

Informe de Laboratorio 03

Tema: JavaScript

| Nota |
|------|
| |

| Estudiante | Escuela | Asignatura |
|--|--|--|
| Victor Gonzalo Maldonado Vilca vmaldonadov@unsa.edu.pe | Escuela Profesional de Ingeniería de Sistemas | Programación Web 2 Semestre: III Código: 1702122 |

| Tarea | Tema | Duración |
|-------|------------|----------|
| 03 | JavaScript | 2 horas |

| Semestre académico | Fecha de inicio | Fecha de entrega |
|--------------------|------------------------|-----------------------|
| 2024 - A | Del 9 de abril de 2024 | Al 15 de mayo de 2024 |

1. Tarea

Ejercicios Propuestos:

- Escriba una función que reciba el número de día de la fecha actual `new Date()` y devuelva el texto del día de la semana correspondientes. Por ejemplo si recibe 0, devolvería "Domingo".
- Escriba una página web que reciba un texto y al presionar un botón muestre el mismo texto invertido en otra sección (div). Por ejemplo si se escribe "Hola", se mostraría como "aloH".
- Escribir una página que muestre cuántos días faltan para el día de Arequipa!
- Escribir un página que reciba el URL de la sesión de google meet de hoy y devuelva el código de la sesión sin guiones separadores.
- Escribir una página que permita calcular la suma de todos los valores de una tabla de valores dinámica. La idea es crear una página web con un formulario que te permita decir cuantos valores tendrá la tabla, luego, al enviar el formulario la tabla se debe crear dinámica y aleatoriamente, junto con otro botón de envió para calcular la suma.
- En su tarea deberán implementar las siguientes páginas.
 - Pagina1.html - Cree una página web con un texto y dos botones (al estilo del ejemplo del foco que se enciende y apaga) que permitan cambiar el tamaño de la letra de un texto, intente hacerlo también con los colores.
 - Pagina2.html - Cree una página web que permita realizar las operaciones aritmética, lógicas y de bits básicas, de manera dinámica(se podrá elegir cualquier operador) y se trabajará con dos argumentos.
- Resolver los 67 ejercicios de javascript en [w3schools.com](https://www.w3schools.com) y subir un pantallazo con su nombre y apellido.

2. Entregables

- Informe de trabajo
- URL: Repositorio de GitHub
- URL: Vídeo en Youtube

3. Equipos, materiales y temas utilizados

- HTML
- CSS
- JavaScript
- w3schools
- GitHub

4. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- <https://github.com/Victor-Gonzalo-Maldonado-Vilca/pw2-lab03.git>

5. URL de Vídeo en Youtube

- URL de youtube, donde estará el video explicativo de los ejercicios.
- <https://www.youtube.com/watch?v=-BQzJIAwX4s>

6. Desarrollo del trabajo

6.1. Ejercicio 1

- Colocando en un Array los valores de los días, mediante la presente funcion que recibira como parámetro un entero se logrará retornar un string que sera el día correspondiente.

Listing 1: Script 1 - parte 1

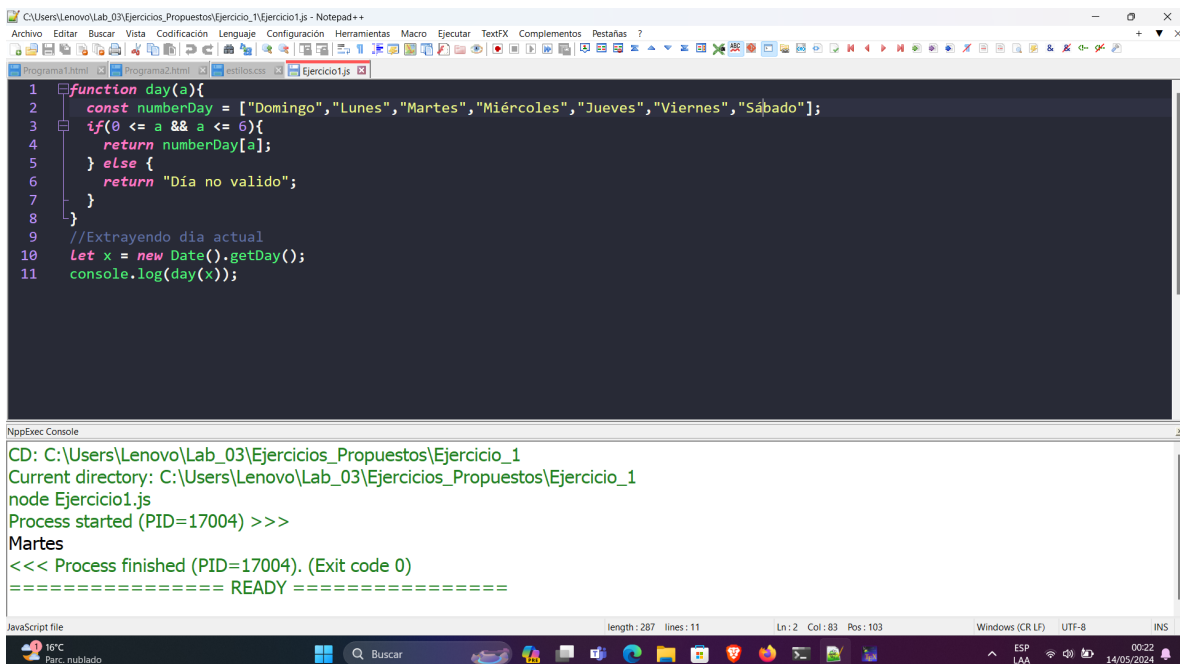
```
1 function day(a){  
2   const numberDay = ["Domingo", "Lunes", "Martes", "Mircoles", "Jueves", "Viernes", "Sbado"];  
3   if(0 <= a && a <= 6){  
4     return numberDay[a];  
5   } else {  
6     return "Da no valido";  
7   }  
8 }
```

- Extrayendo el día actual usando la clase Date y sus métodos correspondientes, además se llama a la anterior función

Listing 2: Script 1 - parte 2

```
9 //Extrayendo dia actual
10 let x = new Date().getDay();
11 console.log(day(x));
```

6.1.1. Ejecución



The screenshot shows a Notepad++ window with the following JavaScript code:

```
1 function day(a){
2   const numberDay = ["Domingo", "Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado"];
3   if(0 <= a && a <= 6){
4     return numberDay[a];
5   } else {
6     return "Día no valido";
7   }
8 }
9 //Extrayendo dia actual
10 let x = new Date().getDay();
11 console.log(day(x));
```

Below the code editor is a terminal window showing the execution of the script:

```
CD: C:\Users\Lenovo\Lab_03\Ejercicios_Propuestos\Ejercicio_1
Current directory: C:\Users\Lenovo\Lab_03\Ejercicios_Propuestos\Ejercicio_1
node Ejercicio1.js
Process started (PID=17004) >>>
Martes
<<< Process finished (PID=17004). (Exit code 0)
===== READY =====
```

Figura 1: Ejecución - Ejercicio 1

6.2. Ejercicio 2

6.2.1. HTML

- Etiqueta head donde se encontrarán las etiquetas meta, el título y los estilos

Listing 3: HTML - parte 1

```
3 <head>
4   <title>Ejercicio 2</title>
5   <meta charset="UTF-8">
6   <meta name="author" content="Victor Gonzalo Maldonado Vilca"/>
7   <link rel="stylesheet" href="css/estilos.css"/>
8 </head>
```

- Agregar un formulario para obtener el texto que ingrese el usuario

Listing 4: HTML - parte 2

```
12 <form id="formulario">
13   <label class="label" for="texto"><b>Texto:</b></label>
14   <input class="text" type="text" id="texto" name="texto"/>
15
16   <input class="invertir" type="submit" value="Invertir"/>
17 </form>
```

- Este código JavaScript escucha el evento de envío de un formulario ('submit'). Al recibir este evento, evita la acción predeterminada, invierte una cadena de texto ingresada por el usuario y muestra el resultado en la página web.

