

Informe de Laboratorio 01

Tema: JavaScript

Nota

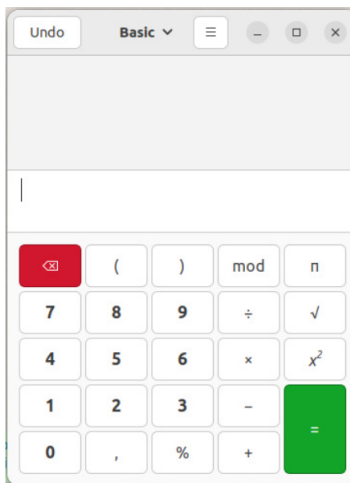
Estudiante	Escuela	Asignatura
Paulo Andre hidalgo Chinchay phidalgo@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación web 2 Semestre: III Código: 20223011

Laboratorio	Tema	Duración
01	JavaScript	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - A	Del 2 Junio 2023	Al 5 Junio 2023

1. Tarea

- Utilizando JavaScript, genere un teclado aleatorio para Banca por Internet. Funcionalidad: Los números presionados generan una clave.
- Utilizando JavaScript, genere una calculadora como la siguiente imagen.



Con memoria y botones para recorrer la pila. Agregue 3 botones: M1: memoria M2: memoria ANS

2. Equipos, materiales y temas utilizados

- Sistema Operativo Ubuntu GNU Linux 23 lunar 64 bits Kernell 6.2.
- Sistema Operativo Windows 11 pro versión 22H2 de 64 bits.
- VIM 9.0.
- Git 2.39.2.
- Visual Studio Code 1.78.2.
- Cuenta en GitHub con el correo institucional.
- Stackoverflow para desordenar un arreglo
- <https://stackoverflow.com/questions/2450954/how-to-randomize-shuffle-a-javascript-array>
-

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- <https://github.com/PauloUNSA/pw2-lab-c-23a.git>
- URL para el laboratorio 01 en el Repositorio GitHub.
- <https://github.com/PauloUNSA/pw2-lab-c-23a/tree/main/lab2>

4. Ejercicio 1

4.1. Calculadora con funciones especiales

Listing 1: Index.html con divs para agregar los botones

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <html lang="es">
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Teclado random</title>
9   <script src="./script.js"></script>
10 </head>
11 <body>
12
13 <body onload="carga()">
14   <input type="text" id="resultado" readonly>
15   <br><br>
16   <button id="boton1"><div id="1"></div></button>
17   <button id="boton2"><div id="2"></div></button>
18   <button id="boton3"><div id="3"></div></button>
19   <br>
20   <button id="boton4"><div id="4"></div></button>
```

```

21 <button id="boton5"><div id="5"></div></button>
22 <button id="boton6"><div id="6"></div></button>
23 <br>
24 <button id="boton7"><div id="7"></div></button>
25 <button id="boton8"><div id="8"></div></button>
26 <button id="boton9"><div id="9"></div></button>
27 <br>
28 <button id="boton0"><div id="0"></div></button>
29 <button onclick="borrar()">Borrar</button>
30
31 </body>
32 </html>

```

- El código JS hace que los números se impriman en un orden relativamente correcto

Listing 2: Código JavaScript

```

1 var numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0];
2 function carga(){
3     for (var i = 0; i < numeros.length; i++) {
4         document.getElementById(i).innerHTML = numeros[i]
5     }
6 }

```

2	3	4
5	6	7
8	9	0
1	Borrar	

4.2. Teclado desordenado con función JS

Listing 3: Ahora los divs tienen su respectiva función

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Teclado random</title>
8     <script src="./script.js"></script>
9 </head>
10 <body onload="carga()">
11     <input type="text" id="resultado" readonly>
12     <br><br>
13     <button id="boton1" onclick="teclear(1)"><div id="1"></div></button>
14     <button id="boton2" onclick="teclear(2)"><div id="2"></div></button>
15     <button id="boton3" onclick="teclear(3)"><div id="3"></div></button>
16     <br>

