***  donde tenemos que ingresar un numero que nos muestre la tabla de multiplicar de ese numero, este caso insertamos 12:



Donde la consola nos arroja la tabla de multiplicar basada en ese numero:  
  


Para el tercer metodo de la misma manerra obtenemos los datos desde un ***prompt().*** En este caso, debemos ingresar si queremos par o impar, tecleamos la palabra “pares”.



Dentro de la consola nos muestra las tablas de multiplicar de cada uno de los numero pares:  
  