Listing 5: Script 2

```
24 <script>
25   document.getElementById("formulario").addEventListener('submit', function(event){
26     event.preventDefault();
27     const cadena = document.getElementById("texto").value;
28     function invert(str){
29       return str.split('').reverse().join('');
30     }
31     document.getElementById("inver").textContent = invert(cadena);
32   });
33 </script>
```

6.2.2. CSS

- Agregar estilos a la página.

Listing 6: Estilos

```
1 body{
2   font-size: 25px;
3 }
4
5 .label{
6   color: red;
7 }
8
9 .text{
10  border-radius: 10px;
11  padding-left: 15px;
12  border: 1px solid black;
13 }
14
15 .invertir{
16  background: #b2494e;
17  border: 1px solid #7c3336;
18  border-radius: 10px;
19  color: white;
20 }
21
22 .invertir:hover{
```

```

23     background: #c16d71;
24     border: 1px solid #b2494e;
25 }
26
27 .respuesta #invertir{
28     text-align: center;
29     color: green;
30     height: 10px;
31     margin-top: 0px;
32 }
33
34 #inciso{
35     margin-bottom: 0px;
36     color: green;
37     text-align: center;
38 }

```

6.2.3. Ejecución



Figura 2: Ejecución - Ejercicio 2

6.3. Ejercicio 3

6.3.1. HTML

- Etiqueta head donde se encontrarán las etiquetas meta, el título y los estilos

Listing 7: HTML - parte 1

```

3 <head>
4   <title>Ejercicio 3</title>
5   <meta charset="UTF-8">
6   <meta name="author" content="Victor Gonzalo Maldonado Vilca"/>
7   <link rel="stylesheet" href="css/estilos.css"/>
8 </head>

```

- Agregar un formulario para poder realizar un evento al presionar el botón

Listing 8: HTML - parte 2

```
12 <form id="formulario">
13   <label class="label" for="faltan">Cuntos das faltan para el da de
      Arequipa?</label>
14   <input class="boton" type="submit" value="Faltan!!"/>
15 </form>
```

- En esta parte, es donde mediante con el script se logrará colocar la respuesta correcta, en este caso los días que faltan para el día de Arequipa.

Listing 9: HTML - parte 3

```
18 <div>
19   <p id="fecha"><p>
20 </div>
```

- Este script JavaScript captura el evento de envío de un formulario con el ID "formulario". Al recibir este evento, calcula la diferencia en días entre una fecha específica ('2024-08-15') y la fecha actual. Luego, muestra esta diferencia de días en un elemento del documento con el ID "fecha".

Listing 10: Script 3

```
21 <script>
22   document.getElementById("formulario").addEventListener('submit', function(evento){
23     event.preventDefault();
24     const fechaArequipa = new Date('2024-08-15');
25
26     const fechaActual = new Date();
27
28     const fechaDiferencia = fechaArequipa - fechaActual;
29
30     document.getElementById("fecha").textContent =
      (Math.ceil(fechaDiferencia/(1000*60*60*24)));
31   });
32 </script>
```

6.3.2. CSS

- Agregar estilos a la página.

Listing 11: Estilos

```
1 body{
2   font-size: 25px;
3 }
4
5 .contenedor label{
6   color: green;
7 }
8
9 .contenedor input{
10  background: #b2494e;
```

```
11 border-radius: 10px;
12 color: white;
13 border: 1px solid #7c3336;
14 }
15
16 .contenedor input:hover{
17     background: #c16d71;
18     border: 1px solid #b2494e;
19 }
20
21 #fecha {
22     text-align: center;
23     color: blue;
24 }
```

6.3.3. Ejecución

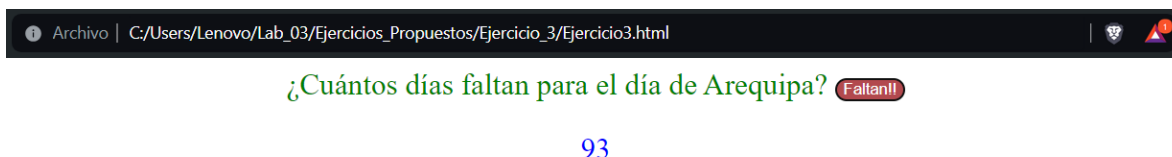


Figura 3: Ejecución - Ejercicio 3

6.4. Ejercicio 4

6.4.1. HTML

- Etiqueta head donde se encontrarán las etiquetas meta, el título y los estilos

Listing 12: HTML - parte 1

```
3 <head>
4   <title>Ejercicio 4 </title>
5   <meta charset="UTF-8">
6   <meta name="author" content="Victor Gonzalo Maldonado Vilca"/>
7   <link rel="stylesheet" href="css/estilos.css"/>
8 </head>
```

- Agregar un formulario donde el usuario colocara el enlace meet.

Listing 13: HTML - parte 2

```
11 <form id="formulario">
12   <label class="label" for="enlace"><b>URL meet:</b></label>
13   <input type="text" id="enlace" name="enlace"></input>
14   <input type="submit" class="boton" value="Generar"/>
15 </form>
```

- En esta parte, es donde mediante con el script se logrará colocar la respuesta correcta, en este caso el código del enlace meet.

Listing 14: HTML - parte 3

```
17 <div id="resultado">
18   <p><b>Codigo :</b></p>
19   <p id="codig"><p>
20 </div>
```

- Este script JavaScript captura el envío de un formulario y evita su acción predeterminada. Extrae un código específico de una URL ingresada por el usuario, eliminando los primeros 24 caracteres y los guiones. Luego, muestra este código en la página.

Listing 15: Script 4

```
21 <script>
22   document.getElementById("formulario").addEventListener('submit', function(evento){
23     evento.preventDefault();
24     const url = document.getElementById("enlace").value;
25     function obtenerCodigo(enlace){
26       codigo = enlace.substring(24,enlace.length);
27       if (codigo){
28         return codigo.replace(/-/g, '');
29       } else {
30         return "No hay cdigo";
31       }
32     }
33     document.getElementById("codig").textContent = obtenerCodigo(url);
34   });
35 </script>
```

6.4.2. CSS

- Agregar estilos a la página.