```

```

17 <button id="boton4" onclick="teclear(4)"><div id="4"></div></button>
18 <button id="boton5" onclick="teclear(5)"><div id="5"></div></button>
19 <button id="boton6" onclick="teclear(6)"><div id="6"></div></button>
20 <br>
21 <button id="boton7" onclick="teclear(7)"><div id="7"></div></button>
22 <button id="boton8" onclick="teclear(8)"><div id="8"></div></button>
23 <button id="boton9" onclick="teclear(9)"><div id="9"></div></button>
24 <br>
25 <button id="boton0" onclick="teclear(0)"><div id="0"></div></button>
26 <button onclick="borrar()">Borrar</button>
27
28 </body>
29 </html>

```

- En base al código brindado por Stackoverflow se creo la función desorden que aleatoriza el teclado

Listing 4: Codigo JavaScript

```

1 var numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
2 desorden(numeros)
3 function carga(){
4     for (var i = 0; i < numeros.length; i++) {
5         document.getElementById(i).innerHTML = numeros[i]
6     }
7 }
8 function teclear(i){
9     var num =document.getElementById("boton"+i).innerText
10    document.getElementById("resultado").value += num
11 }
12 function desorden(arreglo) {
13     var indexActual = arreglo.length;
14     var aux, random;
15     while (0 !== indexActual) {
16
17         random = Math.floor(Math.random() * indexActual)
18         indexActual--
19
20         aux = arreglo[indexActual]
21         arreglo[indexActual] = arreglo[random]
22         arreglo[random] = aux
23     }
24     return arreglo
25 }

```

2	9	4
0	1	7
8	3	6
5	Borrar	

4.3. Teclado terminado con borrado implementado

Listing 5: Se implmento la función borrar

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Teclado random</title>
8   <script src="./script.js"></script>
9   <link rel="stylesheet" href="./style.css">
10 </head>
11 <body onload="carga()">
12   <input type="text" id="resultado" readonly>
13   <br><br>
14   <button id="boton1" onclick="teclear(1)"><div id="1"></div></button>
15   <button id="boton2" onclick="teclear(2)"><div id="2"></div></button>
16   <button id="boton3" onclick="teclear(3)"><div id="3"></div></button>
17   <br>
18   <button id="boton4" onclick="teclear(4)"><div id="4"></div></button>
19   <button id="boton5" onclick="teclear(5)"><div id="5"></div></button>
20   <button id="boton6" onclick="teclear(6)"><div id="6"></div></button>
21   <br>
22   <button id="boton7" onclick="teclear(7)"><div id="7"></div></button>
23   <button id="boton8" onclick="teclear(8)"><div id="8"></div></button>
24   <button id="boton9" onclick="teclear(9)"><div id="9"></div></button>
25   <br>
26   <button id="boton0" onclick="teclear(0)"><div id="0"></div></button>
27   <button onclick="borrar()" class="borrar">Borrar</button>
28
29 </body>
30 </html>
```

- Para borrar se dejaba en blanco el cuadro de texto

Listing 6: Codigo JavaScript

```
1 var numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
2 desorden(numeros)
3 function carga(){
4   for (var i = 0; i < numeros.length; i++) {
5     document.getElementById(i).innerHTML = numeros[i]
6   }
7 }
8 function teclear(i){
9   var num =document.getElementById("boton"+i).innerText
10  document.getElementById("resultado").value += num
11 }
12 function desorden(arreglo) {
13   var indexActual = arreglo.length;
14   var aux, random;
15   while (0 !== indexActual) {
16
```

```

17     random = Math.floor(Math.random() * indexActual)
18     indexActual--
19
20     aux = arreglo[indexActual]
21     arreglo[indexActual] = arreglo[random]
22     arreglo[random] = aux
23 }
24 return arreglo
25 }
26 function borrar(){
27     document.getElementById("resultado").value = ""
28 }

```

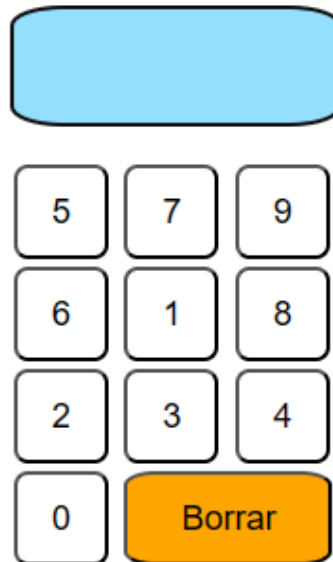
Listing 7: Codigo CSS

