









	Desarrollo Web	Sección A
	Randy Gabriel Choc Montes	
	0901 16 1288	Fecha de Entrega: 13 sept. 19

Informe – Ordenamientos en JavaScript

1. Creamos nuevas carpetas y nuevos archivos.

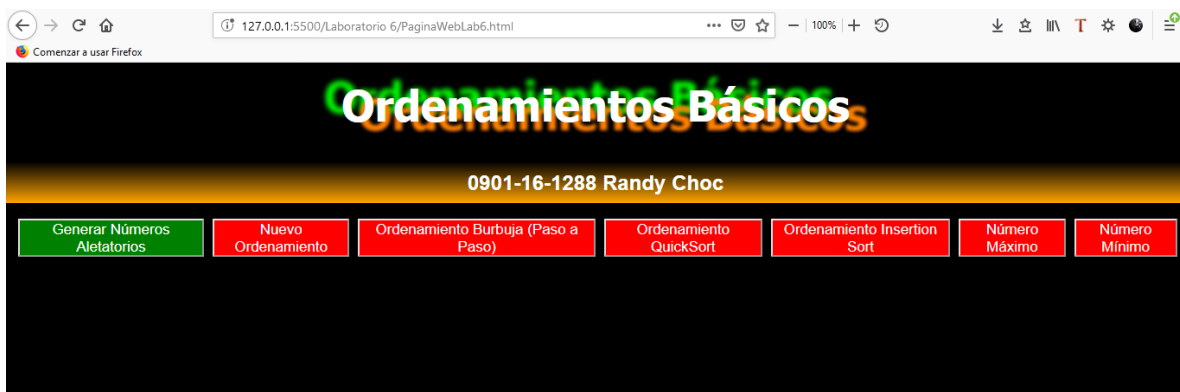
Nombre	Fecha de modifica...	Tipo	Tamaño
 css	04/09/2019 8:18 p....	Carpeta de archivos	
 documentos	11/09/2019 1:30 a. ...	Carpeta de archivos	
 js	11/09/2019 1:26 a. ...	Carpeta de archivos	
 PaginaWebLab6.html	11/09/2019 1:00 a. ...	Archivo HTML	2 KB

2. En la carpeta **js** agregamos un archivo **script.js**


Nombre	Fecha de modifica...	Tipo	Tamaño
 CompletoTresMetodos4.txt	11/09/2019 1:26 a. ...	Documento de tex...	18 KB
 script.js	11/09/2019 12:57 a...	Archivo JS	18 KB
 CompletoBur4.txt	08/09/2019 6:40 p....	Documento de tex...	9 KB
 DinamicoCon2Digitos3.txt	08/09/2019 1:05 p....	Documento de tex...	4 KB
 scriptPruebas2.txt	08/09/2019 10:07 a...	Documento de tex...	10 KB
 scriptHELP.txt	04/09/2019 8:18 p....	Documento de tex...	6 KB

Hay mas archivos con extensión txt, esto se hizo con el fin de hacer un respaldo del código creado por ordenamiento.

3. Creamos la estructura básica en HTML



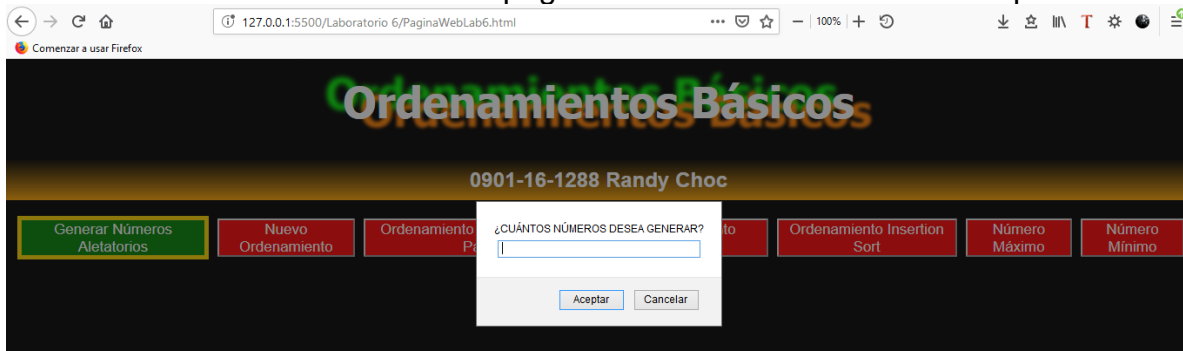
4. Se agregó diseño con CSS. El archivo está en la carpeta **css** en el archivo **estilos.css**

