Desarrollo Web en Entorno Cliente



Javascript

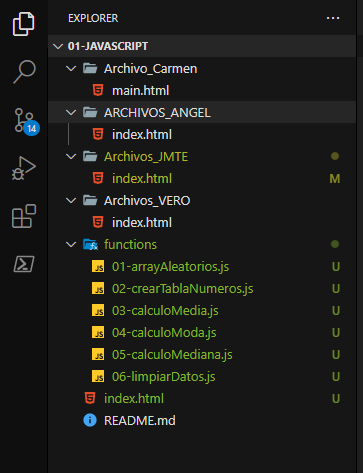
|  |
| --- |
| VERÓNICA BONIS MARTÍN |
| MARIA CARMEN CORREA HERAS |
| ÁNGEL SÁNCHEZ-SIERRA CRUZ |
| JOSÉ MARÍA TENREIRO EIRANOVA  https://github.com/ITTDAW/01-JavaScript.git |
|  |

**Requerimiento 1**

Realiza un programa en JavaScript que recorra un array de 10 números y calcule la media aritmética de diez números. Los números deben de estar comprendidos entre el 0 y 100, si se detectará algún número fuera de ese rango no se tendría en cuenta para su cálculo.

Cada integrante del equipo debe de proporcionar un algoritmo con la solución y se debe elegir la mejor solución de todas las propuestas con unas conclusiones.

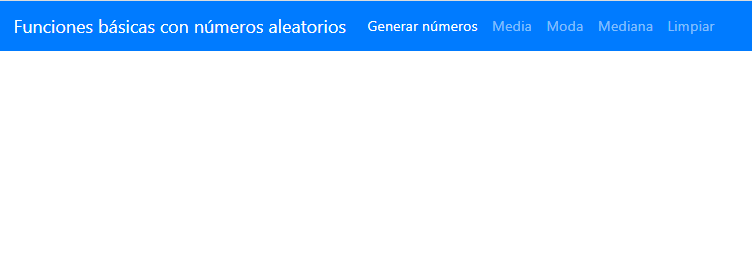
Nuestro proyecto tiene la siguiente presentación en el directorio de archivos:



Existen cuatro carpetas en donde los integrantes del grupo programan sus funciones y exponen la forma de realizar la primera parte del ejercicio. Después en la raíz del directorio nos encontramos con los archivos del grupo en los que se ha trabajado conjuntamente para llevar a cabo su desarrollo. El proyecto conjunto se ha decidido hacer una modularización del mismo para que trabajar en una función concreta sea mas fácil que introducir toda la información en un solo archivo. De esta manera, cada función será independiente y después se le llamara desde el archivo index.html cuando sea preciso. Se detalla más adelante el funcionamiento del proyecto.

Cada integrante del grupo ha realizado la primera parte del trabajo en un archivo index.html en su propio directorio en donde trabaja mediante los elementos  *HTML Script* (<script>) que se utilizan para insertar o hacer referencia a un script ejecutable dentro de un documento HTML. En ese elemento Script cada integrante, implementa su función para crear 10 números aleatorios e introducirlos en un array de números. Además, se calcula la media de estos números y después mediante un console.log se presenta estos datos en la consola del navegador para su visualización.

Cuando se ponen en común las ideas de la primera parte del trabajo, se comienza a trabajar en el proyecto común. Se decide hacer un diseño mínimo del html con una barra de navegación en donde facilitamos al usuario elegir que función quiere ejecutar:



Se le da un formato de estilo sencillo mediante bootstrap que ha sido cargado mediante link en el head del archivo html.

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

A su vez, hemos cargado todas las funciones que tenemos en archivos independientes mediante el elemento <script></script>

<!-- Este primer archivo JS, lo que nos hace es crear un array y rellenarlo con 10 numeros aleatorios de 0 a 100, tambien

ordena ese array de menor a mayor para una mejor lectura (y porque despues nos será necesario) -->

<script language="javascript" src="./functions/01-arrayAleatorios.js"></script>

<!-- Este archivo lee los numeros del array creado y los introduce en una tabla creada en JS mediante una

  función lo cual nos permite una mejor presentación de los mismos -->

<script language="javascript" src="./functions/02-crearTablaNumeros.js"></script>