```

1 body {
2     font-family: "Ubuntu", sans-serif;
3 }
4 input[type="text"] {
5     width: 10%;
6     height: 50px;
7     font-size: 20px;
8     text-align: center;
9     padding: 5px;
10    border-radius: 15%;
11    background-color: rgb(147, 223, 253);
12 }
13 .borrar{
14     width: 110px;
15     background-color: orange;
16 }
17 button {
18     width: 50px;
19     height: 50px;
20     font-size: 18px;
21     margin: 2px;
22     background-color: white;
23     border-radius: 15%;
24     justify-content: center;
25 }

```

2	9	4
0	1	7
8	3	6
5	Borrar	



5. Ejercicio 1

5.1. Implementación de teclado en orden con for de JS

Listing 8: Index.html con divs para agregar los botones

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <html lang="es">
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Teclado random</title>
9   <script src="./script.js"></script>
10 </head>
11 <body>
12
13 <body onload="carga()">
14   <input type="text" id="resultado" readonly>
15   <br><br>
16   <button id="boton1"><div id="1"></div></button>
17   <button id="boton2"><div id="2"></div></button>
18   <button id="boton3"><div id="3"></div></button>
19   <br>
20   <button id="boton4"><div id="4"></div></button>
21   <button id="boton5"><div id="5"></div></button>
22   <button id="boton6"><div id="6"></div></button>
23   <br>
24   <button id="boton7"><div id="7"></div></button>
25   <button id="boton8"><div id="8"></div></button>
26   <button id="boton9"><div id="9"></div></button>
27   <br>
28   <button id="boton0"><div id="0"></div></button>

```

```
29 <button onclick="borrar()">Borrar</button>
30
31 </body>
32 </html>
```

- El código JS hace que los números se impriman en un orden relativamente correcto

Listing 9: Código JavaScript

```
1 var numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0];
2 function carga(){
3     for (var i = 0; i < numeros.length; i++) {
4         document.getElementById(i).innerHTML = numeros[i]
5     }
6 }
```

2	3	4
5	6	7
8	9	0
1	Borrar	

5.2. Teclado desordenado con función JS

Listing 10: Ahora los divs tienen su respectiva función

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Teclado random</title>
8     <script src="./script.js"></script>
9 </head>
10 <body onload="carga()">
11     <input type="text" id="resultado" readonly>
12     <br><br>
13     <button id="boton1" onclick="teclear(1)"><div id="1"></div></button>
14     <button id="boton2" onclick="teclear(2)"><div id="2"></div></button>
15     <button id="boton3" onclick="teclear(3)"><div id="3"></div></button>
16     <br>
17     <button id="boton4" onclick="teclear(4)"><div id="4"></div></button>
18     <button id="boton5" onclick="teclear(5)"><div id="5"></div></button>
19     <button id="boton6" onclick="teclear(6)"><div id="6"></div></button>
20     <br>
21     <button id="boton7" onclick="teclear(7)"><div id="7"></div></button>
22     <button id="boton8" onclick="teclear(8)"><div id="8"></div></button>
23     <button id="boton9" onclick="teclear(9)"><div id="9"></div></button>
24     <br>
```



```

25 <button id="boton0" onclick="teclear(0)"><div id="0"></div></button>
26 <button onclick="borrar()">Borrar</button>
27
28 </body>
29 </html>

```

- En base al código brindado por Stackoverflow se creo la función desorden que aleatoriza el teclado

Listing 11: Codigo JavaScript

```

1 var numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
2 desorden(numeros)
3 function carga(){
4     for (var i = 0; i < numeros.length; i++) {
5         document.getElementById(i).innerHTML = numeros[i]
6     }
7 }
8 function teclear(i){
9     var num =document.getElementById("boton"+i).innerText
10    document.getElementById("resultado").value += num
11 }
12 function desorden(arreglo) {
13     var indexActual = arreglo.length;
14     var aux, random;
15     while (0 !== indexActual) {
16
17         random = Math.floor(Math.random() * indexActual)
18         indexActual--
19
20         aux = arreglo[indexActual]
21         arreglo[indexActual] = arreglo[random]
22         arreglo[random] = aux
23     }
24     return arreglo
25 }

```

2	9	4
0	1	7
8	3	6
5	Borrar	

5.3. Teclado terminado con borrado implementado

Listing 12: Se implmento la función borrar

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>

