	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES

CARRERA: ING. SISTEMAS

ASIGNATURA: HYPERMEDIAL

NRO. PRÁCTICA:

03

TÍTULO PRÁCTICA: RESOLUCION DE PROBLEMAS SOBRE JAVASCRIPT.

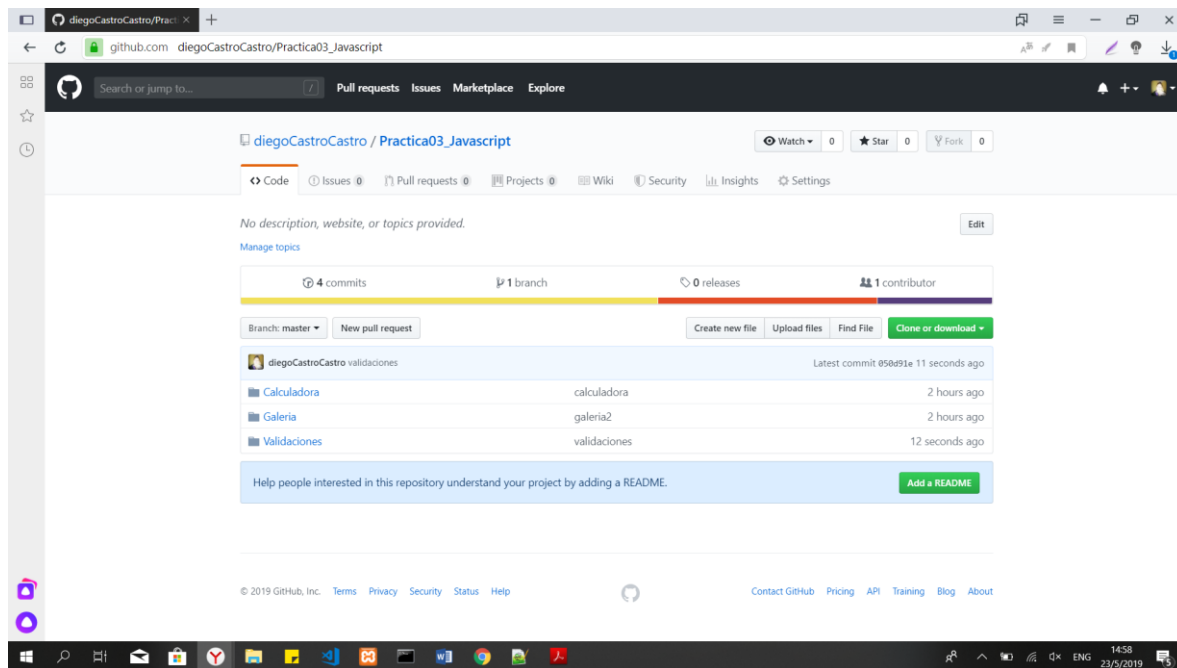
OBJETIVO ALCANZADO:


Desarrollar cada practica y comprende mejor el lenguaje que se utilizo.

ACTIVIDADES DESARROLLADAS

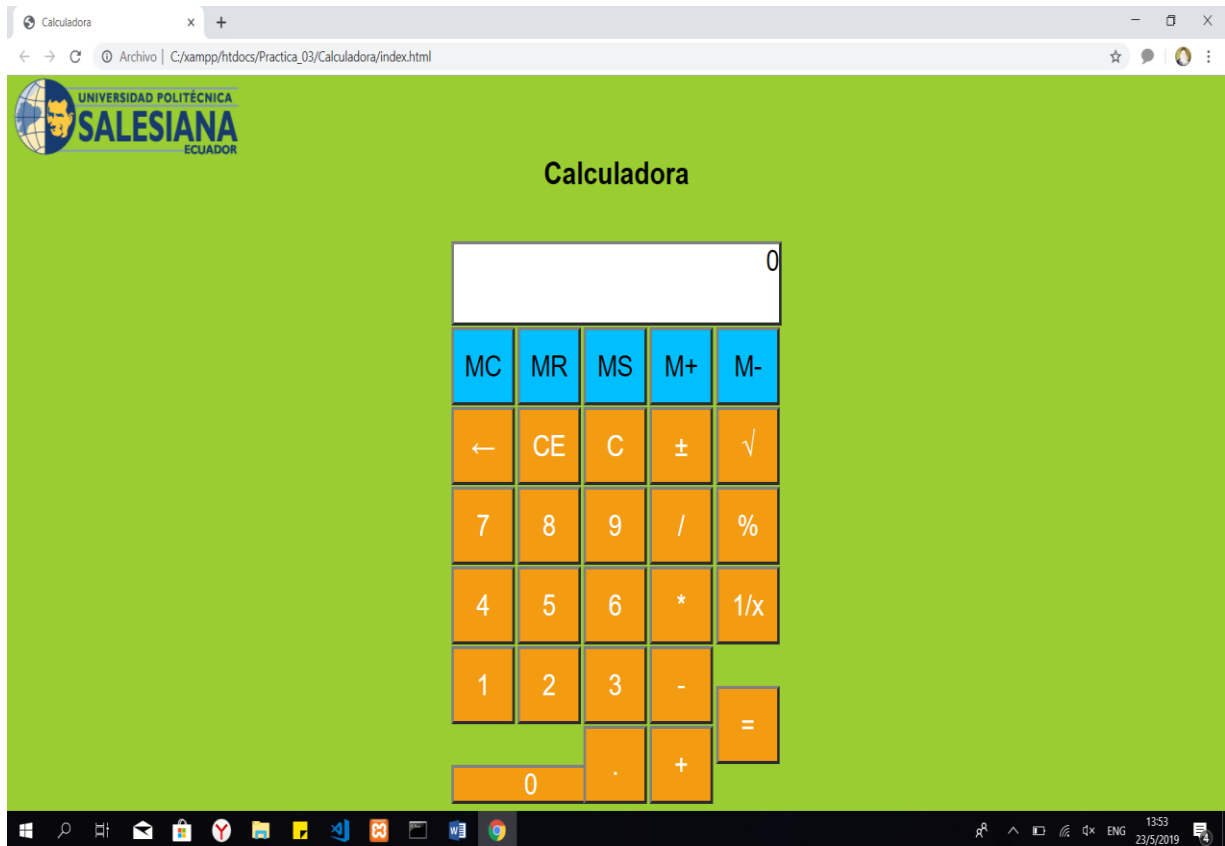
Repositorio GitHub.