<!-- Este archivo, lee el valor obtenido en la variable suma del primer archivo js y calcula la media de los numeros del array

mediante una función-->

<script language="javascript" src="./functions/03-calculoMedia.js"></script>

<!-- Este archivo lee los numeros del array creados y calcula la moda  mediante una función-->

<script language="javascript" src="./functions/04-calculoModa.js"></script>

<!-- Este archivo lee los numeros del array creados y calcula la mediana mediante una función -->

<script language="javascript" src="./functions/05-calculoMediana.js"></script>

<!-- Este archivo limpia los datos de la pantalla mediante una funcion -->

<script language="javascript" src="./functions/06-limpiarDatos.js"></script>

**Si el usuario pulsa la opción Generar números, hace llamamiento al evento onclick definido en el *Elemento HTML*Anchor <a> . Este evento a su vez hace llamamiento a la función crear()**

              <a class="nav-link" href="#"  onclick="crear()">Generar números <span class="sr-only" >(current)</span></a>

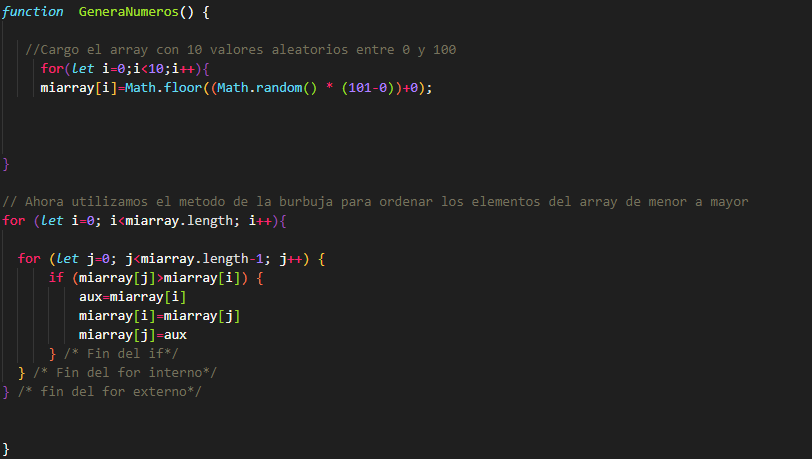
Esto dispara el que se ejecute la función crear() que tenemos en el archivo 02-crearTablaNumeros.js . En este archivo lo primero que se hace es llamar a la función GenerarNumeros() incluido en el archivo 01-arrayAleatorios.js

*function* crear(){

    //Llamamos a la funcion GenerarNumeros() para crear nuestro array

    GeneraNumeros();

Esta función GeneraNumeros() realiza su trabajo y carga en un array 10 números aleatorios y después los ordena de menor a mayor por el método de la burbuja según el código que está definido en ese archivo, se adjunta una captura para su visualización:



De esta manera ya tenemos un array de 10 números ordenados de menor a mayor (lo cual será indispensable después para el cálculo de otras funciones)

La función crear() además crea una tabla de las mismas columnas que el tamaño del array e introduce en cada celda el valor de cada elemento del array. Es solo un proceso para una presentación más visual en el html:

suma=0;

    /\*Creamos una variable llamada tabla e introducimos el codigo necesario para crear elementos en html "<loquesea>", para definir

    atributos de esa tabla se hace como se muestra en pantalla con border\*/

*let* tabla="<table border=\"2\">";

    //Recorremos el array

    for(*let* i=0;i<miarray.length;i++){

    //Le sumamos un td a cada posicion del array y le concatenamos el valor del array en esa posición

    tabla+="<td>"+ miarray[i];

    // Utilizo la variable suma como un acumulador de los elementos del array para calcular despues la media

    suma=suma+miarray[i]

    }

A su vez, aquí, utilizamos la variable suma utilizándola de acumulador para después calcular la media de los números. Acumula la suma de los elementos del array.

Para la presentación de estos datos en el html utilizamos El DOM que es una de las APIs más usadas en la Web, pues permite ejecutar código en el navegador para acceder e interactuar con cualquier nodo del documento.

Todo lo que está dentro de un documento HTML es un objeto en javascript, las etiquetas que están dentro del archivo es

conocido como DOM (Document Object Model). Para poder manipular los objetos del dom con javascript es necesario que obtengas

 el elemento, para eso se declara un id, ya que un id no se puede repetir, solo existirá un único id por elemento.