```

```

4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Teclado random</title>
8     <script src="./script.js"></script>
9     <link rel="stylesheet" href="./style.css">
10  </head>
11  <body onload="carga()">
12      <input type="text" id="resultado" readonly>
13      <br><br>
14      <button id="boton1" onclick="teclear(1)"><div id="1"></div></button>
15      <button id="boton2" onclick="teclear(2)"><div id="2"></div></button>
16      <button id="boton3" onclick="teclear(3)"><div id="3"></div></button>
17      <br>
18      <button id="boton4" onclick="teclear(4)"><div id="4"></div></button>
19      <button id="boton5" onclick="teclear(5)"><div id="5"></div></button>
20      <button id="boton6" onclick="teclear(6)"><div id="6"></div></button>
21      <br>
22      <button id="boton7" onclick="teclear(7)"><div id="7"></div></button>
23      <button id="boton8" onclick="teclear(8)"><div id="8"></div></button>
24      <button id="boton9" onclick="teclear(9)"><div id="9"></div></button>
25      <br>
26      <button id="boton0" onclick="teclear(0)"><div id="0"></div></button>
27      <button onclick="borrar()" class="borrar">Borrar</button>
28
29  </body>
30  </html>

```

- Para borrar se dejaba en blanco el cuadro de texto

Listing 13: Código JavaScript

```

1  var numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
2  desorden(numeros)
3  function carga(){
4      for (var i = 0; i < numeros.length; i++) {
5          document.getElementById(i).innerHTML = numeros[i]
6      }
7  }
8  function teclear(i){
9      var num =document.getElementById("boton"+i).innerText
10     document.getElementById("resultado").value += num
11 }
12 function desorden(arreglo) {
13     var indexActual = arreglo.length;
14     var aux, random;
15     while (0 !== indexActual) {
16
17         random = Math.floor(Math.random() * indexActual)
18         indexActual--
19
20         aux = arreglo[indexActual]
21         arreglo[indexActual] = arreglo[random]
22         arreglo[random] = aux
23     }

```

```

24     return arreglo
25 }
26 function borrar(){
27     document.getElementById("resultado").value = ""
28 }

```

Listing 14: Código CSS

```

1 body {
2     font-family: "Ubuntu", sans-serif;
3 }
4 input[type="text"] {
5     width: 10%;
6     height: 50px;
7     font-size: 20px;
8     text-align: center;
9     padding: 5px;
10    border-radius: 15%;
11    background-color: rgb(147, 223, 253);
12 }
13 .borrar{
14     width: 110px;
15     background-color: orange;
16 }
17 button {
18     width: 50px;
19     height: 50px;
20     font-size: 18px;
21     margin: 2px;
22     background-color: white;
23     border-radius: 15%;
24     justify-content: center;
25 }

```

2	9	4
0	1	7
8	3	6
5	Borrar	

5

7

9

6

1

8

2

3

4

0

Borrar

```
C:\USERS\PAULO\PW2-LAB-C-23A\LAB1
| estilos.css
| index.html
|---fotos
|   BusterYYo.jpg
|---latex
|   lab1_paulo-hidalgo.tex
|   lab1_paulo-hidalgo.pdf
|---build
|   lab1_paulo-hidalgo.aux
|   lab1_paulo-hidalgo.fdb_latexmk
|   lab1_paulo-hidalgo.fls
|   lab1_paulo-hidalgo.log
|   lab1_paulo-hidalgo.out
|   lab1_paulo-hidalgo.pdf
|   lab1_paulo-hidalgo.synctex.gz
|---img
|   logo_abet.png
|   logo_episunsa.png
|   logo_unsa.jpg
|   Segundo-commit.png
|   ultimo-commit.png
|---src
|   css01.css
|   index01.html
|   index02.html
```

6. 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.w3.org/Style/Examples/011/firstcss.es.html>