https://github.com/diegoCastroCastro/Practica03_Javascript



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

1. Calculadora.



• Código html.

```
<DOCTYPE html>
<html>

<head>
  <meta charset="utf-8" />
  <title>Calculadora</title>

  <link rel="stylesheet" type="text/css" media="screen" href="css/estilo.css" />
  <script type="text/javascript" src="js/funciones.js"></script>
  
</head>
```

```
<body>
  <noscript>
    Your browser doesn't support or has disabled JavaScript
  </noscript>

  <center>
    <H1>Calculadora</H1>

    <div class="contenedor">
      <form>
        <table>
          <tr>
            <td colspan="5">
              <p type="text" id="textoPantalla">0</p>
            </td>
          <tr>
            <td><input type="button" class="columna2" value="MC" ></td>
            <td><input type="button" class="columna2" value="MR" > </td>
            <td><input type="button" class="columna2" value="MS" > </td>
            <td><input type="button" class="columna2" value="M+" > </td>
            <td><input type="button" class="columna2" value="M-" > </td>
          </tr>
          <tr>
            <td><input type="button" class="columna1" value="←"
onclick="retro()"></td>
            <td><input type="button" class="columna1" value="CE"
onclick="borradoParcial()"></td>
            <td><input type="button" class="columna1" value="C"
onclick="borradoTotal()"> </td>
            <td><input type="button" class="columna1" value="±"
onclick="signoOpuesto()"> </td>
            <td><input type="button" class="columna1" value="√"
onclick="raiz()"></td>
          </tr>
          <tr>
            <td><input type="button" class="columna1" value="7"
onclick="numero('7')"></td>
            <td><input type="button" class="columna1" value="8"
onclick="numero('8')"></td>
```

```

        <td><input type="button" class="columna1" value="9"
onclick="numero('9')"></td>
        <td><input type="button" class="columna1" value="/"
onclick="operacion('/')"></td>
        <td><input type="button" class="columna1" value="%"
onclick="porcentaje()"></td>

    </tr>
    <tr>
        <td><input type="button" class="columna1" value="4"
onclick="numero('4')"></td>
        <td><input type="button" class="columna1" value="5"
onclick="numero('5')"></td>
        <td><input type="button" class="columna1" value="6"
onclick="numero('6')"></td>
        <td><input type="button" class="columna1" value="*"
onclick="operacion('*')"></td>
        <td><input type="button" class="columna1" value="1/x"
onclick="inve('1/x')"></td>

    </tr>
    <tr>
        <td><input type="button" class="columna1" value="1"
onclick="numero('1')"></td>
        <td><input type="button" class="columna1" value="2"
onclick="numero('2')"></td>
        <td><input type="button" class="columna1" value="3"
onclick="numero('3')"></td>
        <td><input type="button" class="columna1" value="-"
onclick="operacion('-')"></td>
        <td rowspan="2"><input type="button" class="columna1"
value="=" onclick="igual()"></td>
    </tr>
    <tr>
        <td colspan="2"><input type="button" id="cero" value="0"
onclick="numero('0')"></td>
        <td><input type="button" id="punto" value="."
onclick="numero(', ')"></td>
        <td><input type="button" id="suma" id="suma" value="+"
onclick="operacion('+')"></td>

    </tr>
</table>
</form>