Listing 16: Estilos

```
1 body{
2   font-size: 25px;
3 }
4
5 .contenedor{
6   width: 470px;
7   align-items: center;
```



```
8   margin: auto;
9   }
10  .boton{
11    border-radius: 10px;
12    background: #a04146;
13    border: 1px solid #60272a;
14    color: white;
15  }
16
17  .boton:hover{
18    background: #b3666a;
19    border: 1px solid #a04146;
20  }
21
22  .label{
23    color: red;
24  }
25
26  #enlace{
27    border-radius: 10px;
28    width: 250px;
29    border: 1px solid black;
30    padding-left: 15px;
31  }
32
33  #resultado{
34    width: 420px;
35    margin: auto;
36    color: red;
37    text-align: center;
38  }
39
40  #codig{
41    color: green;
42  }
```

6.4.3. Ejecución



Figura 4: Ejecución - Ejercicio 4

6.5. Ejercicio 5

6.5.1. HTML

- Etiqueta head donde se encontrarán las etiquetas meta, el título y los estilos

Listing 17: HTML - parte 1

```
3 <head>
4   <title>Ejercicio 5 </title>
5   <meta charset="UTF-8">
6   <link rel="stylesheet" href="css/estilos.css"/>
7 </head>
```

- Agregar un formulario donde el usuario colocara una cantidad, para que se cree una tabla con valores numéricos aleatorios en cada celda.

Listing 18: HTML - parte 2

```
11 <form id="formulario">
12   <label for="cantidad">Cantidad: </label>
13   <input type="number" id="cantidad" name="cantidad" min="1"></input>
14   <input class="boton" type="submit" value="Generar"/>
15 </form>
```

- En esta parte, es donde mediante con el script se logrará colocar la respuesta correcta, en este caso tanto la tabla como la suma.

Listing 19: HTML - parte 3

```
17 <div id="contenedor">
18 </div>
19 <div id="resultado">
20 </div>
```

- Este script en JavaScript captura el envío de un formulario y evita su acción predeterminada. Luego, crea una tabla con números aleatorios en filas de 50 elementos cada una, basada en la cantidad ingresada por el usuario. Finalmente, agrega un botón "Sumar" para realizar una acción adicional al hacer clic en él.

Listing 20: Script 5 - parte 1

```
21 <script>
22   document.getElementById("formulario").addEventListener('submit', function(evento){
23     evento.preventDefault();
24     const cantidad = parseInt(document.getElementById("cantidad").value);
25     const tabla = document.getElementById("contenedor");
26     tabla.innerHTML = "<table id='tabla'
27                       border='1'><thead><caption>Valores</caption></thead><tbody
28                       style='text-align: center'></tbody></table>";
29     const cuerpoTabla =
30       document.getElementById("tabla").getElementsByTagName("tbody")[0];
31     let i = 1;
32     while (i <= cantidad) {
33       let fila = document.createElement("tr");
34       for (let c = 0; c < 50; c++) {
```

```
32     let aleatorio = Math.floor(Math.random() * 100) + 1;
33     let cell = document.createElement("td");
34     cell.textContent = aleatorio;
35     fila.appendChild(cell);
36     i++;
37     if (i > cantidad) {
38         break;
39     }
40 }
41 cuerpoTabla.appendChild(fila);
42 }
43
44 tabla.innerHTML += "<br><button class='boton'
45     onclick='sumarValores()'>Sumar</button>"
46 );
```

- La función suma los valores numéricos de una tabla en la página web y muestra el resultado en verde en el elemento con el ID resultado”.

Listing 21: Script 5 - parte 2

```
46 function sumarValores(){
47     const valores =
48         document.getElementById("tabla").getElementsByTagName("tbody")[0].getElementsByTagName("td");
49     let suma = 0;
50     for(let i = 0; i < valores.length; i++){
51         const valor = parseInt(valores[i].textContent);
52         suma += valor;
53     }
54     const res = document.getElementById("resultado");
55     res.innerHTML = '<p style="color: green">La Suma es: ' + suma + '</p>';
56 }
57 </script>
```

6.5.2. CSS

- Agregar estilos a la página.

Listing 22: Estilos

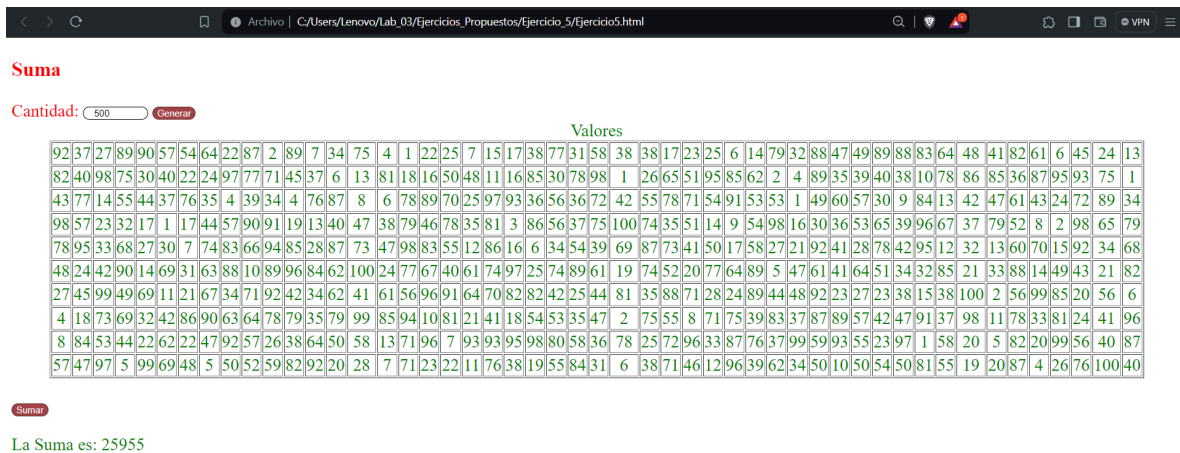
```
1 body{
2     font-size: 25px;
3     color: red;
4 }
5
6 #tabla{
7     margin: auto;
8     color: green;
9 }
10
11 .boton{
12     border-radius: 10px;
13     background: #a04146;
14     border: 1px solid #60272a;
15     color: white;
```

```

16 }
17
18 .boton:hover{
19     background: #b3666a;
20     border: 1px solid #a04146;
21 }
22
23 #cantidad{
24     border-radius: 10px;
25     border: 1px solid black;
26     padding-left: 15px;
27     width: 75px;
28 }

```

6.5.3. Ejecución



Suma

Cantidad:

Valores

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|-----|----|
| 92 | 37 | 27 | 89 | 90 | 57 | 54 | 64 | 22 | 87 | 2 | 89 | 7 | 34 | 75 | 4 | 1 | 22 | 25 | 7 | 15 | 17 | 38 | 77 | 31 | 58 | 38 | 38 | 17 | 23 | 25 | 6 | 14 | 79 | 32 | 88 | 47 | 49 | 89 | 88 | 83 | 64 | 48 | 41 | 82 | 61 | 6 | 45 | 24 | 13 |
| 82 | 40 | 98 | 75 | 30 | 40 | 22 | 24 | 97 | 77 | 71 | 45 | 37 | 6 | 13 | 81 | 18 | 16 | 50 | 48 | 11 | 16 | 85 | 30 | 78 | 98 | 1 | 26 | 65 | 51 | 95 | 85 | 62 | 2 | 4 | 89 | 35 | 39 | 40 | 38 | 10 | 78 | 86 | 85 | 36 | 87 | 95 | 93 | 75 | 1 |
| 43 | 77 | 14 | 55 | 44 | 37 | 76 | 35 | 4 | 39 | 34 | 4 | 76 | 87 | 8 | 6 | 78 | 89 | 70 | 25 | 97 | 93 | 36 | 56 | 36 | 72 | 42 | 55 | 78 | 71 | 54 | 91 | 53 | 53 | 1 | 49 | 60 | 57 | 30 | 9 | 84 | 13 | 42 | 47 | 61 | 43 | 24 | 72 | 89 | 34 |
| 98 | 57 | 23 | 32 | 17 | 1 | 17 | 44 | 57 | 90 | 91 | 19 | 13 | 40 | 47 | 38 | 79 | 46 | 78 | 35 | 81 | 3 | 86 | 56 | 37 | 75 | 100 | 74 | 35 | 51 | 14 | 9 | 54 | 98 | 16 | 30 | 36 | 53 | 65 | 39 | 96 | 67 | 37 | 79 | 52 | 8 | 2 | 98 | 65 | 79 |
| 78 | 95 | 33 | 68 | 27 | 30 | 7 | 74 | 83 | 66 | 94 | 85 | 28 | 87 | 73 | 47 | 98 | 83 | 55 | 12 | 86 | 16 | 6 | 34 | 54 | 39 | 69 | 87 | 73 | 41 | 50 | 17 | 58 | 27 | 21 | 92 | 41 | 28 | 78 | 42 | 95 | 12 | 32 | 13 | 60 | 70 | 15 | 92 | 34 | 68 |
| 48 | 24 | 42 | 90 | 14 | 69 | 31 | 63 | 88 | 10 | 89 | 96 | 84 | 62 | 100 | 24 | 77 | 67 | 40 | 61 | 74 | 97 | 25 | 74 | 89 | 61 | 19 | 74 | 52 | 20 | 77 | 64 | 89 | 5 | 47 | 61 | 41 | 64 | 51 | 34 | 32 | 85 | 21 | 33 | 88 | 14 | 49 | 43 | 21 | 82 |
| 27 | 45 | 99 | 49 | 69 | 11 | 21 | 67 | 34 | 71 | 92 | 42 | 34 | 62 | 41 | 61 | 56 | 96 | 91 | 64 | 70 | 82 | 82 | 42 | 25 | 44 | 81 | 35 | 88 | 71 | 28 | 24 | 89 | 44 | 48 | 92 | 23 | 27 | 23 | 38 | 15 | 38 | 100 | 2 | 56 | 99 | 85 | 20 | 56 | 6 |
| 4 | 18 | 73 | 69 | 32 | 42 | 86 | 90 | 63 | 64 | 78 | 79 | 35 | 79 | 99 | 85 | 94 | 10 | 81 | 21 | 41 | 18 | 54 | 53 | 35 | 47 | 2 | 75 | 55 | 8 | 71 | 75 | 39 | 83 | 37 | 87 | 89 | 57 | 42 | 47 | 91 | 37 | 98 | 11 | 78 | 33 | 81 | 24 | 41 | 96 |
| 8 | 84 | 53 | 44 | 22 | 62 | 22 | 47 | 92 | 57 | 26 | 38 | 64 | 50 | 58 | 13 | 71 | 96 | 7 | 93 | 93 | 95 | 98 | 80 | 58 | 36 | 78 | 25 | 72 | 96 | 33 | 87 | 76 | 37 | 99 | 59 | 93 | 55 | 23 | 97 | 1 | 58 | 20 | 5 | 82 | 20 | 99 | 56 | 40 | 87 |
| 57 | 47 | 97 | 5 | 99 | 69 | 48 | 5 | 50 | 52 | 59 | 82 | 92 | 20 | 28 | 7 | 71 | 23 | 22 | 11 | 76 | 38 | 19 | 55 | 84 | 31 | 6 | 38 | 71 | 46 | 12 | 96 | 39 | 62 | 34 | 50 | 10 | 50 | 54 | 50 | 81 | 55 | 19 | 20 | 87 | 4 | 26 | 76 | 100 | 40 |

La Suma es: 25955

Figura 5: Ejecución - Ejercicio 5

6.6. Ejercicio 6 - Programa 1

6.6.1. HTML

- Etiqueta head donde se encontrarán las etiquetas meta, el título y los estilos

Listing 23: HTML - parte 1

```

3 <head>
4   <title>Ejercicio 6 </title>
5   <meta charset="UTF-8">
6   <link rel="stylesheet" href="css/estilos.css"/>
7 </head>

```

- Crear botones con atributos onclick que cambiaran el tamaño y color del texto.

Listing 24: HTML - parte 2

```

9      <div>
10         <center>
11             <h1 id="text">HOLA MUNDO!!!</h1>
12             <button id="normal" class="boton" onclick="tamaoNormal()">Text-N</button>
13             <button id="agrandar" class="boton" onclick="tamaoAgrandado()">Text-A</button>
14             <button id="black" class="boton" onclick="colorNormal()">Color-N</button>
15             <button id="red" class="boton" onclick="colorRojo()">Color-R</button>
16         </center>
17     </div>

```

- El script JavaScript cambia dinámicamente el tamaño y el color del texto en un elemento específico de una página web, usando funciones de por medio que manipulan los elementos del HTML, en este caso el texto.

Listing 25: Script 6 - programa 1

```

18     <script>
19         function tamaoNormal(){
20             document.getElementById("text").style.fontSize = '32px';
21         }
22         function tamaoAgrandado(){
23             document.getElementById("text").style.fontSize = '45px';
24         }
25         function colorNormal(){
26             document.getElementById("text").style.color = 'black';
27         }
28         function colorRojo(){
29             document.getElementById("text").style.color = 'red';
30         }
31     </script>

```

6.6.2. Ejecución

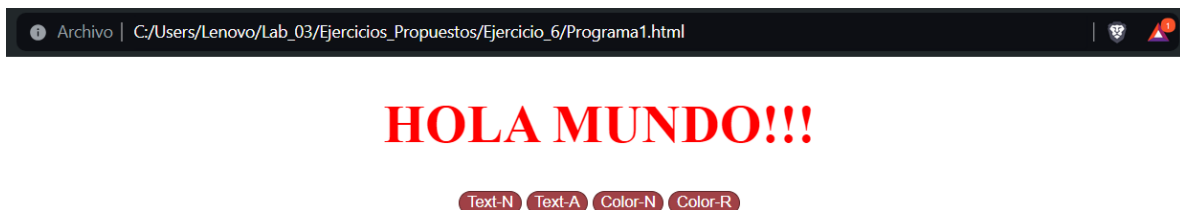


Figura 6: Ejecución - Ejercicio 6 P1

6.7. Ejercicio 6 - Programa 2

6.7.1. HTML

- Etiqueta head donde se encontrarán las etiquetas meta, el título y los estilos

Listing 26: HTML - parte 1

```
3 <head>
4   <title>Ejercicio 6 </title>
5   <meta charset="UTF-8">
6   <link rel="stylesheet" href="css/estilos.css"/>
7 </head>
```

- Implementar un formulario que pedirá al usuario dos numeros y un operador que se calculará mas adelante.