 Con el  innerHTML se añade html dentro de un elemento, al ponerle "" , lo que hacemos es dejar en blanco ese elemento

 de esta forma es como si reseteamos los contenidos de esos elementos (borramos lo que habia)\*

/\*Ahora presentamos en el elemento con id resultado el contenido de la tabla y reseteamos los valores de los otros elementos

    para que cuando se pulse todo el rato en generar numeros no sigan los resultados anteriores en pantalla\*/

    document.getElementById("resultado").innerHTML=tabla;

    document.getElementById("resultadomedia").innerHTML="";

    document.getElementById("resultadomoda").innerHTML="";

    document.getElementById("resultadomediana").innerHTML="";

De esta manera, ya tendríamos nuestros números aleatorios presentados en la pantalla en una tabla y ordenados de menor a mayor.

Si el usuario pulsa en la barra la opción media, se realiza una llamada al evento onclick que ejecuta la función media()

                <a class="nav-link" href="#" onclick="media()">Media</a>

Entonces llamaremos a la función que está cargada en el archivo 03-calculoMedia.js

Esta función es muy sencilla, con el dato que obtenemos del sumatorio suma, lo dividimos entre 10 o el tamaño del array y lo presentamos en pantalla. Además añadimos que si el array está vacío, nos presente un mensaje de que no hay números generados por si el usuario pulsa media antes de generar números.

*function* media(){

*let* media= suma/10;

  /\*Para evitar el undefined, le decimos que si no existe array compuesto de numeros aun, nos presente en el elemento correspondiente

  un texto explicativo\*/

  if (miarray.length==0){

    document.getElementById("resultadomedia").innerHTML= "No hay números generados" ;

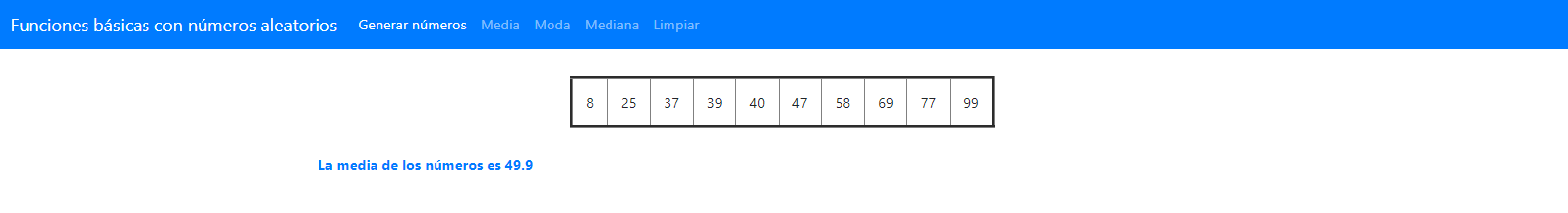
}else{

  // Si todo es correcto en el elemento con id resultadomedia, presentamos el valor de la media

   document.getElementById("resultadomedia").innerHTML= "La media de los números es "+ media ;

}

}



**Si el usuario pulsa la opción Moda, entonces se hace llamada al evento onclick que dispara la función moda():**

                <a class="nav-link" href="#" onclick="moda()">Moda</a>

Esta función se encuentra dentro del archivo 04-calculoModa.js :

Sin duda esta es la función más difícil de calcular ya que pueden existir muchos supuestos que hay que contemplarlos para calcularla:

El supuesto más fácil es el del siguiente ejemplo :

1,2,3,4,4,5,6,7,8

En este ejemplo no habría discusión y la moda seria el numero 4 ya que es el número que se repite únicamente.

El problema al implementar esta función es que puede darse el caso de que la moda sean dos números porque se repiten el mismo número de veces como por ejemplo

1,2,2,3,4,4,5,6,7,8

En este caso la moda serían los números 2 y 4

Por ello, el código de esta función se nos hizo un poco difícil pero al final entre todos conseguimos el resultado correcto y lo que pretendíamos.

Para calcular el valor de las repeticiones vamos a utilizar el método de enfrentar el array contra si mismo mediante una matriz y de esta forma obtener el número de repeticiones. El código se encuentra todo comentado, explicando cada paso dado para conseguir nuestro propósito que es no dejar fuera ningún caso que puede darse en la moda.