```

```
</div>
</center>
</body>
</html>
```

• Código js

```
window.onload = function () { //Acciones tras cargar la página
    pantalla = document.getElementById("textoPantalla"); //elemento pantalla de salida
}
x = "0"; //número en pantalla
xi = 1; //iniciar número en pantalla: 1=si; 0=no;
coma = 0; //estado coma decimal 0=no, 1=si;
ni = 0; //número oculto o en espera.
op = "no"; //operación en curso; "no" = sin operación.

//mostrar número en pantalla según se va escribiendo:
function numero(xx) { //recoge el número pulsado en el argumento.
    if (x == "0" || xi == 1) { // inicializar un número,
        pantalla.innerHTML = xx; //mostrar en pantalla
        x = xx; //guardar número
        if (xx == ".") { //si escribimos una coma al principio del número
            pantalla.innerHTML = "0."; //escribimos 0.
            x = xx; //guardar número
            coma = 1; //cambiar estado de la coma
        }
    }
    else { //continuar escribiendo un número
        if (xx == "." && coma == 0) { //si escribimos una coma decimal por primera vez
            pantalla.innerHTML += xx;
            x += xx;
            coma = 1; //cambiar el estado de la coma
        }
        //si intentamos escribir una segunda coma decimal no realiza ninguna acción.
        else if (xx == "." && coma == 1) { }
        //Resto de casos: escribir un número del 0 al 9
        else {
            pantalla.innerHTML += xx;
            x += xx
        }
    }
}
```

```
xi = 0 //el número está iniciado y podemos ampliarlo.
}

function igual() {
  if (op == "no") { //no hay ninguna operación pendiente.
    pantalla.innerHTML = x; //mostramos el mismo número
  }
  else { //con operación pendiente resolvemos
    sl = ni + op + x; // escribimos la operación en una cadena
    sol = eval(sl) //convertimos la cadena a código y resolvemos
    pantalla.innerHTML = sol //mostramos la solución
    x = sol; //guardamos la solución
    op = "no"; //ya no hay operaciones pendientes
    xi = 1; //se puede reiniciar la pantalla.
  }
}

function retro() { //Borrar sólo el último número escrito.
  cifras = x.length; //hayar número de caracteres en pantalla
  br = x.substr(cifras - 1, cifras) //describir último caracter
  x = x.substr(0, cifras - 1) //quitar el ultimo caracter
  if (x == "") { x = "0"; } //si ya no quedan caracteres, pondremos el 0
  if (br == ".") { coma = 0; } //Si el caracter quitado es la coma, se permite
  escribirla de nuevo.
  pantalla.innerHTML = x; //mostrar resultado en pantalla
}

function borradoParcial() {
  pantalla.innerHTML = 0; //Borrado de pantalla;
  x = 0; //Borrado indicador número pantalla.
  coma = 0; //reiniciamos también la coma
}

function borradoTotal() {
  pantalla.innerHTML = 0; //poner pantalla a 0
  x = "0"; //reiniciar número en pantalla
  coma = 0; //reiniciar estado coma decimal
  ni = 0 //indicador de número oculto a 0;
  op = "no" //borrar operación en curso.
}

function signoOpuesto() {
  nx = Number(x); //convertir en número
  nx = -nx; //cambiar de signo
```

```
x = String(nx); //volver a convertir a cadena
pantalla.innerHTML = x; //mostrar en pantalla.
}

function raiz() {
    x = Math.sqrt(x) //resolver raíz cuadrada.
    pantalla.innerHTML = x; //mostrar en pantalla resultado
    op = "no"; //quitar operaciones pendientes.
    xi = 1; //se puede reiniciar la pantalla
}

function porcentaje() {
    x = x / 100 //dividir por 100 el número
    pantalla.innerHTML = x; //mostrar en pantalla
    igual() //resolver y mostrar operaciones pendientes
    xi = 1 //reiniciar la pantalla
}

function operacion(s) {
    igual() //si hay operaciones pendientes se realizan primero
    ni = x //ponemos el 1º número en "numero en espera" para poder escribir el segundo.
    op = s; //guardamos tipo de operación.
    xi = 1; //inicializar pantalla.
}

function teclado(elEvento) {
    evento = elEvento || window.event;
    k = evento.keyCode; //número de código de la tecla.
    //teclas numéricas del teclado alfanumérico
    if (k > 47 && k < 58) {
        p = k - 48; //buscar número a mostrar.
        p = String(p) //convertir a cadena para poder añadir en pantalla.
        numero(p); //enviar para mostrar en pantalla
    }
    //Teclas del teclado numérico. Seguimos el mismo procedimiento que en el anterior.
    if (k > 95 && k < 106) {
        p = k - 96;
        p = String(p);
        numero(p);
    }
    if (k == 110 || k == 190) { numero(".") } //teclas de coma decimal
    if (k == 106) { operacion('*') } //tecla multiplicación
    if (k == 107) { operacion('+') } //tecla suma
```

```
if (k == 109) { operacion('-') } //tecla resta
if (k == 111) { operacion('/') } //tecla división
if (k == 32 || k == 13) { igualar() } //Tecla igual: intro o barra espaciadora
if (k == 46) { borradoTotal() } //Tecla borrado total: "supr"
if (k == 8) { retro() } //Retroceso en escritura : tecla retroceso.
if (k == 36) { borradoParcial() } //Tecla borrado parcial: tecla de inicio.
}
function inve() {
  nx = Number(x);
  nx = (1 / nx);
  x = String(nx);
  pantalla.innerHTML = x;
  xi = 1; //reiniciar pantalla al pulsar otro número.
}
```