Listing 27: HTML - parte 2

```
10 <form id="formulario">
11   <label for="number1">Nmero 1:</label>
12   <input class="number" type="number" id="number1"/>
13   <br>
14   <label for="operador">Operador:</label>
15   <select class="operator" id="operador">
16     <option value="+>Suma (+)</option>
17     <option value="->Resta (-)</option>
18     <option value="*>Multiplicacin (</option>
19     <option value="/">Divisin (/)</option>
20     <option value="%>Modulo (%)</option>
21     <option value="**">Potencia (**)</option>
22     <option value="&&">And lgico (&&</option>
23     <option value="||">Or lgico (||)</option>
24     <option value="^">Xor lgico (^)</option>
25     <option value="<">Menor lgico (<</option>
26     <option value=" ">Mayor lgico (></option>
27     <option value="<=">Menor igual lgico (<=</option>
28     <option value=" ">Mayor igual lgico (>=</option>
29     <option value="&">And bit (&</option>
30     <option value="|">Or bit (|)</option>
31     <option value="^">Xor bit (^)</option>
32     <option value="<<">Desplazamiento a la izquierda (<<</option>
33     <option value=" ">Desplazamiento a la derecha (>></option>
34   </select>
35   <br>
36   <label for="number2">Nmero 2:</label>
37   <input class="number" type="number" id="number2"/>
38   <input class="boton" type="submit" value="Calcular"/>
39 </form>
```

- En esta parte, es donde mediante con el script se logrará colocar la respuesta correcta, en este caso el cálculo de los números , con el operador correspondiente.

Listing 28: HTML - parte 3

```
41 <div id="respuesta">
42 </div>
```

- El script JavaScript captura el envío de un formulario, evita su acción predeterminada y calcula el resultado de una operación matemática basada en los números y el operador seleccionados por el usuario. El resultado se muestra en la página.

Listing 29: Script 6 - Programa 2

```
43 <script>
44   document.getElementById("formulario").addEventListener('submit', function (evento){
45     evento.preventDefault();
46     const numUno = parseInt(document.getElementById("number1").value);
47     const numDos = parseInt(document.getElementById("number2").value);
48     const operador = document.getElementById("operador").value;
49     const calcular = calculator(numUno,numDos,operador);
50     const resp = document.getElementById("respuesta");
51     resp.innerHTML = "<p id='parrafo2' style='color: green;'>La respuesta es: " +
52       calcular + "</p>";
53   });
54   function calculator(a,b,c){
55     switch (c){
56       case "+":
57         return a + b;
58       case "-":
59         return a - b;
60       case "*":
61         return a * b;
62       case "/":
63         return a / b;
64       case "%":
65         return a % b;
66       case "**":
67         return a ** b;
68       case "&&":
69         return a && b;
70       case "||":
71         return a || b;
72       case "^":
73         return a ^ b;
74       case "<":
75         return a < b;
76       case ">":
77         return a > b;
78       case "<=":
79         return a <= b;
80       case ">=":
81         return a >= b;
82       case "&":
83         return a & b;
84       case "|":
85         return a | b;
86       case "<<":
87         return a << b;
88       case ">>":
89         return a >> b;
90       default:
91         return "ERROR";
92     }
93   }
```

93

</script>

6.7.2. Ejecución

Archivo | C:/Users/Lenovo/Lab_03/Ejercicios_Propuestos/Ejercicio_6/Programa2.html

Número 1:

Operador:

Número 2:

La respuesta es: 10

Figura 7: Ejecución - Suma

Archivo | C:/Users/Lenovo/Lab_03/Ejercicios_Propuestos/Ejercicio_6/Programa2.html

Número 1:

Operador:

Número 2:

La respuesta es: 2

Figura 8: Ejecución - Resta

Archivo | C:/Users/Lenovo/Lab_03/Ejercicios_Propuestos/Ejercicio_6/Programa2.html

Número 1:

Operador:

Número 2:

La respuesta es: 35

Figura 9: Ejecución - Multiplicación

Archivo | C:/Users/Lenovo/Lab_03/Ejercicios_Propuestos/Ejercicio_6/Programa2.html

Número 1:

Operador:

Número 2:

La respuesta es: 3

Figura 10: Ejecución - División

Archivo | C:/Users/Lenovo/Lab_03/Ejercicios_Propuestos/Ejercicio_6/Programa2.html

Número 1:

Operador:

Número 2:

La respuesta es: 5

Figura 11: Ejecución - Modulo

Archivo | C:/Users/Lenovo/Lab_03/Ejercicios_Propuestos/Ejercicio_6/Programa2.html

Número 1:

Operador:

Número 2:

La respuesta es: true

Figura 12: Ejecución - Mayor

Archivo | C:/Users/Lenovo/Lab_03/Ejercicios_Propuestos/Ejercicio_6/Programa2.html

Número 1:

Operador:

Número 2:

La respuesta es: false

Figura 13: Ejecución - Menor

CSS de los dos programas:

Listing 30: Estilos

```

1 #contenedor{
2     width: 310px;
3     margin: auto;
4 }
5
6 #cuerpo2{
7     color: red;
8     font-size: 25px;
9 }
10
11 .boton{
12     border-radius: 10px;
13     background: #a04146;
14     border: 1px solid #60272a;
15     color: white;

```

```

16 }
17
18 .boton:hover{
19     background: #b3666a;
20     border: 1px solid #a04146;
21 }
22
23 .operator{
24     border-radius: 10px;
25     width: 200px;
26     border: 1px solid black;
27 }
28
29 .number{
30     border-radius: 10px;
31     padding-left: 15px;
32     width: 75px;
33     border: 1px solid black;
34 }
35
36 #parrafo2{
37     text-align: center;
38 }