*function* moda(){

    /\*Declaramos dos variables, la primera establece el numero maximo de veces que se repite un numero y que obtenemos

    cuando ya hemos recorrido toda nuestra matriz, como vamos a comparar una matriz formada por los valores del array en las

    cabeceras, lo establecemos a 1 porque ese valor lo va a dar siempre aunque no haya moda y hay que descartarlo.

      y la segunda es el valor de la moda que obtendremos en la posicion correspondiente\*/

*let* maximaVecesQueSeRepite = 1;

*let* modavalor =0;

    /\*Declaramos una variable string cadena, la razón es porque si hay mas de un numero que se repite las mismas veces que otro

    la moda estaria representada por mas de un numero y lo concatenaremos en este string\*/

*let* cadena=""

    //Establecemos una variable booleana porque si no hay ningun numero repetido, vamos a escribir un texto

*let* sihay=false;

  //Establecemos una matriz del mismo numero de filas que de columnas y con el valor de los terminos del array

  //Recorremos la fila i

  for(*let* i=0; i<miarray.length; i++){

    //Cada vez que volvemos a i reseteamos el valor de las veces que se repite

*let*  vecesQueSeRepite = 0;

   //ahora recorremos la fila j

                for(*let* j=0; j<miarray.length; j++){

                  /\*cada vez que el termino del array en la posición i es igual al termino del array en la posición j,

                  sumamos una unidad a vecesQueSeRepite\*/

                    if(miarray[i] == miarray[j])

                        vecesQueSeRepite++;

                  }

        /\*Salimos de ese segundo for y comparamos el termino vecesQueSeRepite con maxiamVecesQueSeRepite:

      ---->  primero lo comparamos si el numero vecesQueSeRepite que acabamos de obtener es igual a maximaVecesQueSeRepite\*/

        if(vecesQueSeRepite == maximaVecesQueSeRepite){

         //Aqui entraria tambien cuando no hay moda, pero no pone la variable sihay=true y por lo tanto sabemos que no hay moda

          /\*Si el numero de vecesQueSeRepite es igual a maximaVecesQueSeRepite , entonces, primero tenemos que comparar

          que no sea nuestro numero repetido que ya hemos añadido a la cadena, aquí es importante tener el array ordenado,

          sino nuestro código no funcionaria\*/

                      if (miarray[i]==miarray[i-1]){

                      }else{

          /\*Si es un numero nuevo que tiene las mismas repeticiones que otro que ya teniamos añadido, lo añadimos a nuestra cadena\*/

                                 modavalor=miarray[i];

                                cadena= cadena + modavalor + " , "

                                maximaVecesQueSeRepite = vecesQueSeRepite;

                       }

        /\*En el momento que encontramos un valor para vecesQueSeRepite mayor que el maximaVecesQueSeRepite. \*/

        } else if (vecesQueSeRepite>maximaVecesQueSeRepite){

                 //Establecemos que si hay moda, si hay numeros repetidos, en el primer numero ya entra en este bucle y lo establece

                          sihay=true;

                         if (miarray[i]==miarray[i-1]){

                        }else{

        /\*Y por ultimo al ser un valor mayor, eso quiere decir que tenemos un nuevo valor unico de moda y hay que resetear la cadena

        para solo incluir este valor en ella, y establecemos este como el valor de maximaVecesQueSeRepite\*/

                                   cadena="";

                                    modavalor=miarray[i];

                                    maximaVecesQueSeRepite = vecesQueSeRepite;

                                    cadena= cadena + modavalor + " , "