• Código css.

```
*{
  margin: 0px;
  padding: 0px;
}
body{
  background-color: yellowgreen;
}
.contenedor{
  position: relative;
  top: 50px;
}
#textoPantalla{
  text-align: right;
  font-size: 30px;
  font-family: Arial,Helvetica;
  color: black;
  border: 3px;
  border-color: #808080;
  border-style: outset;
```



```
background: white;
width: 410px;
height: 80px;
}

.columna1{
font-size: 30px;
font-family: Arial,Helvetica;
color: white;
border: 3px;
border-color: #808080;
border-style: outset;
background: #f39c12;
width: 80px;
height: 80px;
}


.columna2{
font-size: 30px;
font-family: Arial,Helvetica;
color:black;
border: 3px;
border-color: #808080;
border-style: outset;
background: #00BFFF;
width: 80px;
height: 80px;
}

#cero{
position: absolute;
font-size: 30px;
font-family: Arial,Helvetica;
color: white;
border: 3px;
border-color: #808080;
border-style: outset;
background: #f39c12;
width: 200px;
height: 40px;
}

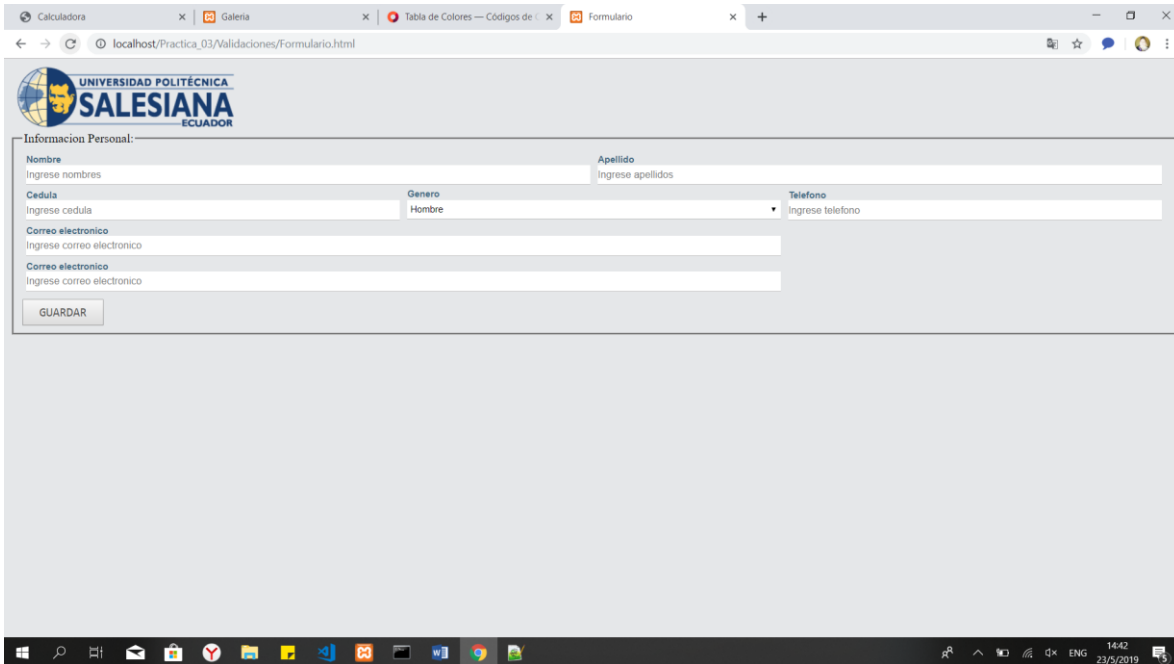
#punto{
position: relative;
font-size: 30px;
```

```
font-family: Arial,Helvetica;
color: white;
border: 3px;
border-color: #808080;
border-style: outset;
background: #f39c12;
width: 80px;
height: 80px;
}

#suma{
  position: relative;
  font-size: 30px;
  font-family: Arial,Helvetica;
  color: white;
  border: 3px;
  border-color: #808080;
  border-style: outset;
  background: #f39c12;
  width: 80px;
  height: 80px;
}
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

2. Validación formulario.



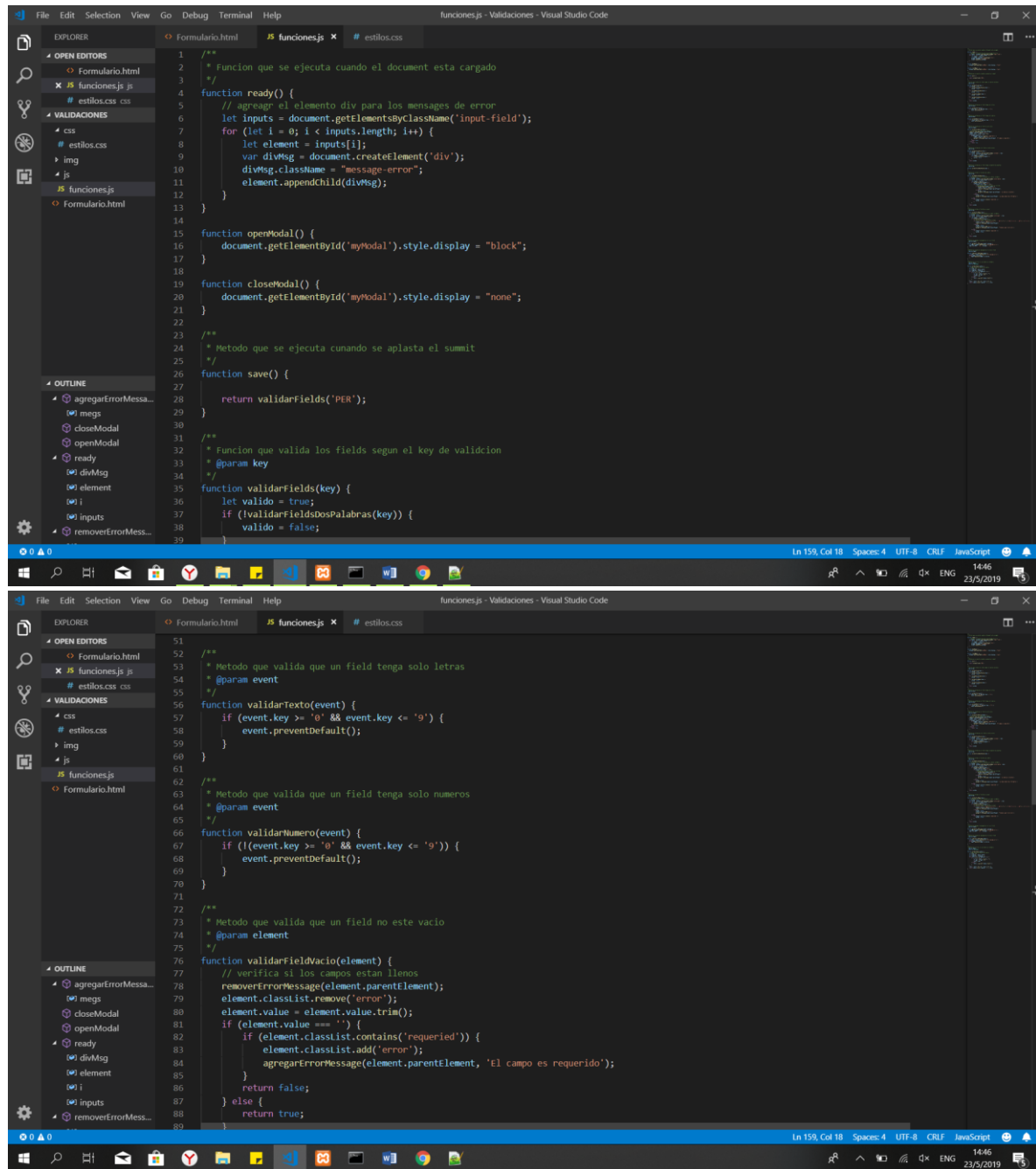
Codigo html.

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <title>Formulario</title>