Nombre	Fecha de modifica...	Tipo	Tamaño
 estilos.css	11/09/2019 1:18 a. ...	Archivo CSS	2 KB

```
# estilos.css x
C: > Users > Randy > Desktop > UMG > 8.vo Semestre > Desarrollo Web > Labs-DesarrolloWeb > Laboratorio 6 > css > # estilos.css
1  *
2  {
3      font-family: sans-serif;
4      color: white;
5      margin: 0;
6      padding: 0;
7  }
8  h1
9  {
10     color: white;
11     text-align: center;
12     font-family: "Tahoma";
13     font-size: 50px;
14     text-shadow: 20px 10px 3px #ff8000,
15                 -20px -10px 5px lime;
16 }
17 h2
18 {
19     color: white;
20     text-align: center;
21     padding: 10px;
22     background: -moz-linear-gradient(black, Orange);
23 }
24 body
25 {
26     background-color: black;
27     text-align: justify;
28     color: white;
29 }
30
```

Algo importante de mencionar es que se hizo uso del mecanismo **Flexbox**.

5. Para tener interacción en la página web se hizo uso de JavaScript



6. Ejemplo de código en JavaScript

```
# estilos.css  JS script.js x
C: > Users > Randy > Desktop > UMG > 8.vo Semestre > Desarrollo Web > Labs-DesarrolloWeb > Laboratorio 6 > js > JS script.js > ...

1
2 //pedimos cuantos numeros generar
3 var cantidadAle = 0;
4 function cantAle()
5 {
6     cantidadAle = prompt("¿CUÁNTOS NÚMEROS DESEA GENERAR?", "");
7     alert("CANTIDAD A GENERAR: " + cantidadAle);
8     return cantidadAle;
9 }
10
11 //nuevo ordenamiento, todo desde cero
12 function nuevoBur()
13 {
14     document.getElementById("pilas").innerHTML= "";
15     //limpiarDiv.textContent = ("");
16     //limpiarDiv.style.clear();
17     //limpiarDiv.innerHTML = ("");
18     incP = 0;
19     listaAle = [];
20
21     lista = [];
22     listaAle = [];
23     nBur = 0; //cantidad de datos en el arreglo
24     auxBur = []; //variable temporal para lista
25     contadorDivBur = 1;
26     listaPBur = [];
27     largoauxBur = 0;
28     contaauxBur = 0;
29     aumBur = 0;
30     listaauxBur = 0;
31     listaMaximo = 0;
32     listaMinimo = 0;
```

7. Código Ordenamiento Burbuja

```
// Algoritmo de burbuja
for (k = 1; k < nBur; k++) //1er FOR 1-5
{
    for (i = 0; i < (nBur-k); i++) //2do FOR 0-(9-1) |
    {
        if (lista[i] > lista[i + 1])
        {
            auxBur = lista[i];
            lista[i] = lista[i + 1];
            lista[i + 1] = auxBur;
        }
        //console.log("Lista #" + contadorDivBur + ": " + lista + " Y aum: " + aumBur);
        contadorDivBur += 1;
        listaPBur[aumBur] = lista.join("");
        aumBur++;
    }
}
```