```

6.8. Ejercicios w3schools

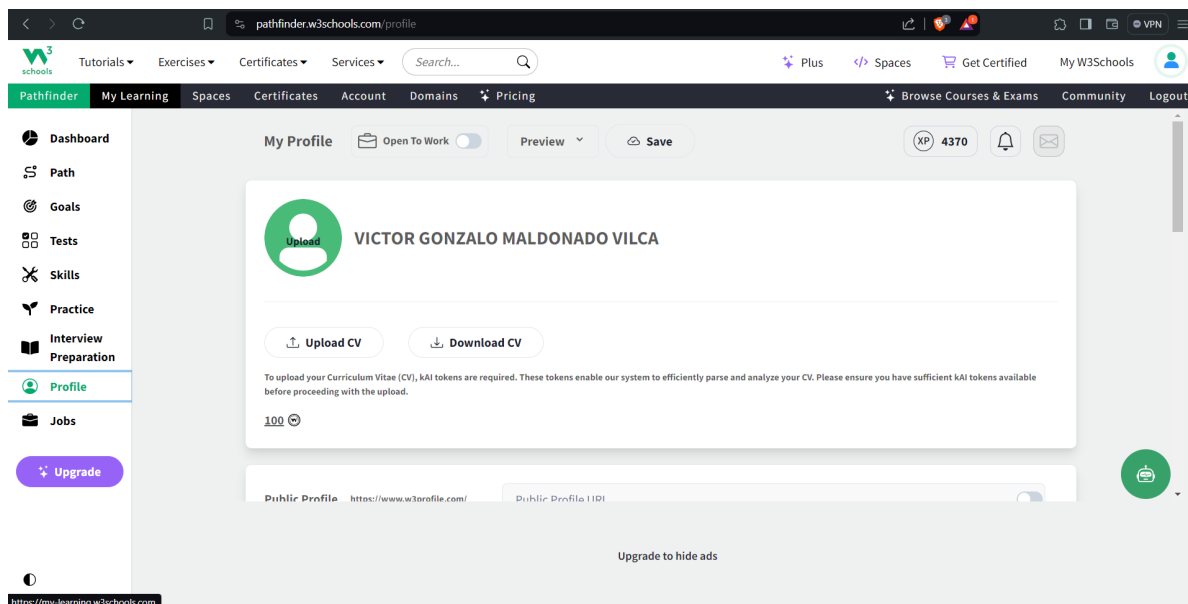
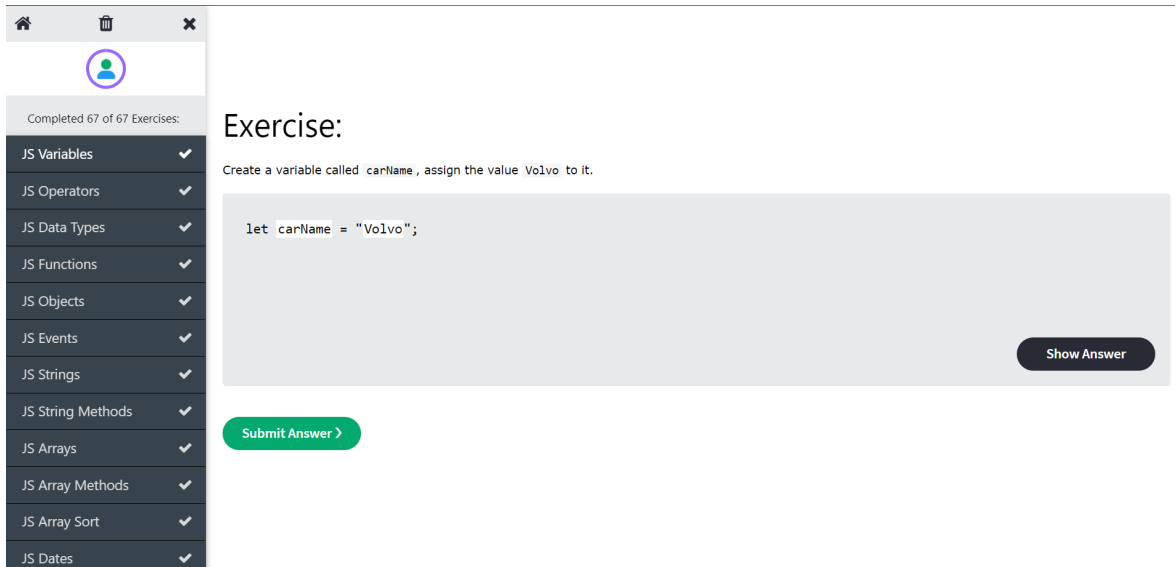


Figura 14: Usuario - w3schools



The screenshot shows the w3schools JavaScript exercises page. On the left is a sidebar with a list of topics, each with a checkmark indicating completion. The main area is titled 'Exercise:' and contains a text input field with the code `let carName = "Volvo";`. Below the input field are two buttons: 'Submit Answer >' and 'Show Answer'.

Completed 67 of 67 Exercises:

- JS Variables ✓
- JS Operators ✓
- JS Data Types ✓
- JS Functions ✓
- JS Objects ✓
- JS Events ✓
- JS Strings ✓
- JS String Methods ✓
- JS Arrays ✓
- JS Array Methods ✓
- JS Array Sort ✓
- JS Dates ✓

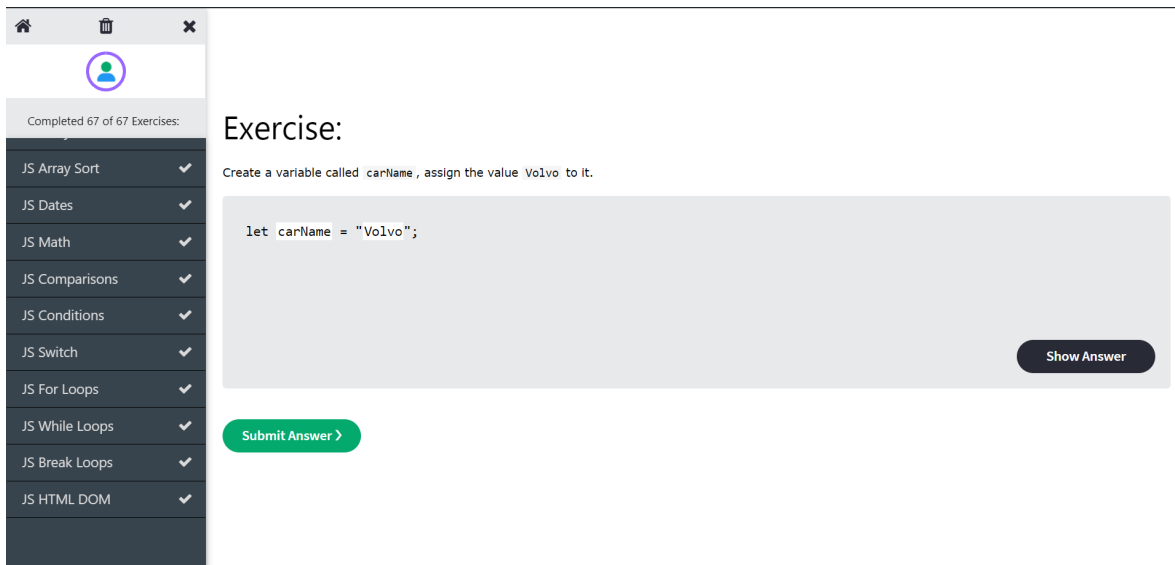
Exercise:

Create a variable called `carName`, assign the value `Volvo` to it.

```
let carName = "Volvo";
```

Submit Answer > Show Answer

Figura 15: Ejercicios - w3schools



This screenshot is similar to the previous one, showing the w3schools JavaScript exercises page. The sidebar list of topics is different, showing topics from 'JS Array Sort' to 'JS HTML DOM'. The main exercise area is identical, showing the same code input field and buttons.