7    <link rel="stylesheet" type="text/css" href="css/estilos.css">
8    <script src="js/funciones.js" rel="script" type="text/javascript"></script>
9    
10 </head>
11
12 <body onload="ready()">
13   <div id="myModal" class="modal">
14     <form class="modal-content formulario" style="width: 45%;">
15       <fieldset>
16         <legend>Información Personal:</legend>
17
18         <div class="modal-context">
19           <div class="row">
20             <div class="input-field col s5">
21               <input id="nombre" type="text" name="nombre" class="valid-two-wordsPER required"
22                 placeholder="Ingrese nombres" maxlength="30" onkeypress="validarTexto(event)">
23               <label for="nombre">Nombre </label>
24             </div>
25             <div class="input-field col s5">
26               <input id="apellido" type="text" class="valid-two-wordsPER required"
27                 placeholder="Ingrese apellidos" maxlength="30" onkeypress="validarTexto(event)">
28               <label>Apellido </label>
29             </div>
30             <div class="input-field col s3">
31               <input id="cedula" type="text" class="valid-cedulaPER required" placeholder="Ingrese cedula"
32                 maxlength="10" onkeypress="validarNumero(event)">
33               <label>Cedula </label>
34             </div>
35             <div class="input-field col s3">
36               <select>
37                 <option value="H"> Hombre </option>
38                 <option value="M"> Mujer </option>
39               </select>