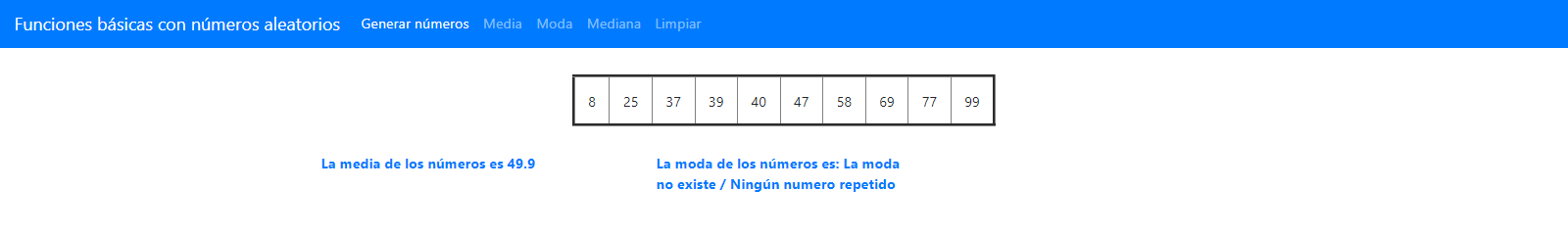
                        }

         }

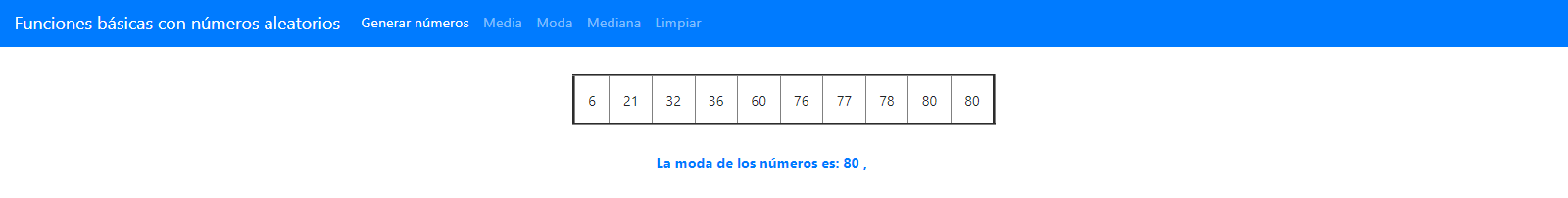
  }

A continuación, dejamos unos ejemplos de la ejecución de nuestro código para los casos comentados:

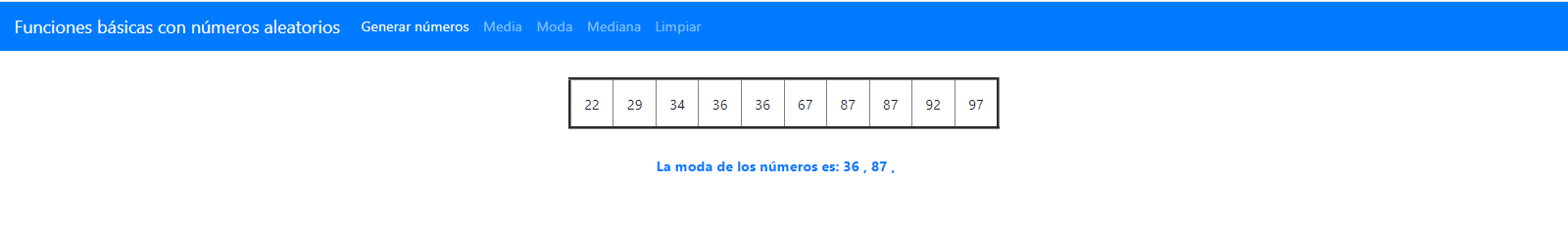
En el primero, la moda no existe, no hay ningun numero repetido:

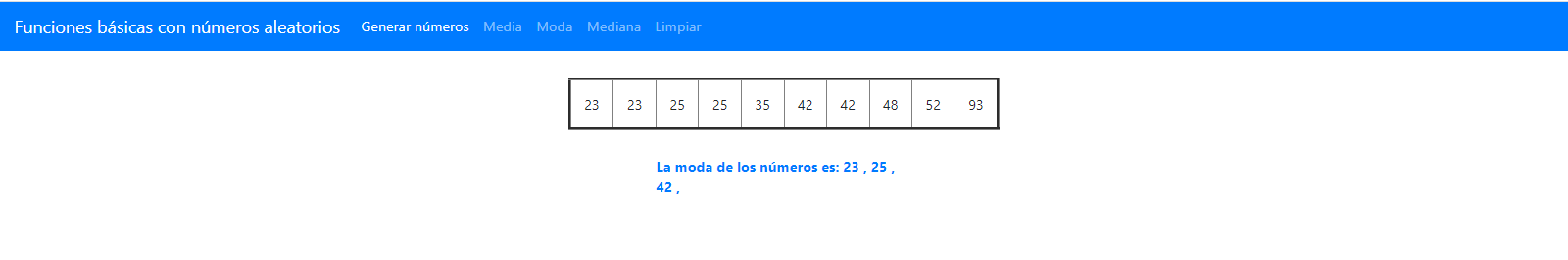


En el segundo, el caso más simple, existe un único número que se repite :