Completed 67 of 67 Exercises:

- JS Array Sort ✓
- JS Dates ✓
- JS Math ✓
- JS Comparisons ✓
- JS Conditions ✓
- JS Switch ✓
- JS For Loops ✓
- JS While Loops ✓
- JS Break Loops ✓
- JS HTML DOM ✓

Exercise:

Create a variable called `carName`, assign the value `Volvo` to it.

```
let carName = "Volvo";
```

Submit Answer > Show Answer

Figura 16: Ejercicios - w3schools

6.9. Uso de GitHub : Creación de repositorio

6.9.1. Cuenta de GitHub

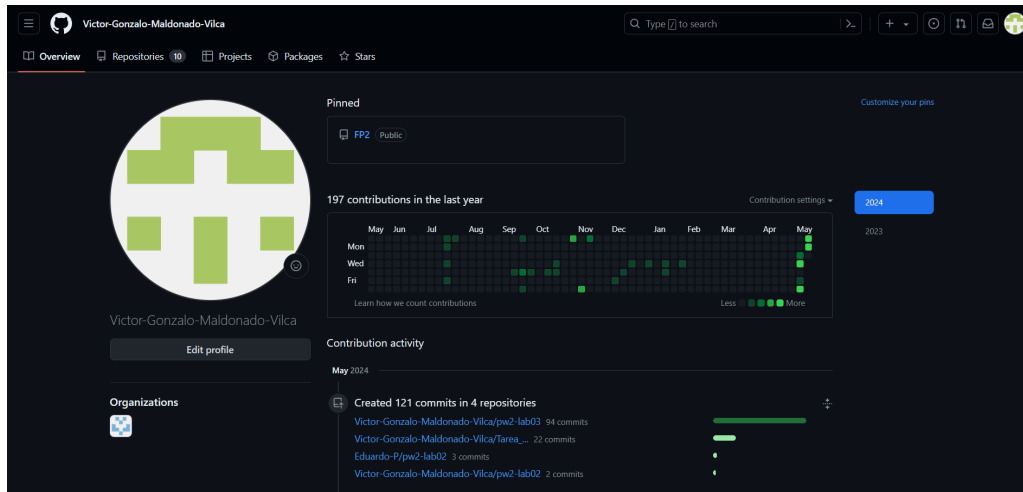


Figura 17: Usuario

6.10. Creación de un Nuevo Repositorio

6.10.1. Comandos

```
git init

git remote add origin https://github.com/Victor-Gonzalo-Maldonado-Vilca/pw2-lab03.git

git remote -v

echo "# Mi proyecto" > README.md

git add .

git commit -m "Agregar README.md"

git push -u origin master
```

Figura 18: configuración inicial

6.10.2. Implementación de Readme.md



Figura 19: README.md

6.10.3. Registro de cambios en mi código

■ Comandos

```
> git add .
> git commit -m "Agregando estilos a los problemas 1 y 2 pertenecientes al ejercicio 6 de los propuestos"
[master 15f820d] Agregando estilos a los problemas 1 y 2 pertenecientes al ejercicio 6 de los propuestos
3 files changed, 49 insertions(+), 7 deletions(-)
create mode 100644 Ejercicios_Propuestos/Ejercicio_6/css/estilos.css
> git push origin master
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 1.10 KiB | 1.10 MiB/s, done.
Total 8 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/Victor-Gonzalo-Maldonado-Vilca/pw2-lab03.git
387f0bc..15f820d master -> master
```