```


Código js.



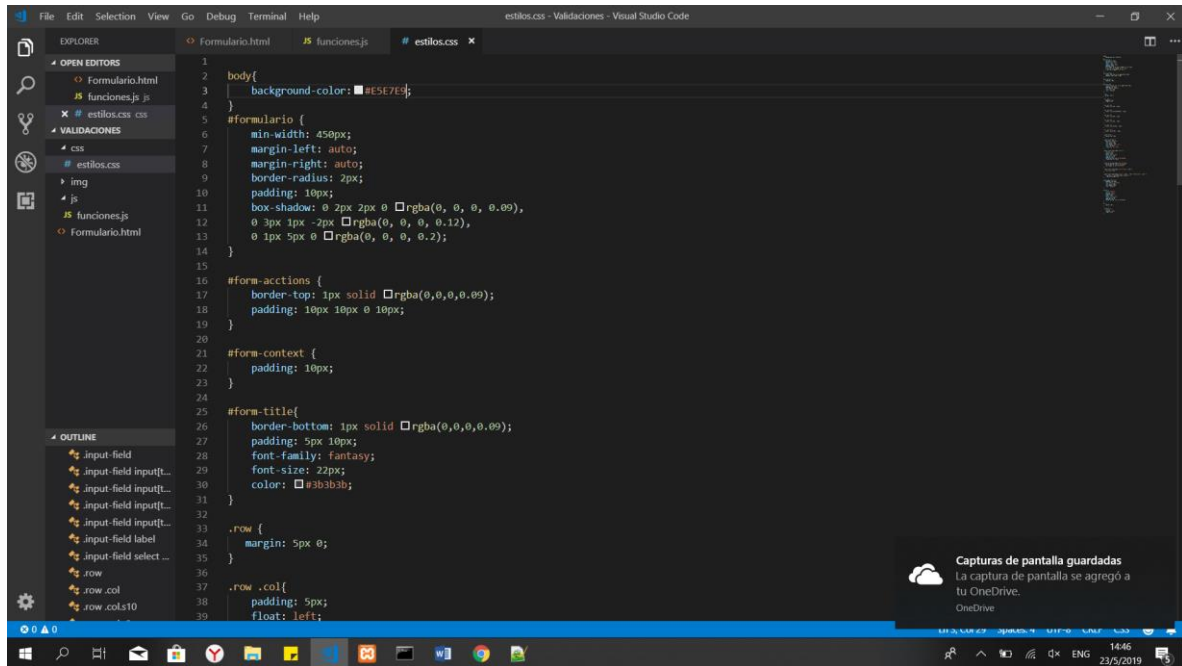
```

1  /**
2  * Funcion que se ejecuta cuando el document esta cargado
3  */
4  function ready() {
5      // agregar el elemento div para los mensajes de error
6      let inputs = document.getElementsByClassName("input-field");
7      for (let i = 0; i < inputs.length; i++) {
8          let element = inputs[i];
9          var divMsg = document.createElement('div');
10         divMsg.className = "message-error";
11         element.appendChild(divMsg);
12     }
13 }
14
15 function openModal() {
16     document.getElementById("myModal").style.display = "block";
17 }
18
19 function closeModal() {
20     document.getElementById("myModal").style.display = "none";
21 }
22
23 /**
24 * Metodo que se ejecuta cuando se aplasta el submit
25 */
26 function save() {
27     return validarFields("PER");
28 }
29
30 /**
31 * Funcion que valida los fields segun el key de validcion
32 * @param key
33 */
34 function validarFields(key) {
35     let valido = true;
36     if (!validarFieldsDosPalabras(key)) {
37         valido = false;
38     }
39 }
40
41 /**
42 * Metodo que valida que un field tenga solo letras
43 * @param event
44 */
45 function validarTexto(event) {
46     if (event.key >= '0' && event.key <= '9') {
47         event.preventDefault();
48     }
49 }
50
51 /**
52 * Metodo que valida que un field tenga solo numeros
53 * @param event
54 */
55 function validarNumero(event) {
56     if (!event.key >= '0' && event.key <= '9') {
57         event.preventDefault();
58     }
59 }
60
61 /**
62 * Metodo que valida que un field no este vacio
63 * @param element
64 */
65 function validarFieldVacio(element) {
66     // verifica si los campos estan llenos
67     removerErrorMessage(element.parentElement);
68     element.classList.remove("error");
69     element.value = element.value.trim();
70     if (element.value === '') {
71         if (element.classList.contains("requerido")) {
72             element.classList.add("error");
73             agregarErrorMessage(element.parentElement, 'El campo es requerido');
74         }
75         return false;
76     } else {
77         return true;
78     }
79 }
80

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		


Código css.



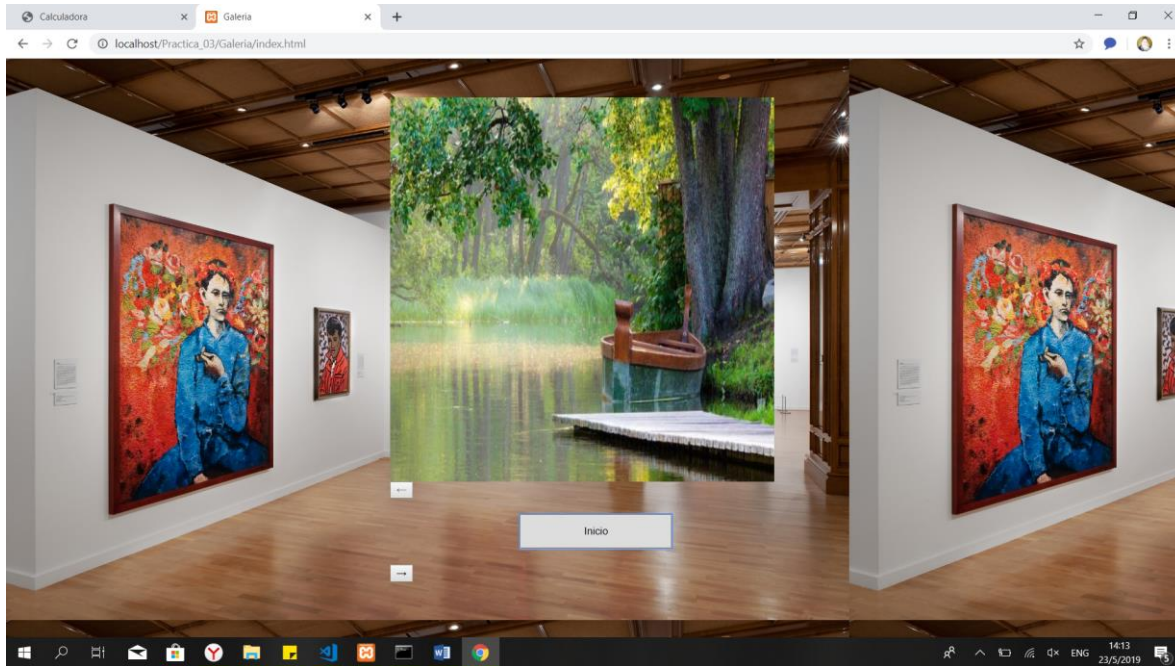
```

1  body{
2  |   background-color: #E9E7E9;
3  | }
4  |
5  | #formulario {
6  |   min-width: 450px;
7  |   margin-left: auto;
8  |   margin-right: auto;
9  |   border-radius: 2px;
10 |   padding: 10px;
11 |   box-shadow: 0 2px 2px 0 rgba(0, 0, 0, 0.09),
12 |               0 3px 1px -2px rgba(0, 0, 0, 0.12),
13 |               0 1px 5px 0 rgba(0, 0, 0, 0.2);
14 | }
15 |
16 | #form-actions {
17 |   border-top: 1px solid rgba(0,0,0,0.09);
18 |   padding: 10px 10px 0 10px;
19 | }
20 |
21 | #form-context {
22 |   padding: 10px;
23 | }
24 |
25 | #form-title{
26 |   border-bottom: 1px solid rgba(0,0,0,0.09);
27 |   padding: 5px 10px;
28 |   font-family: fantasy;
29 |   font-size: 22px;
30 |   color: #3b3b3b;
31 | }
32 |
33 | .row {
34 |   margin: 5px 0;
35 | }
36 |
37 | .row .col{
38 |   padding: 5px;
39 |   float: left;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