Y el tercer caso, el más complicado y por el cual la implementación de la moda se extiende en nuestro código, que existan dos números o más que se repiten las mismas veces:





**Si el usuario pulsa la opción Mediana, en este caso se hace llamamiento al evento onclick que ejecuta la función mediana()**

                <a class="nav-link" href="#" onclick="mediana()">Mediana</a>

Entonces llamaremos a la función que está cargada en el archivo 05-calculoMediana.js

Esta función es sencilla de utilizar, pero siempre hay que llevar el array ordenado para que funcione. Recordamos que nosotros ya ordenamos el array justo después de crearlo. De esta manera la implementación de la función es muy sencilla:

*function* mediana(){

    //Nosotros ya tenemos el array ordenado de menor a mayor y asi lo recibimos, si no estuviera ordenado, habría que hacerlo.

    /\*Si el numero de elementos del array es par, definimos dos posiciones, la posicion miarray.lenght/2 y la anterior,

    despues miramos el valor del elemento del array en esa posicion, sumamos ambos y la mitad seria la el valor de la mediana.\*/

    if (miarray.length%2==0) {

*var* pos1=miarray.length/2

*var* pos2=pos1-1

*var* elem1=miarray[pos1]

*var* elem2=miarray[pos2]

*var* mediana=(elem1+elem2)/2

    /\*Si el numero de elementos es impar, calculamos una nueva posicion restando uno al numero de elementos del array y lo dividimos entre

    dos, el valor de ese elemento del array en esa posicion es la mediana\*/

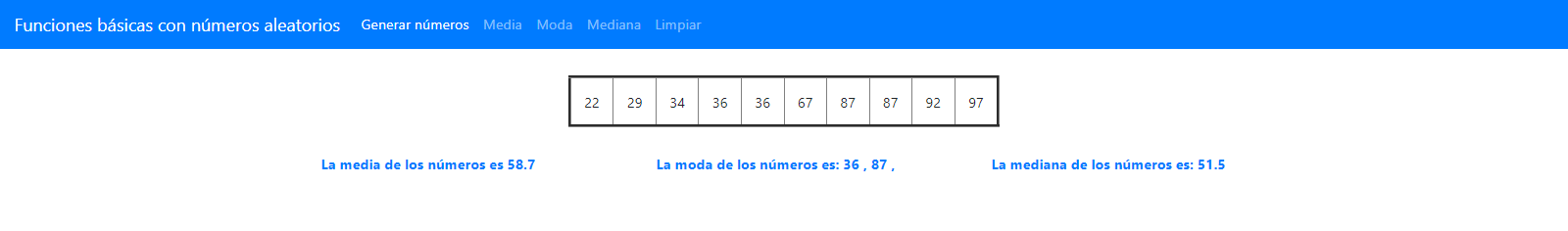
    } else {

*var* pos\_central=(miarray.length-1)/2

*var* mediana=miarray[pos\_central]

    }

Y después solamente nos queda presentarla en pantalla mediante el uso del DOM que ya explicamos anteriormente.



Por ultimo si el usuario elige la opción limpiar hacemos llamamiento al evento onclick que ejecuta la función limpiar()

                <a class="nav-link" href="#" onclick="limpiar()">Limpiar</a>

Esta función lo único que realiza es mediante el uso del DOM la limpieza de todo el contenido de los elementos que le especificamos en los id correspondientes:

*function* limpiar(){

    miarray=[];

    document.getElementById("resultado").innerHTML="";

    document.getElementById("resultadomedia").innerHTML="";

    document.getElementById("resultadomoda").innerHTML="";

    document.getElementById("resultadomediana").innerHTML="";

    }

**IMPORTANTE: TODO EL CODIGO SE ENCUENTRA EXPLICADO MEDIANTE COMENTARIOS, ESTE DOCUMENTO ES UNA INTRODUCCIÓN PARA DETALLAR LA ESTRUCTURA DE LOS DIRECTORIOS Y LOS PASOS DADOS EN EL PROYECTO DE UNA FORMA RAPIDA Y QUE SE PUEDA COMPROBAR MEDIANTE ALGUNOS EJEMPLOS SU CORRECTO FUNCIONAMIENTO.**