Figura 20: Comandos

■ Commits

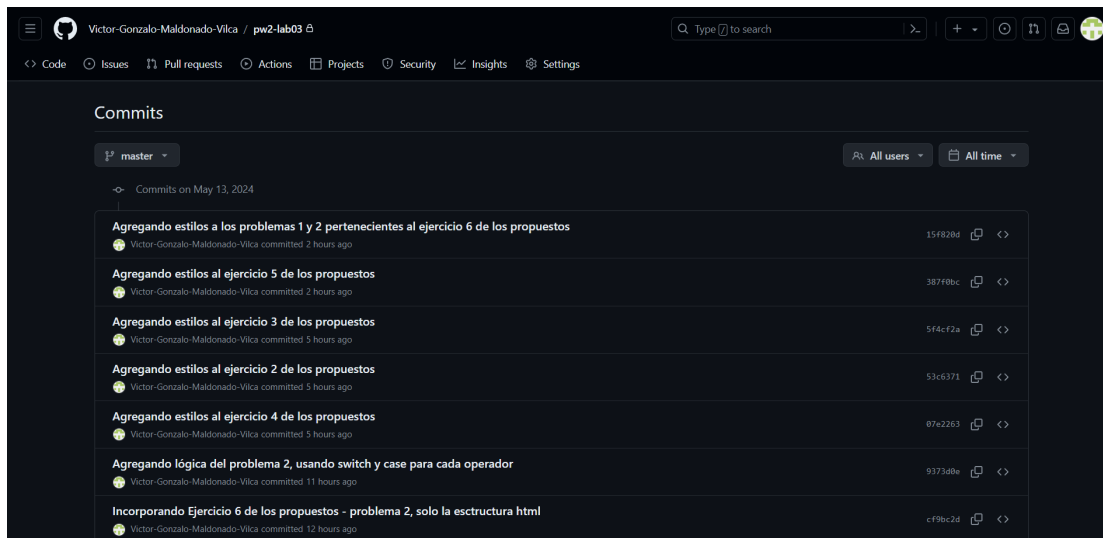


Figura 21: Commits

6.10.4. Repositorio

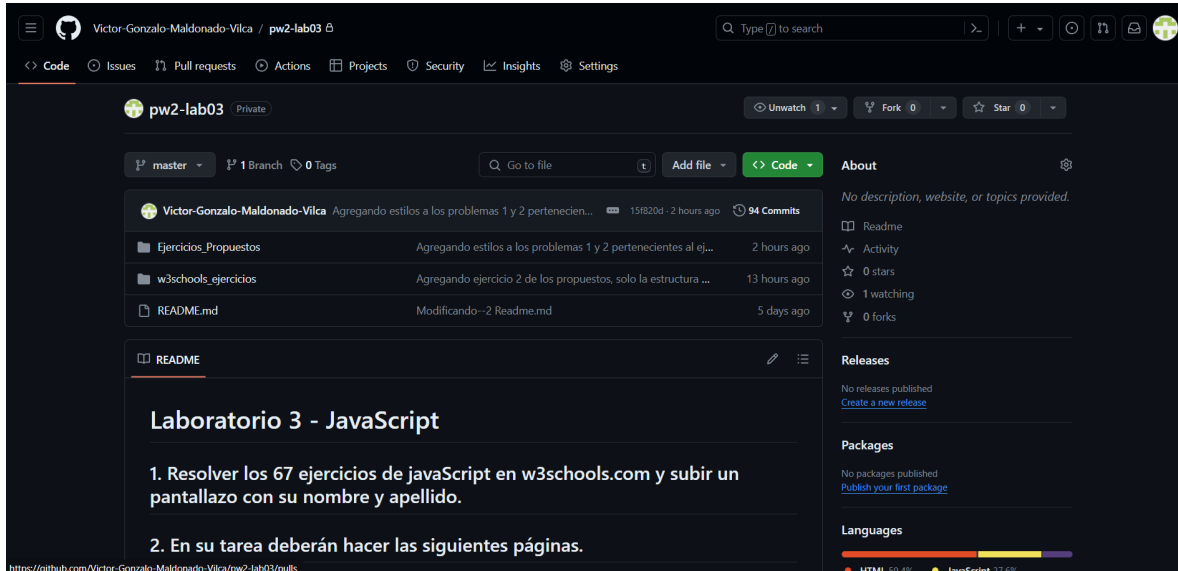


Figura 22: Repositorio

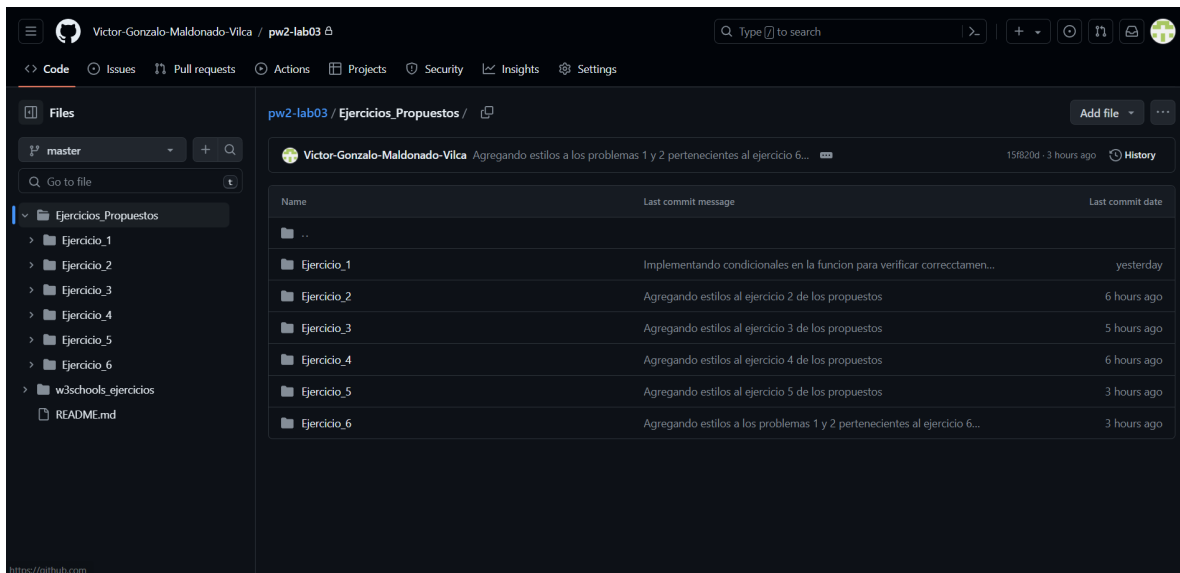


Figura 23: Repositorio - visualización

6.10.5. Proyecto compartido con el profesor de github

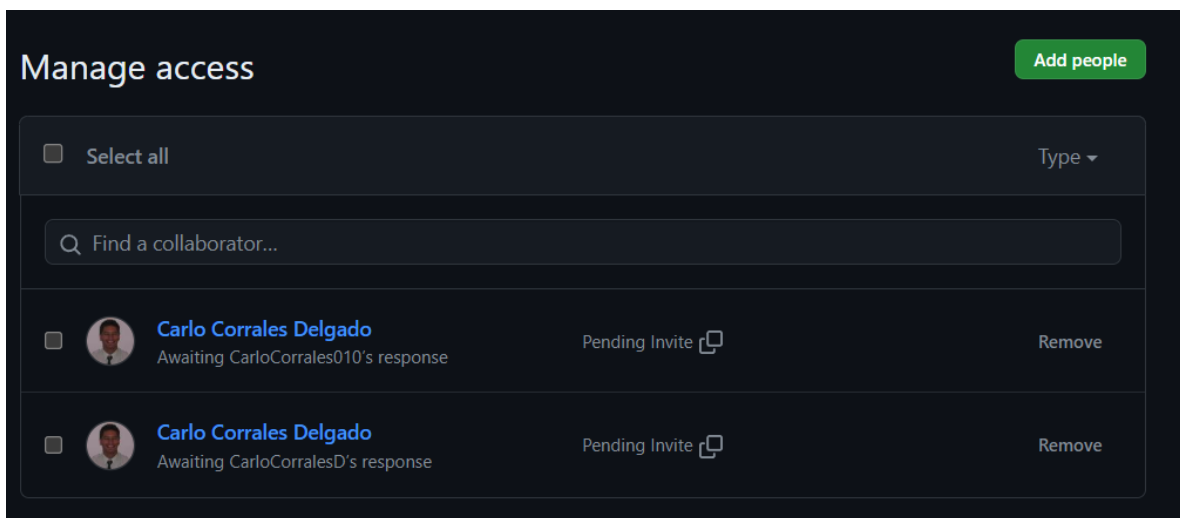


Figura 24: Compartir con el Profesor

6.11. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 1: Niveles de desempeño

| | Nivel | | | |
|---------------|----------------------|-----------------|--------------------|---------------------|
| Puntos | Insatisfactorio 25 % | En Proceso 50 % | Satisfactorio 75 % | Sobresaliente 100 % |
| 2.0 | 0.5 | 1.0 | 1.5 | 2.0 |
| 4.0 | 1.0 | 2.0 | 3.0 | 4.0 |

Tabla 2: Rúbrica para contenido del Informe y demostración

| | Contenido y demostración | Puntos | Checklist | Estudiante | Profesor |
|-------------------------|--|--------|-----------|------------|----------|
| 1. GitHub | Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar. | 2 | X | 2 | |
| 2. Commits | Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación). | 4 | X | 4 | |
| 3. Código fuente | Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones. | 2 | X | 2 | |
| 4. Ejecución | Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente. | 2 | X | 2 | |
| 5. Pregunta | Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación). | 2 | X | 2 | |
| 6. Fechas | Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos. | 2 | X | 2 | |
| 7. Ortografía | El documento no muestra errores ortográficos. | 2 | X | 2 | |
| 8. Madurez | El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación). | 4 | X | 4 | |
| Total | | 20 | | 20 | |

7. Referencias

- <https://www.w3schools.com/>
- https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_variables1
- <https://git-scm.com/doc>