3. Galería de imágenes.



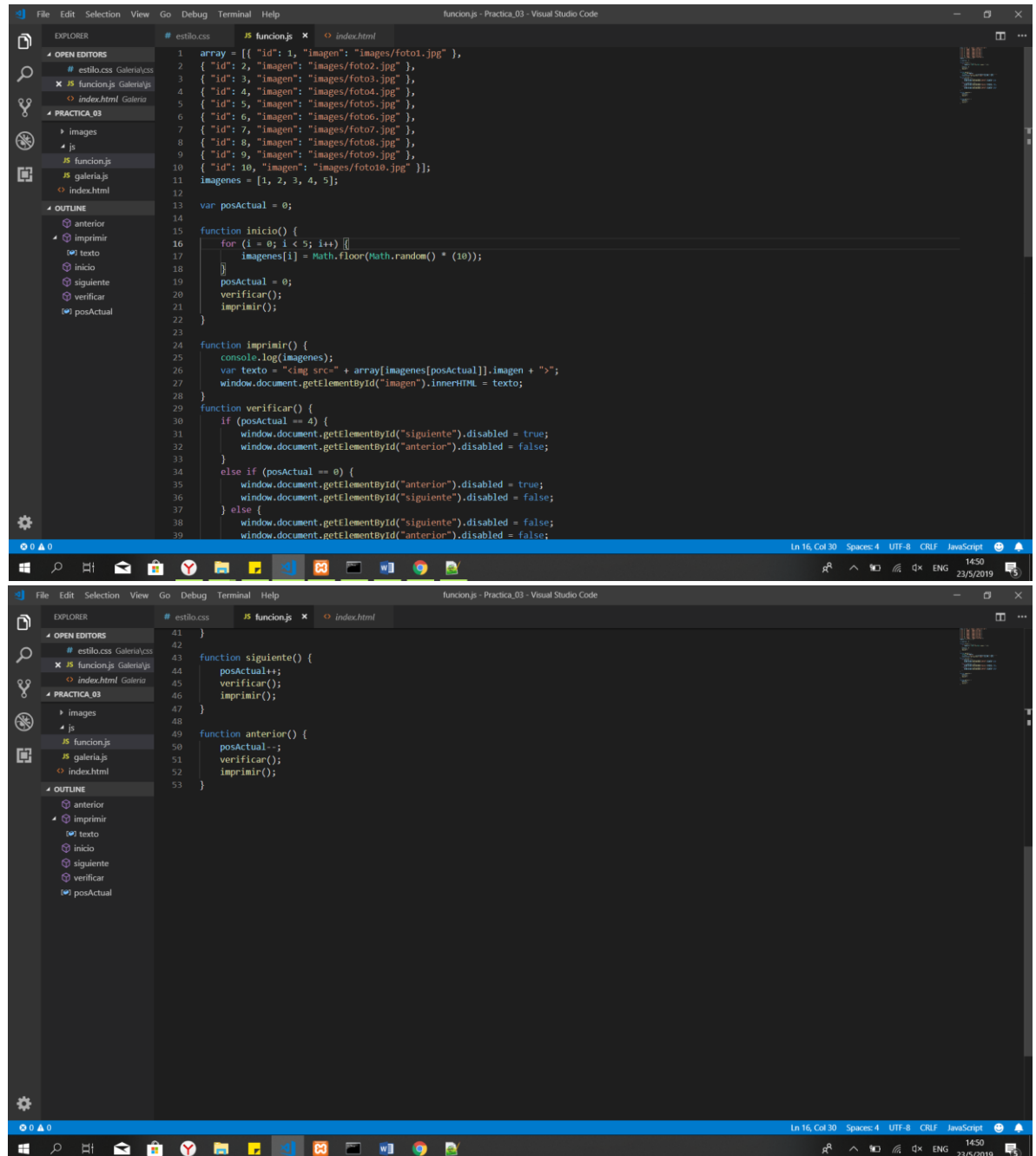
• Código html.

```

1  <DOCTYPE html>
2  <html>
3
4  <head>
5    <link rel="stylesheet" type="text/css" media="screen" href="css/estilo.css" />
6    <script src="js/function.js"></script>
7
8    <meta charset="utf-8" />
9    <title>Galeria</title>
10 </head>
11
12 <body>
13   <div id="imagen"></div>
14
15   <button id="anterior" onclick="anterior();"></button>
16   <button id="inicio" onclick="inicio();">Inicio</button>
17   <button id="siguiente" onclick="siguiente();"></button>
18
19 </body>
20
21 </html>

```


• Código js.