8. Código Ordenamiento QuickSort

```
//ordenamiento QuickSort
function quicksort(primeros,ultimo)
{
    //definimos variables indices
    i = primeros
    j = ultimo
    //sacamos el pivote de la mitad del arreglo
    pivote = arreglo[parseInt((i+j)/2)];
    //repetir hasta que i siga siendo menor que j
    do
    {
        //mientras arreglo[i] sea menor a pivote
        while(arreglo[i]<pivote) i++; //mientras j sea mayor a pivote
        while(arreglo[j]>pivote) j--;

        //si i es menor o igual a j, los valores ya se cruzaron
        if(i<=j)
        {
            //variable temporal auxiliar para guardar valor de arreglo[j]
            aux=arreglo[j];

            //intercambiamos los valores de arreglo[j] y arreglo[i]
            arreglo[j] = arreglo[i]
            arreglo[i] = aux
            // incrementamos y decrementamos i y j
            i++;
            j--;
        }
    }while(i<j);

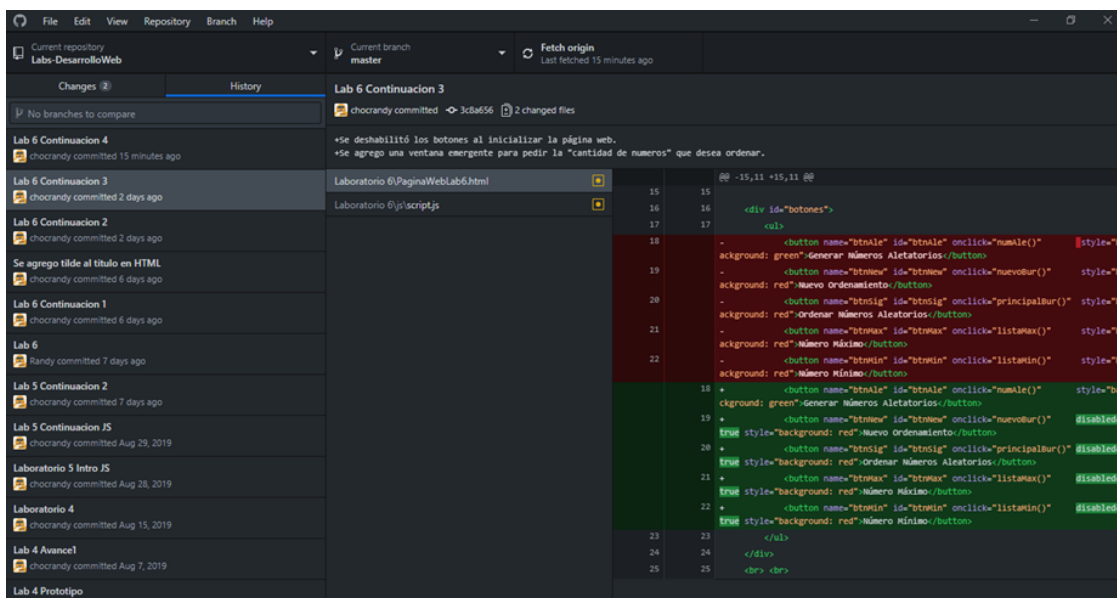
    //si primeros es menor a j llamamos la funcion nuevamente
    if(primeros<j)
    {
        quicksort(primeros,j);
    } //si ultimo es mayor que i llamamos la funcion nuevamente
    if(ultimo>i)
    {
        quicksort(i,ultimo);
    }
}
```

9. Código Ordenamiento Insert Sort

```
const insertionSort = arreglo =>
{
  const l = arreglo.length;
  let j, temp;

  for ( let i = 1; i < l; i++ )
  {
    j = i;
    temp = arreglo[ i ];
    while ( j > 0 && arreglo[ j - 1 ] > temp )
    {
      arreglo[ j ] = arreglo[ j - 1 ];
      j--;
    }
    arreglo[ j ] = temp;
  }
  return arreglo;
};
```

10. Para mantener un control de las versiones se utiliza GitHub.



11. Para revisar dicho laboratorio debe ir al siguiente link:

<https://github.com/chocrandy/Labs-DesarrolloWeb.git>

Nota:

Seleccionar folder llamado **Laboratorio 6**