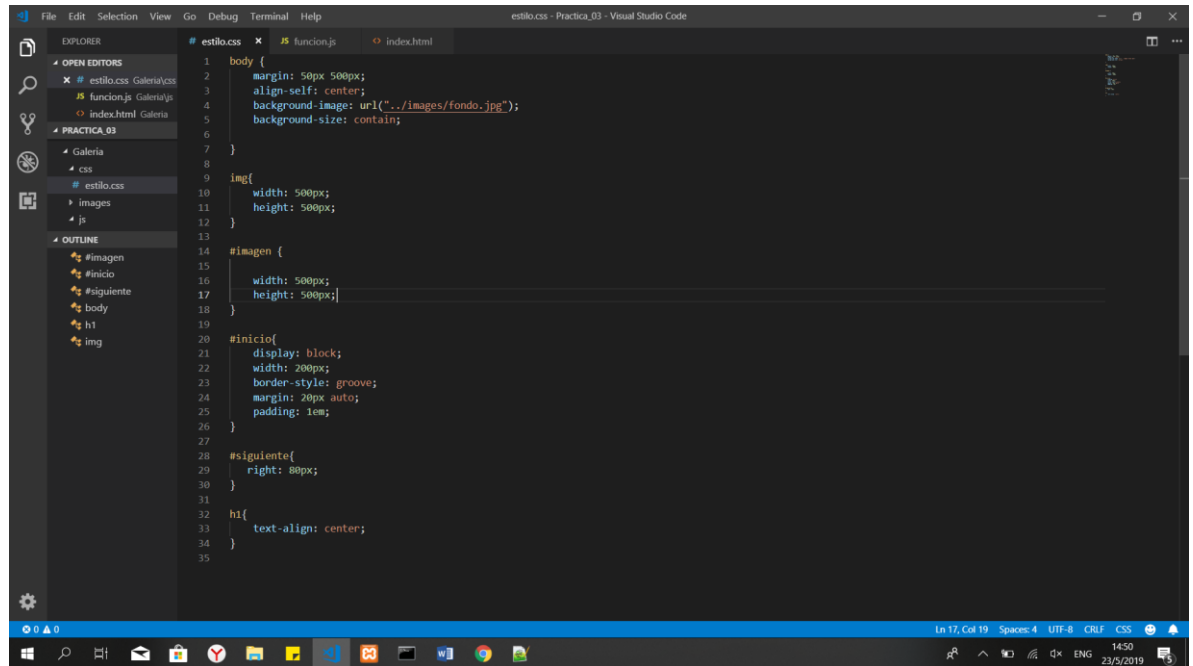
```

1  array = [{ "id": 1, "imagen": "imagenes/foto1.jpg" },
2  { "id": 2, "imagen": "imagenes/foto2.jpg" },
3  { "id": 3, "imagen": "imagenes/foto3.jpg" },
4  { "id": 4, "imagen": "imagenes/foto4.jpg" },
5  { "id": 5, "imagen": "imagenes/foto5.jpg" },
6  { "id": 6, "imagen": "imagenes/foto6.jpg" },
7  { "id": 7, "imagen": "imagenes/foto7.jpg" },
8  { "id": 8, "imagen": "imagenes/foto8.jpg" },
9  { "id": 9, "imagen": "imagenes/foto9.jpg" },
10 { "id": 10, "imagen": "imagenes/foto10.jpg" } ];
11 imagenes = [1, 2, 3, 4, 5];
12
13 var posActual = 0;
14
15 function inicio() {
16     for (i = 0; i < 5; i++) {
17         imagenes[i] = Math.floor(Math.random() * (10));
18     }
19     posActual = 0;
20     verificar();
21     imprimir();
22 }
23
24 function imprimir() {
25     console.log(imagenes);
26     var texto = "img src=" + array[imagenes[posActual]].imagen + ">";
27     window.document.getElementById("imagen").innerHTML = texto;
28 }
29
30 function verificar() {
31     if (posActual == 4) {
32         window.document.getElementById("siguiente").disabled = true;
33         window.document.getElementById("anterior").disabled = false;
34     }
35     else if (posActual == 0) {
36         window.document.getElementById("anterior").disabled = true;
37         window.document.getElementById("siguiente").disabled = false;
38     }
39     else {
40         window.document.getElementById("siguiente").disabled = false;
41         window.document.getElementById("anterior").disabled = false;
42     }
43 }
44
45 function siguiente() {
46     posActual++;
47     verificar();
48     imprimir();
49 }
50
51 function anterior() {
52     posActual--;
53     verificar();
54     imprimir();
55 }

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

• Código css.



```

1  body {
2      margin: 50px 500px;
3      align-self: center;
4      background-image: url("../images/fondo.jpg");
5      background-size: contain;
6  }
7
8
9  img{
10     width: 500px;
11     height: 500px;
12 }
13
14 #imagen {
15
16     width: 500px;
17     height: 500px;
18 }
19
20 #inicio{
21     display: block;
22     width: 200px;
23     border-style: groove;
24     margin: 20px auto;
25     padding: 1em;
26 }
27
28 #siguiente{
29     right: 80px;
30 }
31
32 h1{
33     text-align: center;
34 }
35

```

RESULTADO(S) OBTENIDO(S):


Se logró realizar las practicas, aunque existieron inconvenientes que a la final se pudieron ser solucionados.

CONCLUSIONES:

Este tipo de lenguajes de programación son de mucha importancia porque nos permiten elaborar muchos programas basadas en páginas web, nos permiten mantener un código ordenado con el desarrollador que se usó en esta ocasión Visual Estudio Code.

RECOMENDACIONES:

Mantener un código ordenado.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Nombre del estudiante: Diego Castro

Firma del estudiante jefe de grupo:

