

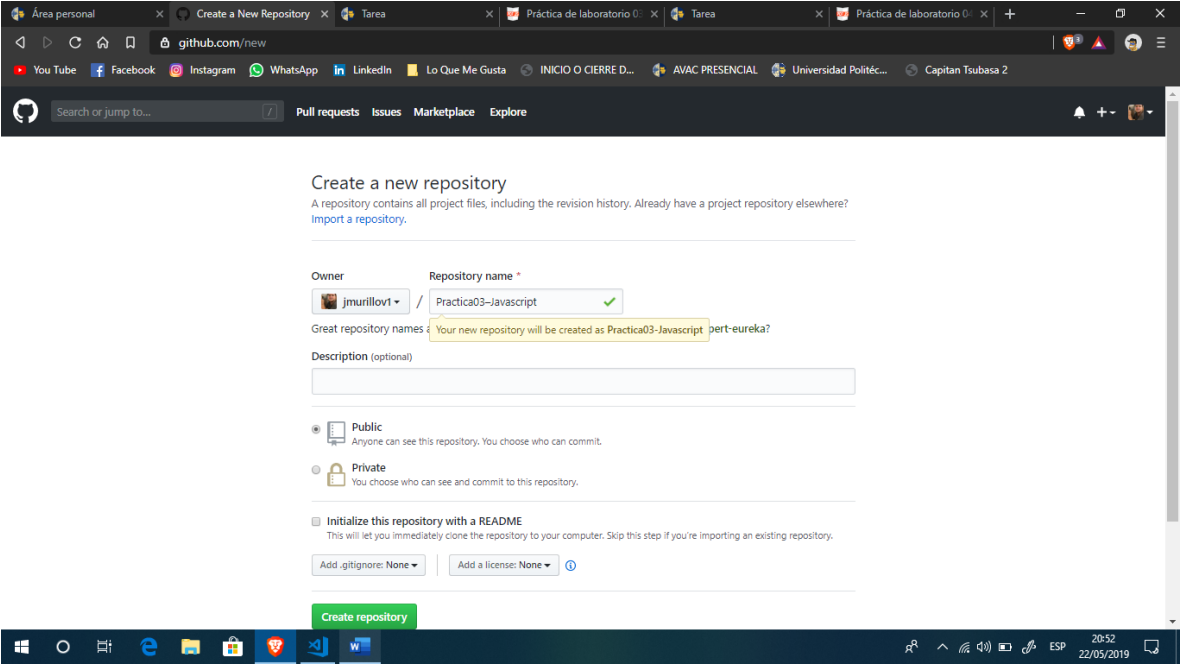

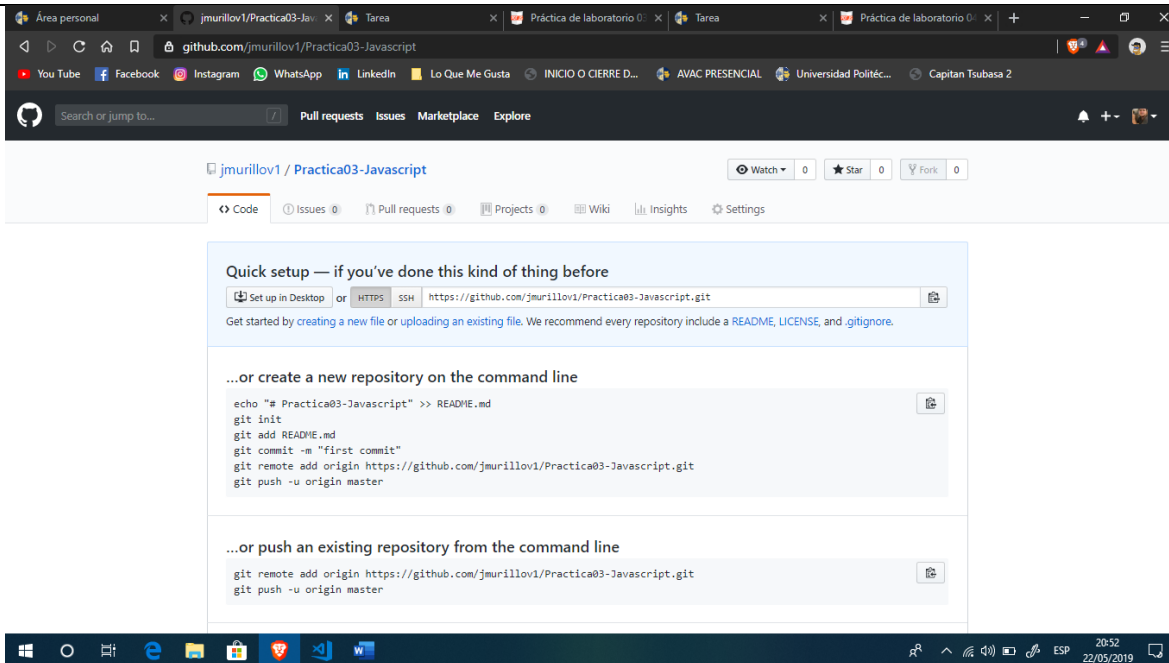


| | | |
|--|-------------------------------|-------------------------------|
|  | VICERRECTORADO DOCENTE | Código: GUIA-PRL-001 |
| | CONCEJO ACADÉMICO | Aprobación: 2016/04/06 |
| Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación | | |





| | | | |
|---|---|--|--|
|  | | FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES | |
| CARRERA: INGENIERÍA DE SISTEMAS | | ASIGNATURA: HYPERMEDIAL | |
| NRO. PRÁCTICA: | 3 | TÍTULO PRÁCTICA: Resolución de problemas sobre JavaScript | |
| OBJETIVO ALCANZADO: Entender y organizar de una mejor manera los sitios de web en Internet Diseñar adecuadamente elementos gráficos en sitios web en Internet. Crear sitios web aplicando estándares actuales. | | | |
| ACTIVIDADES DESARROLLADAS | | | |
| 1. Crear un repositorio en GitHub con el nombre “Practica03 – Javascript” Para realizar este paso previamente creamos una cuenta en GitHub, ahora conectado a la cuenta en línea procedemos a crear el repositorio donde guardaremos la práctica. | | | |
|  | | | |
| Luego tendremos la siguiente pantalla: | | | |

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



Con esto ya hemos creado nuestro repositorio. En este lugar nos dará la url del repositorio que es necesaria para configurar la forma de subir los archivos.

2. Crear una carpeta para la solución de cada ejercicio antes mencionado.

| | | |
|--|------------------|---------------------|
|  .git | 22/05/2019 21:20 | Carpeta de archivos |
|  1.Calculadora | 22/05/2019 21:07 | Carpeta de archivos |
|  2.Validaciones | 22/05/2019 21:03 | Carpeta de archivos |
|  3.Galeria | 22/05/2019 20:58 | Carpeta de archivos |

Evidencia de la creación de las carpetas para cada ejercicio.

3. Realizar un commit y push por cada requerimiento de los puntos antes descritos.

Para el desarrollo de esta práctica tendremos 3 puntos ejercicios a realizar haciendo uso de Javascript.

Para esto debemos abrir el cmd y colocar los siguientes comandos, en el caso de Windows:


- git init
- git add .
- git commit -m "Nombre-Proyecto"
- git remote add origin "url del repositorio"
- git push -u origin master

Estos para la primera vez que se vaya a trabajar con el proyecto, después solo es necesario correr:

- git add .
- git commit -m "Nombre-Proyecto"
- git push -u origin master

Y para que esto funcione previamente debes haber instalado y configurado "Git for Windows".

Ahora luego de correr los comandos tendremos:

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

```

C:\xampp\htdocs\Practica03-Javascript>git init
Reinitialized existing Git repository in C:/xampp/htdocs/Practica03-Javascript/.git/

C:\xampp\htdocs\Practica03-Javascript>git add .

C:\xampp\htdocs\Practica03-Javascript>git commit -m "Practica03-Javascript"
[master (root-commit) c49a5af] Practica03-Javascript
 4 files changed, 326 insertions(+)
 create mode 100644 1.Calculadora/css/estilo.css
 create mode 100644 1.Calculadora/images/icon.png
 create mode 100644 1.Calculadora/index.html
 create mode 100644 1.Calculadora/js/calculadora.js


C:\xampp\htdocs\Practica03-Javascript>git push -u origin master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (10/10), 7.67 KiB | 523.00 KiB/s, done.
Total 10 (delta 0), reused 0 (delta 0)
To https://github.com/jmurillov1/Practica03-Javascript.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

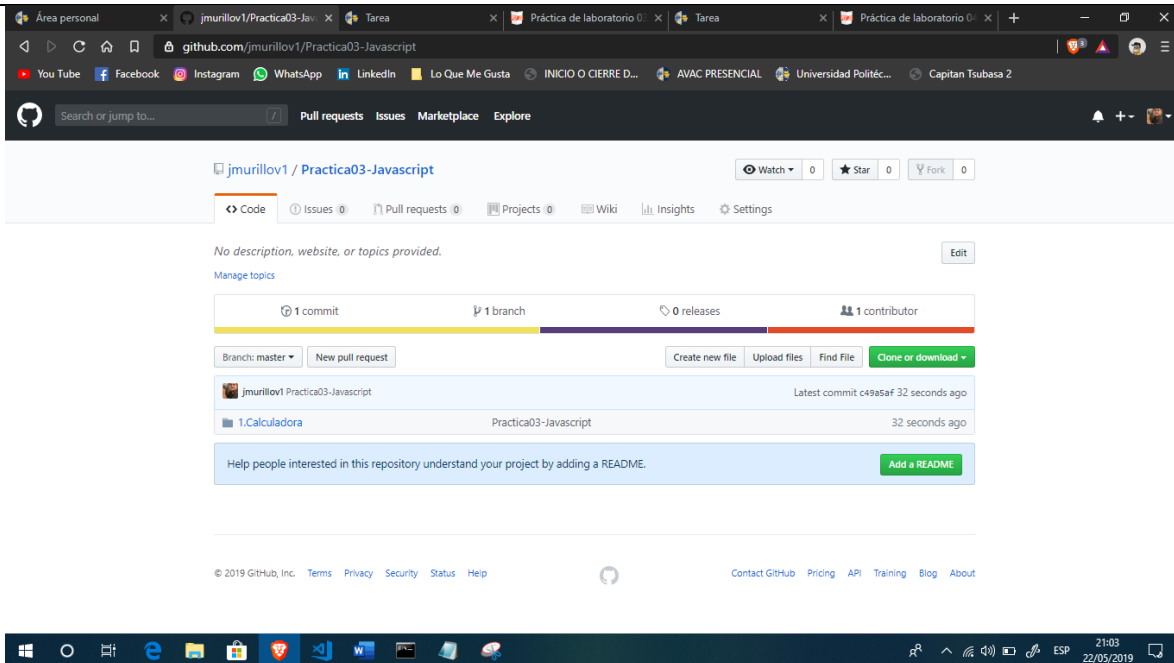
C:\xampp\htdocs\Practica03-Javascript>git remote add origin https://github.com/jmurillov1/Practica03-Javascript.git
fatal: remote origin already exists.

C:\xampp\htdocs\Practica03-Javascript>

```

Con el init iniciamos el repositorio vacío.
Con el add los añadimos.
Con el commit creamos los datos dentro del repositorio.
Con el remote le damos la dirección de donde subir los archivos.
Y con el push guardamos los datos en el repositorio.
Si todo corre bien deberíamos obtener lo siguiente:

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



Como pueden darse cuenta la pagina web en GitHub se actualiza con los nuevos datos subidos, con esto hemos cargado en la pagina el primer punto de la práctica.

El proyecto se fue haciendo poco a poco, como vimos primero voy a explicar el primer ejercicio que es la calculadora:

1. Calculadora

Se pide construir una calculadora en el lenguaje de programación de JavaScript con base a un formulario HTML usando botones y una caja de texto. Además, para que permita realizar operaciones aritméticas de complejidad básica, como: suma, resta, multiplicación, división, raíz cuadrada, entre otros. A continuación, se muestra un ejemplo de las operaciones que debe realizar la calculadora.


Parte HTML5:

```
<!DOCTYPE html>
<html>

<head>
  <title>Calculadora</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="css/estilo.css" />
  <script type="text/javascript" src="js/calculadora.js"></script>
</head>

<body>
  <div class="calculadora">
    <form action="#" name="calculadora" id="calculadora">
      <h6>
        
        Calculadora
      </h6>
      <p id="textoPantalla">0</p>
      <p>
        <input type="button" class="largo" value="&larr;" onclick="retro()" />
        <input type="button" value="CE" onclick="borradoParcial()" />
        <input type="button" value="C" onclick="borradoTotal()" />
        <input type="button" value="+/-" onclick="opuest()" />
        <input type="button" value="√" onclick="raizc()" />
      </p>
      <p>
        <input class="num" type="button" value="7" onclick="numero('7')" />
        <input class="num" type="button" value="8" onclick="numero('8')" />
        <input class="num" type="button" value="9" onclick="numero('9')" />
        <input type="button" value="/" onclick="operar('/')" />
        <input type="button" value="%" onclick="porcent()" />
      </p>
      <p>
        <input class="num" type="button" value="4" onclick="numero('4')" />
        <input class="num" type="button" value="5" onclick="numero('5')" />
        <input class="num" type="button" value="6" onclick="numero('6')" />
        <input type="button" value="*" onclick="operar('*')" />
        <input type="button" value="1/x" onclick="inve()" />
      </p>
      <p>
        <input class="num" type="button" value="1" onclick="numero('1')" />
        <input class="num" type="button" value="2" onclick="numero('2')" />
        <input class="num" type="button" value="3" onclick="numero('3')" />
        <input type="button" value="-" onclick="operar('-')" />
      </p>
      <p>
        <input class="num" id="cero" type="button" value="0" onclick="numero('0')" />
        <input type="button" value="." onclick="numero('.')" />
        <input type="button" value="+" onclick="operar('+)" />
        <input id="igual" type="button" value="=" onclick="igualar()" />
      </p>
    </form>
  </div>
</body>
```

El archivo se llama index.html, para esta parte hicimos uso de las etiquetas antes vistas y ordenamos los distintos componentes que conformar la calculadora, basada en la calculadora estándar de Windows 7. Escribimos las etiquetas básicas de estructura de un archivo html, y en la etiqueta body en dos se configura todo. Aquí dentro se

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

colocó todo en una etiqueta div para un manejo más fácil. Luego, en un form se colocó todo lo requerido por la calculadora, en el orden para que quede en base a la calculadora de referencia antes especificada. En una etiqueta h6 puse el título como el icono de la calculadora, y luego en etiquetas p puse las diferentes líneas de componentes que conforman la calculadora. Dentro de los p para hacer las teclas se hizo uso de etiquetas input de tipo button y para la pantalla de impresión también una etiqueta p con un gran espacio, para que muestre los resultados que se calculen.

En la etiqueta head colocamos la importación del archivo .css y el archivo .js los cuales se encargaron de que funcione correctamente la calculadora.

Además en la parte de los input se llama a los métodos de nuestro archivo .js, todos los eventos son onclick, ejemplo de uso: onclick="porcent()".

CSS3

```
/*aspectos generales: bordes y color de fondo de calculadora*/

#calculadora {
  background-color: #D9E4F1;
  width: 237px;
  height: 320px;
  border: 2.5px solid #f0e0b1;
  border-top: 20px solid #f0e0b1;
}

h6 {
  display: flex;
  margin: 0px;
  position: relative;
  top: -18px;
  align-items: center;
  font-size: 9pt;
  font-style: normal;
  font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;
}

/*pantalla de visualización: bordes, posición, color de fondo, estilo letra*/
```

```
#textoPantalla {
  display: block;
  font-size: 25px;
  text-align: right;
  position: relative;
  left: 7.5px;
  top: 5px;
  width: 210px;
  height: 50px;
  background-color: #f7f9fc;
  border-radius: 2.5px;
  margin-bottom: 10px;
}

/*botones normales: anchura y margen*/

#calculadora [type=button] {
  display: block;
  border: 1px solid gray;
  width: 37px;
  height: 37px;
  border-radius: 5px;
  margin: 4px;
  background-color: #ecf1f8;
}

p {
  margin: 4px;
  display: flex;
  width: 250px;
}
```

```

= .num {
  font-size: 20px;
}

= #calculadora p #cero {
  width: 82px;
}

= #calculadora p #igual {
  height: 86px;
  position: relative;
  top: -49px;
  font-size: 25px;
}

= #calculadora [type=button]:hover {
  background-image: linear-gradient(■ #FFEDD9, ■ #FFD574);
  box-shadow: 0 0 2px 1px ■ #FFD574;
}

= .largo {
  font-size: 25px;
  color: ■ black;
  font-weight: bold;
}

```

Para el archivo CSS tenemos los estilos que llevara para que la calculadora que con una buena estética y tenga parecido a la calculadora de referencia.

Para mayor facilidad de manejo se hizo uso de los selectores antes vistos, con lo cual se fue dando forma al requerimiento solicitado. Hicimos uso de colores para dar el color parecido a la calculadora de referencia.

Para la ubicación del botón igual se hizo uso de el posicionamiento relativo, y para ordenar según filas los botones si hizo uso de la propiedad de pantalla flex.

Con la ayuda de pseudo-classes se dio el efecto de cambio de color de los botones al colocar el mouse sobre el botón, la pseudo-clase usada es hover.

JAVASCRIPT

```

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.
}


```

Para esto use ayuda de internet, aquí tenemos 2 funciones la primera se encarga de obtener el valor de la pantalla de resultados. Luego se declaran las variables necesarias para trabajar y la segunda función se encarga mostrar los datos según el botón que se de click y lo que sea posible según la calculadora.

```
function operar(s) {
    igualar() //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 igualar() {
    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 raizc() {
    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
}
```

Continuamos con las funciones, ahora tenemos la función operar() que se le pasa el tipo de operación a realizar, luego la función igualar() que se encarga de revisar si existen operaciones pendientes y las evalúa. Seguido tenemos la función raíz que se encarga de calcular la raíz cuadrada del valor en pantalla.

```
function porcent() {
    x = x / 100 //dividir por 100 el número
    pantalla.innerHTML = x; //mostrar en pantalla
    igualar() //resolver y mostrar operaciones pendientes
    xi = 1 //reiniciar la pantalla
}
function opuest() {
    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 inve() {
    nx = Number(x);
    nx = (1 / nx);
    x = String(nx);
    pantalla.innerHTML = x;
    xi = 1; //reiniciar pantalla al pulsar otro número.
}
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
}
```

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

Continuando, tenemos la función porcent() que saca el porcentaje(división para 100 del número en pantalla) según el valor en pantalla, la función opuest() que saca el opuesto del valor en pantalla, seguido la función inve() que saca la división de 1 para el número en pantalla, luego la función retro() que borra el último dígito escrito.

```
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.
}
```

Por último, la función borradoParcial() que borra la pantalla y finalmente la función borradoTotal() que devuelve la pantalla a 0.

Commit y Push

```
C:\WINDOWS\system32\cmd.exe
C:\xampp\htdocs\Practica03-Javascript>git commit -m "Practica03-Javascript"
[master fc7f20c] Practica03-Javascript
2 files changed, 170 insertions(+), 100 deletions(-)
create mode 100644 1.Calculadora/css/estilo.css
delete mode 100644 1.Calculadora/js/calculadora.js


C:\xampp\htdocs\Practica03-Javascript>git push -u origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 1.35 KiB | 153.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To https://github.com/jmurillov1/Practica03-Javascript.git
   ae41d4b..fc7f20c  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

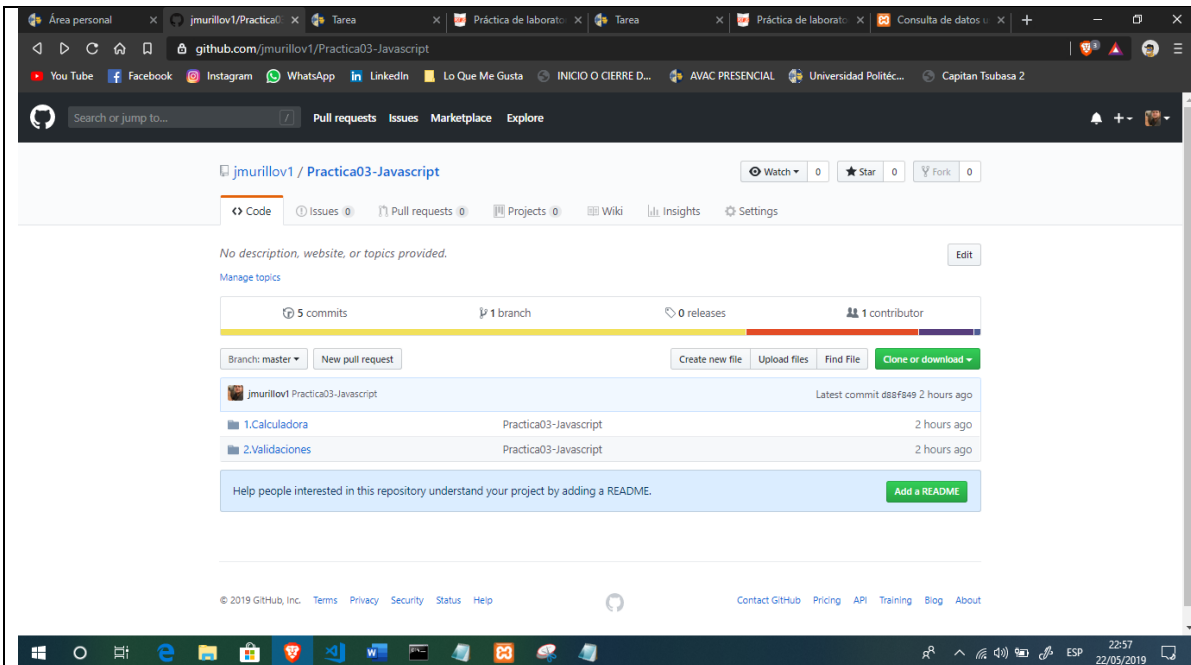
C:\xampp\htdocs\Practica03-Javascript>git add .

C:\xampp\htdocs\Practica03-Javascript>git commit -m "Practica03-Javascript"
[master d88f849] Practica03-Javascript
2 files changed, 101 insertions(+), 91 deletions(-)
create mode 100644 1.Calculadora/js/calculadora.js

C:\xampp\htdocs\Practica03-Javascript>git push -u origin master
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 1.88 KiB | 174.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/jmurillov1/Practica03-Javascript.git
   fc7f20c..d88f849  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

C:\xampp\htdocs\Practica03-Javascript>
```

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



Ahí hemos subido el primer ejercicio a nuestro GitHub.


2. Validaciones

Diseñar una interfaz en HTML que permita ingresar los siguientes campos en un formulario: cedula, nombres, apellidos, dirección, teléfono, fecha de nacimiento y correo electrónico. Luego, usando funciones de JavaScript se debe validar que todos los campos han sido ingresados, además; que los valores ingresados en cada campo del formulario sean correctos teniendo en cuenta las siguientes condiciones:

- Se debe validar qué, en el campo de la cedula, se ingrese sólo número y que la misma sea correcta, en base, al último dígito verificador.
- Se debe validar qué, en el campo de nombres, ingrese únicamente dos nombres y que permita ingresar sólo letras.
- Se debe validar qué, en el campo de apellido, ingrese únicamente dos apellidos y que permita ingresar sólo letras.
- Se debe validar qué, en el campo del teléfono, permita ingresar sólo números.
- Se debe validar que la fecha de nacimiento ingrese en el formato dd/mm/yyyy.
- Se debe validar qué, en el campo correo electrónico, permita ingresar un correo válido. Se considera un correo válido, cuando comienza por tres o más valores alfanuméricos, luego un @, seguido por la extensión "ups.edu.ec" o "est.ups.edu.ec".

Indicaciones:

- Para realizar las validaciones de solo letras, o sólo números. Se las debe realizar en tiempo real, es decir, a medida que el usuario escribe en el campo.
- Todos los campos de entrada dentro del formulario deben de ser de tipo "text".
- Las demás validaciones se realizarán al momento de "enviar" (submit) la información del formulario hacia una página php. Si no cumple las validaciones, se mostrará un mensaje debajo de cada campo con el error y se pintará el campo con un borde rojo que representará que el campo tiene un error. Si se cumple las validaciones, se enviará a una página php, en donde se mostrará únicamente un mensaje que diga "Bienvenido, ¡pasaste las validaciones!".

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

HTML5

```

<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>Sistema de Gestion de Usuarios</title>
  <link href="style.css" rel="stylesheet" type="text/css" />
  <script type="text/javascript" src="js/validaciones.js"></script>
</head>


<body>
  <header class="cab">
    <h1>Insertar Nueva Persona</h1>
  </header>
  <form id="formulario01" method="POST" action="controladores/saludo.php"
    onsubmit="return validarCamposObligatorios()">
    <label for="cedula">Cedula(*)</label>
    <input type="text" id="cedula" name="cedula" value="" placeholder="Ingrese el número de cedula... " />
    <span id="mensajeCedula" class="error"> </span>
    <br>
    <label for="nombres">Nombres (*)</label>
    <input type="text" id="nombres" name="nombres" value="" placeholder="Ingrese sus dos nombres..."
      onkeyup="valLetras(this)" />
    <span id="mensajeNombres" class="error"></span>
    <br>
    <label for="apellidos">Apellidos (*)</label>
    <input type="text" id="apellidos" name="apellidos" value="" placeholder="Ingrese sus dos apellidos..."
      onkeyup="valLetras(this)" />
    <span id="mensajeApellidos" class="error"></span>
    <br>
    <label for="direccion">Dirección (*)</label>
    <input type="text" id="direccion" name="direccion" value="" placeholder="Ingrese su dirección ..." />
    <span id="mensajeDireccion" class="error"></span>
    <span id="mensajeDireccion" class="error"></span>
    <br>
    <label for="telefono">Teléfono (*)</label>
    <input type="text" id="telefono" name="telefono" value="" placeholder="Ingrese su número telefónico..."
      onkeyup="valNumeros(this)" />
    <span id="mensajeTelefono" class="error"></span>
    <br>
    <label for="fecha">Fecha Nacimiento (*)</label>
    <input type="text" id="fechaNacimiento" name="fechaNacimiento" value=""
      placeholder="Ingrese su fecha de nacimiento ..." />
    <span id="mensajeFechaNacimiento" class="error"></span>
    <br>
    <label for="correo">Correo electrónico (*)</label>
    <input type="text" id="correo" name="correo" value="" placeholder="Ingrese su correo electrónico..." />
    <span id="mensajeCorreo" class="error"></span>
    <br>
    <label for="contrasena">Contraseña (*)</label>
    <input type="password" id="contrasena" name="contrasena" value="" placeholder="Ingrese su contraseña..." />
    <span id="mensajeContrasena" class="error"></span>
    <br>
    <input type="submit" id="crear" name="crear" value="Aceptar" />
    <input type="reset" id="cancelar" name="cancelar" value="Cancelar" />
  </form>
</body>

</html>

```

Creamos el archivo .html necesario llamado crear_usuario.html del cual realizaremos las validaciones y nos devolverá un mensaje si está bien al hacer el envío de datos a través del formulario.

En la etiqueta body importamos el archivo .css y .js para el funcionamiento del trabajo. Para nuestro caso utilizaremos una etiqueta form con el método POST para enviar los datos, en acción la ruta del archivo donde se enviar los datos con extensión .php, y el evento a usar que es onsubmit="validarCamposObligatorios()" para la

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

mayoría de datos a enviar, solo para los nombres apellidos y números se usara el evento onkeyup porque se debe validar en tiempo real. Utilizamos un label para poner a que se refieren los campos de entrada de datos, en inputs los campos para ingreso de datos, de tipo text, al final un dos input, uno de tipo submit para enviar y uno de tipo reset para cancelar.

CSS3

```

label {
  margin-top: 20px;
}

.error {
  color: red;
  font-size: 10px;
}

form {
  margin-left: 25px;
  width: 400px;
  float: left;
}


input {
  width: 380px;
  height: 27px;
}

header.cab {
  background-color: green;
}

body {
  background-color: aliceblue;
}

```

El archivo .css para este ejercicio es muy básico ya que los estilos según las validaciones las vamos a dar con ayuda de javascript. En este archivo le damos el estilo a nuestro formulario.

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

PHP

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>

<body>
    <h2> Bienvenido, pasaste las validaciones <?php echo $_POST["nombres"] ?></h2>
</body>

</html>

```

Este archivo es donde se llegará, si al pasar el formulario por las validaciones todo resulta bien. Nos devuelve un saludo de que todo fue un éxito.

JAVASCRIPT

```

function validarCamposObligatorios() {
    var bandera = false
    var banderav = true
    var banderac = false
    var banderae = false
    var banderaf = false
    for (var i = 0; i < document.forms[0].elements.length - 2; i++) {
        var elemento = document.forms[0].elements[i]
        if (elemento.value == '' && (elemento.type == 'text' || elemento.type == 'password')) {
            if (elemento.id == 'cedula') {
                document.getElementById('mensajeCedula').innerHTML = '<br>La cedula esta vacía'
            } else if (elemento.id == 'nombres') {
                document.getElementById('mensajeNombres').innerHTML = '<br>Los nombres estan vacíos'
            } else if (elemento.id == 'apellidos') {
                document.getElementById('mensajeApellidos').innerHTML = '<br>Los apellidos estan vacíos'
            } else if (elemento.id == 'direccion') {
                document.getElementById('mensajeDireccion').innerHTML = '<br>La dirección esta vacía'
            } else if (elemento.id == 'telefono') {
                document.getElementById('mensajeTelefono').innerHTML = '<br>El telefono está vacío'
            } else if (elemento.id == 'fechaNacimiento') {
                document.getElementById('mensajeFechaNacimiento').innerHTML = '<br>La fecha de nacimiento esta vacía'
            } else if (elemento.id == 'correo') {
                document.getElementById('mensajeCorreo').innerHTML = '<br>El correo esta vacío'
            } else if (elemento.id == 'contrasena') {
                document.getElementById('mensajeContrasena').innerHTML = '<br>La contraseña esta vacía'
            }
            elemento.style.border = "1px red solid"
            banderav = false
        }
    }
}

```



```

    }
    if (elemento.id == 'cedula') {
        if (valCedula(elemento.id) == true) {
            banderac = true
        } else {
            elemento.style.border = "1px red solid"
        }
    }
    if (elemento.id == 'correo') {
        if (valCorreo(elemento.id) == true) {
            banderae = true
        } else {
            elemento.style.border = "1px red solid"
        }
    }
    if (elemento.id == 'fechaNacimiento') {
        if (valFecha(elemento.id) == true) {
            banderaf = true
        } else {
            elemento.style.border = "1px red solid"
        }
    }
}
if (banderav == true && banderac == true && banderae == true && banderaf == true) {
    bandera = true
}
console.log(flag)
return flag
}

```

Esta es la función general que al pasar por el submit valida los datos ingresados, primero comprueba que todos los datos sean diferentes de vacío para pasar, esto lo hacemos realizando un recorrido de los elementos que tiene el formulario con un bucle for y preguntando si son de tipo text, luego si pasa eso me devuelve una variable si es falso y me pinta de rojo el borde y me dice el error que tengo, para facilidad del usuario del sistema.

Luego en otro if llamo al método siguiente para validar la cedula, si me devuelve true pasa el requisito, luego el correo y la fecha de nacimiento, se declaro una variable para cada una de estas preguntas de tipo boolean que inician en false, a excepción de una que inicia en true, que se vuelve falsa si algún input este vacío y no deja enviar el formulario.


Cada una de las preguntas que se hacen con los if devuelve un valor true si se cumple y al final pregunta si todas son true la general se vuelve true y se retorna para que se envíen los datos por el formulario.

```

function validar(datos) {
    var na = document.getElementById(datos.id).value
    if (datos.id == 'nombres') {
        var n = na.substr(na.length - 1).charCodeAt(0)
        if ((n >= 65 && n <= 90) || (n >= 97 && n <= 122) || n == 32 || n == 225 || n == 233 || n == 237 || n == 243 || n == 250 || n == 192) {
            nom = na.split(" ").length
            if (nom > 2) {
                datos.style.border = "1px red solid"
                document.getElementById('mensajeNombres').innerHTML = '<br>Solo se permiten dos nombres'
            }
        } else {
            document.getElementById('nombres').value = na.substr(0, na.length - 1)
        }
    } else if (datos.id == 'apellidos') {
        var n = na.substr(na.length - 1).charCodeAt(0)
        if ((n >= 65 && n <= 90) || (n >= 97 && n <= 122) || n == 32 || n == 225 || n == 233 || n == 237 || n == 243 || n == 250 || n == 192) {
            ape = na.split(" ").length
            if (ape > 2) {
                datos.style.border = "1px red solid"
                document.getElementById('mensajeApellidos').innerHTML = '<br>Solo se permiten dos apellidos'
            }
        } else {
            document.getElementById('apellidos').value = na.substr(0, na.length - 1)
        }
    }
}

```

El método de aquí sirve para validar ya sea los nombres o apellidos, como pide el trabajo, primero recibe el objeto que lo llama, obtiene el valor del input y pregunta si es Nombre o Apellido, luego cada carácter lo convierte a ASCII

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

para facilitar la pregunta de si son letras, si cumple estos ingresa y divide el contenido con split, para saber si hay dos palabras, si supera las dos palabras da error de que no cumple con los requisitos, y si un carácter no pertenece a los que se permiten se borra automáticamente, esto aplica tanto para Nombre y Apellidos.


```
function valFecha(valor) {
    var fecha = document.getElementById(valor).value
    if (fecha.length == 10 && fecha != '') {
        var dia = fecha.substr(0, 2)
        var mes = fecha.substr(3, 2)
        var año = fecha.substr(6, 4)
        var s1 = fecha.substr(2, 1)
        var s2 = fecha.substr(5, 1)
        var val1 = false
        var val2 = false
        var val3 = false
        var val4 = false
        var vals = false
        var valg = true
        añov = parseInt(año)
        diav = parseInt(dia)

        if (mes == '01' || mes == '03' || mes == '05' || mes == '07' || mes == '08' || mes == '10' || mes == '12') {
            diav = parseInt(dia)
            if (diav >= 1 && diav <= 31) {
                val1 = true;
            }
        } else if (mes == '02') {
            if (añov % 4 == 0) {
                if (diav >= 1 && diav <= 29) {
                    val2 = true;
                }
            } else {
                if (diav >= 1 && diav <= 28) {
                    val3 = true;
                }
            }
        }

        } else if (mes == '04' || mes == '06' || mes == '09' || mes == '11') {
            if (diav >= 1 && diav <= 30) {
                val4 = true;
            }
        }
    }

    var fechaActual = new Date();
    var diaActual = fechaActual.getDate()
    var mesActual = fechaActual.getMonth() + 1
    var añoActual = fechaActual.getFullYear()
    if (s1 == '/' && s2 == '/') {
        vals = true
    } else {
        document.getElementById('mensajeFechaNacimiento').innerHTML = '<br> El formato de fecha es incorrecto'
        valg = false
    }

    if (parseInt(año) < añoActual) {
        if ((val1 == true || val2 == true || val3 == true || val4 == true) && vals == true) {
            document.getElementById('mensajeFechaNacimiento').innerHTML = ''
        }
    } else if (parseInt(año) == añoActual) {
        if ((val1 == true || val2 == true || val3 == true || val4 == true) && vals == true && parseInt(mes) <= mesActual && parseInt(dia) <= diaActual) {
            document.getElementById('mensajeFechaNacimiento').innerHTML = ''
        } else {
            if (vals == false) {
                document.getElementById('mensajeFechaNacimiento').innerHTML = '<br>El formato de fecha es incorrecto'
                valg = false
            } else {
                document.getElementById('mensajeFechaNacimiento').innerHTML = '<br>La fecha es incorrecta'
                valg = false
            }
        }
    } else {
        document.getElementById('mensajeFechaNacimiento').innerHTML = '<br>La fecha es incorrecta'
        valg = false
    }
}
```

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

```

    } else {
        document.getElementById('mensajeFechaNacimiento').innerHTML = '<br>El año es incorrecto'
        valg = false;
    }
    if (val1 == false && val2 == false && val3 == false && val4 == false) {
        document.getElementById('mensajeFechaNacimiento').innerHTML = '<br>La fecha es incorrecta'
        valg = false
    }
    } else {
        document.getElementById('mensajeFechaNacimiento').innerHTML = '<br>La fecha es incorrecta o está vacía'
        valg = false
    }
    return valg;
}

```

Ahora validamos la fecha, con ayuda de la función substr(), cortamos los datos necesarios como son día, mes y año, primero validamos con ayuda de ifs que los días sean los permitidos, según el mes y según el año si es bisiesto.

Luego validamos si el día que ingresamos es mayor al día actual, seguido de el formato, que no es mas que preguntar si los símbolos slash están donde es su lugar, si cumple las validaciones se hacen true, por una falsa ya no la acepta, caso contrario si todo va bien retorna un true que nos sirve en la función general.

```

function valNumeros(datos) {
    var nums = document.getElementById(datos.id).value
    if (datos.id == 'telefono') {
        document.getElementById('mensajeTelefono').innerHTML = ''
        var n = nums.substr(nums.length - 1).charCodeAt(0)
        if (n >= 48 && n <= 57) {
            if (nums.length > 10) {
                datos.style.border = "1px red solid"
                document.getElementById('mensajeTelefono').innerHTML = '<br>Número de teléfono incorrecto'
            }
        } else {
            document.getElementById('telefono').value = nums.substr(0, nums.length - 1)
        }
    }
}

```

Ahora validamos el teléfono, que así mismo recibe el elemento de donde se llama al método. Se obtiene el valor del input en una variable, que luego cada carácter se pasa a ASCII para facilitar las preguntas en el if, si cumple pasa y pregunta si es mayor a 10 el numero se pinta de rojo el input en cuestión, si no pertenecen los caracteres ingresados a los permitidos se borran automáticamente en tiempo real.

```
function valCedula(valor) {
    var ced = document.getElementById(valor).value.trim()
    var bandera = true;
    var valp = false
    var valtd = false
    var valud = false
    var total = 0
    if (ced.length != 10 || ced.value == '') {
        document.getElementById('mensajeCedula').innerHTML = '<br>Número de cédula vacío, incompleto o excedido'
        bandera = false;
    } else {
        document.getElementById('mensajeCedula').innerHTML = ''
        var prov = parseInt(ced.substr(0, 2))
        if (prov >= 1 && prov <= 24) {
            valp = true
        } else {
            bandera = false;
            document.getElementById('mensajeCedula').innerHTML = '<br>No existe la provincia - 01 a 24'
        }
        var terd = parseInt(ced.substr(2, 1))
        if (terd >= 0 && terd <= 5) {
            valtd = true
        } else {
            bandera = false;
            document.getElementById('mensajeCedula').innerHTML = '<br>Tercer Dígito Incorrecto - 0 a 5'
        }
    }
    var lon = ced.length
    for (var i = 0; i < lon - 1; i++) {
        if (i % 2 == 0) {
            var digp = ced.charAt(i) * 2
            if (digp > 9) digp -= 9
            total += digp
        } else {
            total += parseInt(ced.charAt(i))
        }
    }
    var ud = total % 10 ? 10 - total % 10 : 0;
    if (ud == parseInt(ced.substr(lon - 1, 1))) {
        valud = true
    } else {
        bandera = false;
        document.getElementById('mensajeCedula').innerHTML = '<br>Último Dígito Incorrecto'
    }

    if (valp == true && valtd == true && valud == true) {
    } else if (valp == false && valtd == false) {
        bandera = false;
        document.getElementById('mensajeCedula').innerHTML = '<br>Provincia (01-24) y Tercer Dígito (0-5) Incorrectos'
    }
}
return bandera;
}
```


Aquí le pasamos el valor del elemento desde la función general, luego preguntamos si es igual a 10 la longitud desarrollamos la comprobación de la cédula ecuatoriana, multiplicamos por 2 los números en posiciones pares y por uno en posiciones impares, si la multiplicación por 2 es mayor a 9 se le resta 9 y luego se hace una suma de estos nuevos valores, luego el valor de la suma lo subo al inmediato superior, y restamos el valor de la suma de su inmediato superior y si este numero es igual al ultimo dígito se valida como true.

```
function valCorreo(valor) {
    var correo = document.getElementById(valor).value.trim()
    bandera = true
    var lon = correo.length
    var cm = '@ups.edu.ec'
    var ce = '@est.ups.edu.ec'
    var cont = 0
    if (lon <= 70) {
        var cm1 = correo.substr(lon - 11, lon - 1)
        var ce1 = correo.substr(lon - 15, lon - 1)
        if (cm1 == cm && lon >= 14) {
            var us = correo.substr(0, lon - 11)
            console.log(us)
            for (var i = 0; i < us.length; i++) {
                var n = us.substr(i).charAt(0)
                console.log(n)
                if ((n >= 65 && n <= 90) || (n >= 97 && n <= 122) || (n >= 48 && n <= 57)) {
                    document.getElementById('mensajeCorreo').innerHTML = ''
                } else {
                    cont++;
                }
            }
            if (cont > 0) {
                bandera = false;
                document.getElementById('mensajeCorreo').innerHTML = 'Caracteres no son alfanumericos'
            }
        }
    } else if (ce1 == ce && lon >= 18) {
        var us = correo.substr(0, lon - 15)
        console.log(us)
        for (var i = 0; i < us.length; i++) {
            var n = us.substr(i).charAt(0)
            console.log(n)
            if ((n >= 65 && n <= 90) || (n >= 97 && n <= 122) || (n >= 48 && n <= 57)) {
                document.getElementById('mensajeCorreo').innerHTML = ''
            } else {
                cont++;
            }
        }
        if (cont > 0) {
            bandera = false;
            document.getElementById('mensajeCorreo').innerHTML = 'Caracteres no son alfanumericos'
        }
    } else {
        if (lon == 0) {
            document.getElementById('mensajeCorreo').innerHTML = '<br>El correo esta vacío'
        } else {
            document.getElementById('mensajeCorreo').innerHTML = '<br>Fuera Requisitos - Tener 3 caracteres alfanumericos como mínimo'
            bandera = false
        }
    }
    } else {
        bandera = false;
        document.getElementById('mensajeCorreo').innerHTML = '<br> El correo excede el tamaño establecido'
    }
    return bandera;
}
```

Ahora para validar el correo, al método le pasamos el id del elemento, luego con ese obtenemos el valor del input y preguntamos si es menor o igual a 70 lo validamos, si lo es obtenemos su última parte según la validación ya sea @ups.edu.ec o @est.ups.edu.ec, luego preguntamos si lo obtenido es igual a lo inicial y si es mayor o igual a 14 en el caso de @ups.edu.ec o en el else si es mayor o igual a 18 e igual a @est.ups.edu.ec, si cumple esto, para cada caso recorreremos el valor de input y preguntaremos si hay mínimo 3 caracteres antes de las terminaciones requeridas y con ASCII validamos si son caracteres alfanuméricos.

Y si cumple retornará un true.

Commit y Push

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

```

C:\WINDOWS\system32\cmd.exe

create mode 100644 1.Calculadora/css/estilo.css
delete mode 100644 1.Calculadora/js/calculadora.js

C:\xampp\htdocs\Practica03-Javascript>git push -u origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 1.35 KiB | 153.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To https://github.com/jmurillov1/Practica03-Javascript.git
   ae41d4b..fc7f20c  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

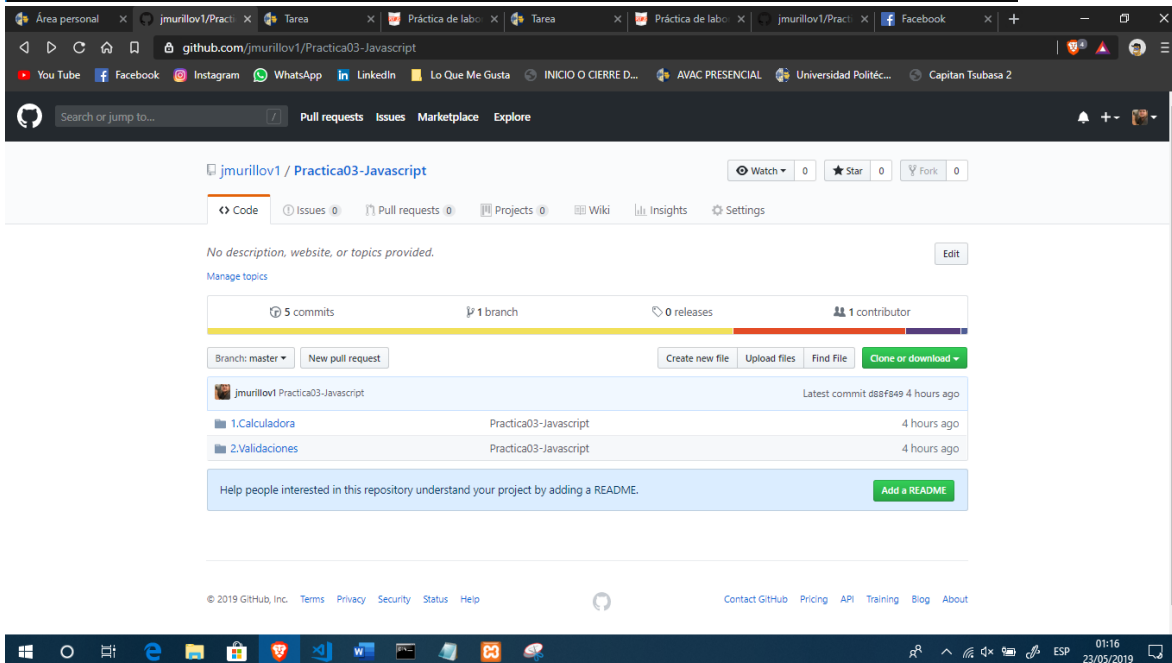
C:\xampp\htdocs\Practica03-Javascript>git add .


C:\xampp\htdocs\Practica03-Javascript>git commit -m "Practica03-Javascript"
[master d88f849] Practica03-Javascript
 2 files changed, 101 insertions(+), 91 deletions(-)
 create mode 100644 1.Calculadora/js/calculadora.js

C:\xampp\htdocs\Practica03-Javascript>git push -u origin master
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 1.88 KiB | 174.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/jmurillov1/Practica03-Javascript.git
   fc7f20c..d88f849  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

C:\xampp\htdocs\Practica03-Javascript>

```



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

3. Galería

Diseñar una interfaz en html que tenga tres botones que diga “Anterior”, “Iniciar”, “Siguiente”, y una imagen. Luego, desde javascript se debe controlar para al hacer clic sobre uno de los botones realice una acción relacionada a una galería de imágenes (ver ejemplo, <https://gihp4c.blog.ups.edu.ec/>)

Indicaciones:

- Se debe tener, un arreglo con los nombres de diez imágenes, previamente descargadas y almacenadas en una carpeta llamada “imagen”.
- La galería de imágenes debe visualizar exclusivamente 5 imágenes.
- Cada vez que se haga clic en el botón iniciar se deben escoger de manera aleatoria cinco imágenes de las diez que se mostrarán en la galería de imágenes.
- Al hacer clic en el botón siguiente y haber llegado a la última imagen disponible, el botón siguiente deberá ser deshabilitado (sólo cuando se ha llegado a la última imagen el botón siguiente deberá estar deshabilitado)
- Al hacer clic en el botón anterior y haber llegado a la primera imagen disponible, el botón anterior deberá ser deshabilitado (sólo cuando se ha llegado a la última imagen y cuando se inicie la galería de imágenes el botón anterior deberá estar deshabilitado)

HTML5

```

<!DOCTYPE html>
<html lang="es">

<head>
  <meta charset="UTF-8">
  <title>Consulta de datos usando AJAX</title>
  <script type="text/javascript" src="js/galeria.js"> </script>
  <link href="css/style.css" rel="stylesheet" type="text/css" />
</head>

<body onload="init()">
  <center>
    <h2>Galería de Fotos</h2>
    <img id="imagen" />
  </center>

  <center>
    <input type="button" id="anterior" name="anterior" value="Anterior" onclick="anterior()" />
    <input type="button" id="iniciar" name="iniciar" value="Iniciar" onclick="iniciar()" />
    <input type="button" id="siguiente" name="siguiente" value="Siguiente" onclick="siguiente()" />
  </center>

</body>

</html>

```

Aquí tenemos el archivo .html que se llama index, donde tenemos escritas las etiquetas necesarias para nuestra actividad, como dije anteriormente en la etiqueta head se importa tanto el archivo .css como .js para que funcione la actividad.

En el body tenemos, una etiqueta para el título, la etiqueta img para visualizar la imagen dentro de una etiqueta center para alinear al centro, así mismo dentro de otra etiqueta center tenemos los inputs para los botones de Anterior, Iniciar, Siguiente alineados al centro.

Para llamar al método tenemos el evento onload para cuando se cargue el body se visualice las 5 imágenes seleccionadas aleatoriamente, caso contrario si quiero diferentes imágenes hacemos click en iniciar y cambiarán, los botones cada uno lleva un evento de onclick.

CSS3

```
#imagen {
  width: 80%;
  height: 500px;
  border-radius: 10px;
}

body {
  background-color: darkgray;
}

input {
  border-radius: 5px;
  width: 90px;
  height: 30px;
}
```

No es mucho uso de reglas css, solo una para las imágenes, el color de fondo del body y otra para los input, los botones.

JAVASCRIPT


```
var fotos = ['foto1', 'foto2', 'foto3', 'foto4', 'foto5', 'foto6', 'foto7', 'foto8', 'foto9', 'foto10']
var alet = []
var pos = 0

function init() {
  pos = 0
  iniciar()
  show()
  loc()
}

function loc() {
  if (pos == 0) {
    document.getElementById("siguiente").disabled = false;
    document.getElementById("anterior").disabled = true;
  } else if (pos == 4) {
    document.getElementById("siguiente").disabled = true;
    document.getElementById("anterior").disabled = false;
  } else {
    document.getElementById("siguiente").disabled = false;
    document.getElementById("anterior").disabled = false;
  }
}

function anterior() {
  pos--
  loc()
  show()
}
```

Como pide el trabajo esta declarado el arreglo con el nombre de las fotos, el nuevo arreglo para las fotos aleatorias resultantes, y la posición para el uso de los botones.

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

La función `init()` se encarga de cuando el body cargue, la posición es 0, llame al método `iniciar()` para que se carguen 5 imágenes aleatorias, `show()` para mostrar la imagen en la posición 0 del arreglo y `loc()` para validar donde estamos y así saber cuándo deshabilitar los botones que lo requieran.

La función `loc()` se encarga de identificar el lugar en donde nos encontramos, si es 0 deshabilita el botón Anterior, si esta entre 0 y 4, ambos están deshabilitados y si está en la posición 4 deshabilita el botón Siguiente.

La función `anterior()` se encarga de hacer que se retroceda, restándole 1 a la posición actual, llamando al método `loc()` para saber dónde estamos y al método `show()` para mostrar la imagen.

```
function iniciar() {
    pos = 0
    loc()
    for (var i = 0; i < 5; i++) {
        var al = Math.round(Math.random() * 9)
        while (alet.includes(fotos[al])) {
            var al = Math.round(Math.random() * 9)
        }
        alet[i] = fotos[al]
    }
    show()
}

function show() {
    var url = 'images/' + alet[pos] + '.jpg'
    document.getElementById('imagen').src = url
}

function siguiente() {
    pos++
    loc()
    show()
}
```

La función `iniciar()` carga las 5 imágenes aleatorias tomadas del arreglo principal, como lo dice su nombre `iniciar`, la posición se vuelve a cero, llama al método `loc()` para saber la ubicación, luego recorre un bucle para darle los valores de las imágenes aleatorias a un nuevo arreglo con el cual se trabaja, ahí crea un número aleatorio entre 0 y 9, los cuales representan la ubicación de las fotos aleatorias escogidas del arreglo principal, luego un `while` pregunta si el valor obtenido ya se encuentra en el arreglo, si lo está calcula de nuevo hasta obtener uno diferente, luego le damos al nuevo arreglo el valor obtenido del arreglo general en la posición que nos da el aleatorio y finalmente muestra la imagen, según lo obtenido.

La función `show()` se encarga de mostrar la imagen según el valor del arreglo.

La función `siguiente()` se encarga de hacer que se vaya hacia adelante, sumándole 1 a la posición actual, llamando al método `loc()` para saber dónde estamos y al método `show()` para mostrar la imagen.

Commit y Push


```
C:\WINDOWS\system32\cmd.exe


Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 1.88 KiB | 174.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/jmurillov1/Practica03-Javascript.git
   fc7f20c..d88f849  master -> master
branch 'master' set up to track remote branch 'master' from 'origin'.

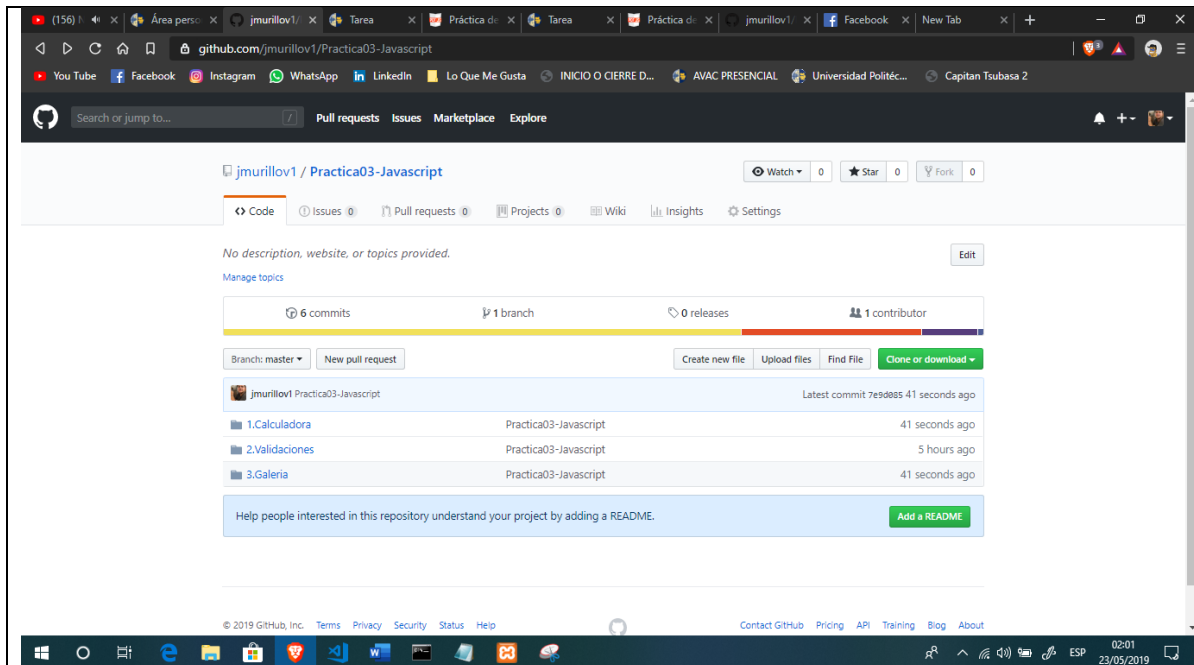
C:\xampp\htdocs\Practica03-Javascript>git add .

C:\xampp\htdocs\Practica03-Javascript>git commit -m "Practica03-Javascript"
[master 7e9d085] Practica03-Javascript
 14 files changed, 93 insertions(+), 1 deletion(-)
 create mode 100644 3.Galeria/css/style.css
 create mode 100644 3.Galeria/images/foto1.jpg
 create mode 100644 3.Galeria/images/foto10.jpg
 create mode 100644 3.Galeria/images/foto2.jpg
 create mode 100644 3.Galeria/images/foto3.jpg
 create mode 100644 3.Galeria/images/foto4.jpg
 create mode 100644 3.Galeria/images/foto5.jpg
 create mode 100644 3.Galeria/images/foto6.jpg
 create mode 100644 3.Galeria/images/foto7.jpg
 create mode 100644 3.Galeria/images/foto8.jpg
 create mode 100644 3.Galeria/images/foto9.jpg
 create mode 100644 3.Galeria/index.html
 create mode 100644 3.Galeria/js/galeria.js

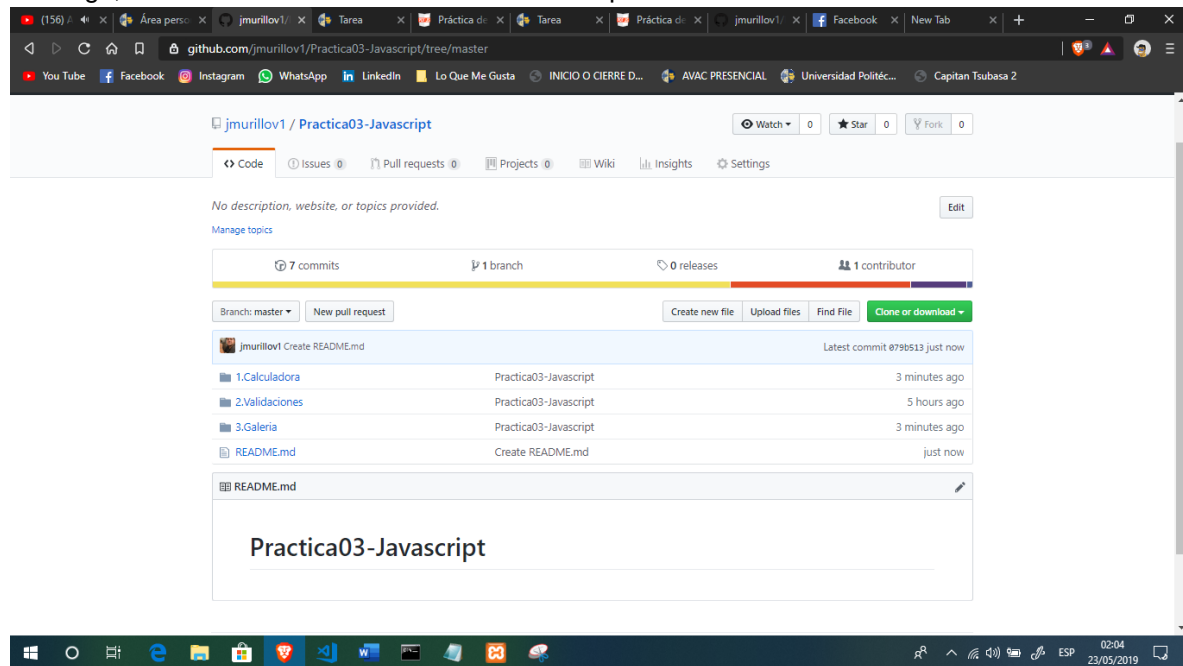
C:\xampp\htdocs\Practica03-Javascript>git push -u origin master
Enumerating objects: 26, done.
Counting objects: 100% (26/26), done.
Delta compression using up to 4 threads
Compressing objects: 100% (19/19), done.
Writing objects: 100% (22/22), 8.85 MiB | 3.97 MiB/s, done.
Total 22 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/jmurillov1/Practica03-Javascript.git
   d88f849..7e9d085  master -> master
branch 'master' set up to track remote branch 'master' from 'origin'.

C:\xampp\htdocs\Practica03-Javascript>
```

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



4. Luego, se debe crear el archivo README del repositorio de GitHub.




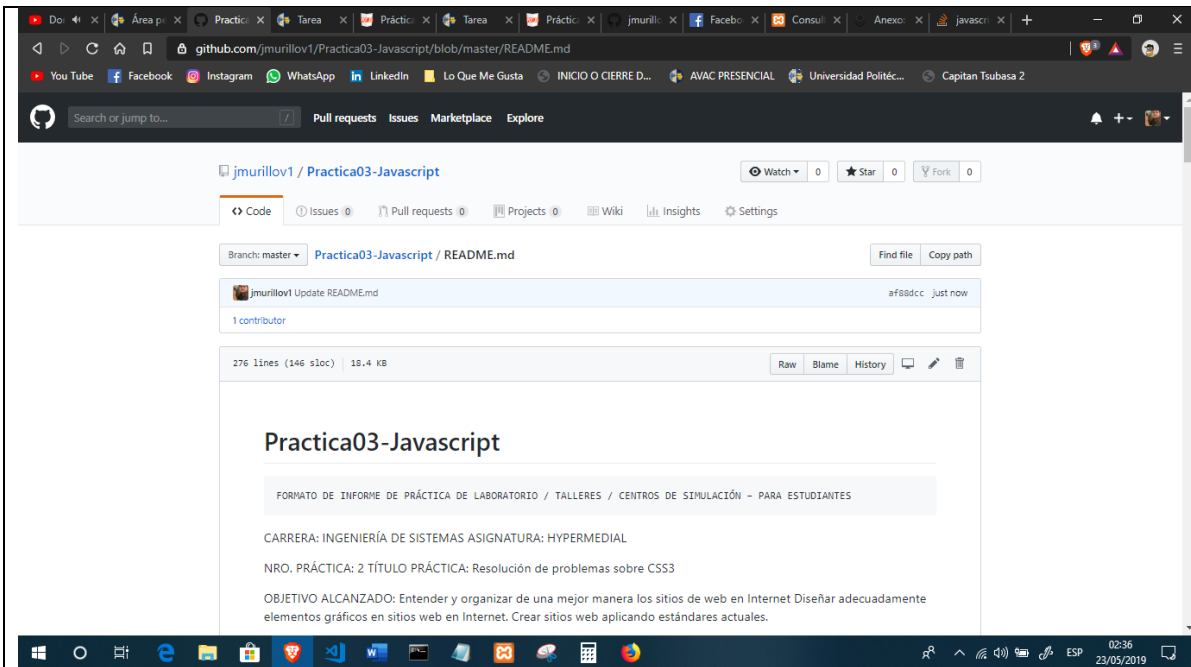
5. Información de GitHub (usuario y URL del repositorio de la práctica).

Usuario: jmurillov1

URL del repositorio: <https://github.com/jmurillov1/Practica03-Javascript>

6. En el archivo README del repositorio debe constar la misma información del informe de resultados de la práctica que se indica en el siguiente punto.

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



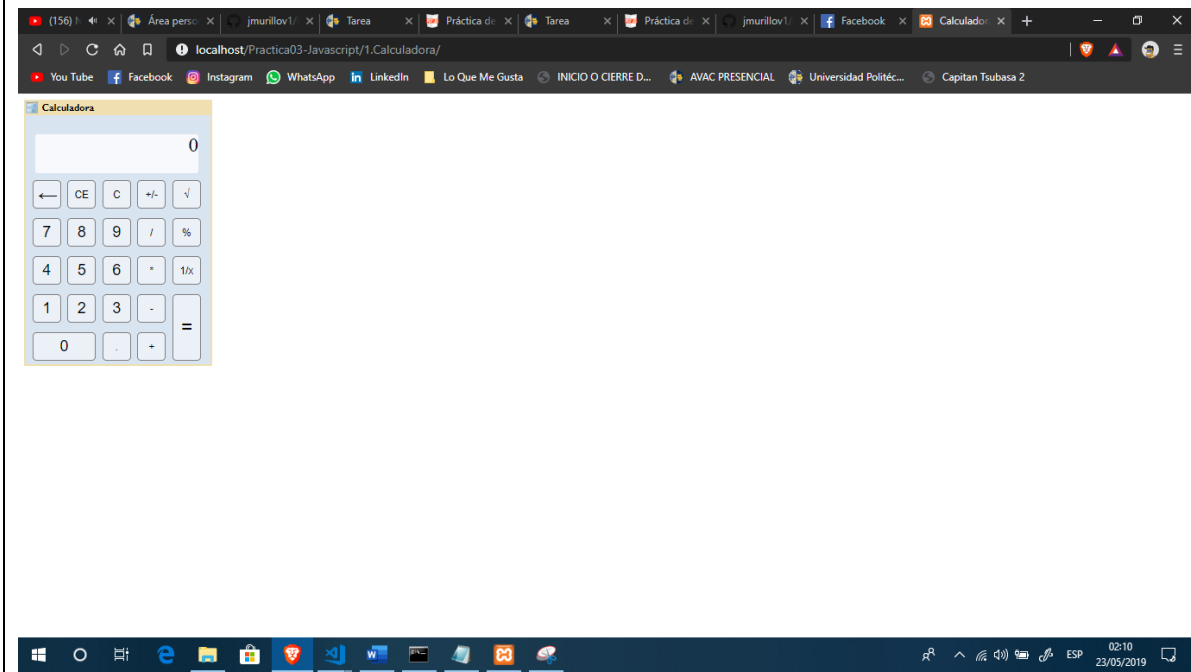
RESULTADO(S) OBTENIDO(S):


- Tener el conocimiento suficiente para que el estudiante pueda entender y organizar de una mejor manera los sitios de web y de negocios en Internet.
- Tener el conocimiento sobre nuevas etiquetas y su uso.
- Aprender a usar HTML5 y CCS3.

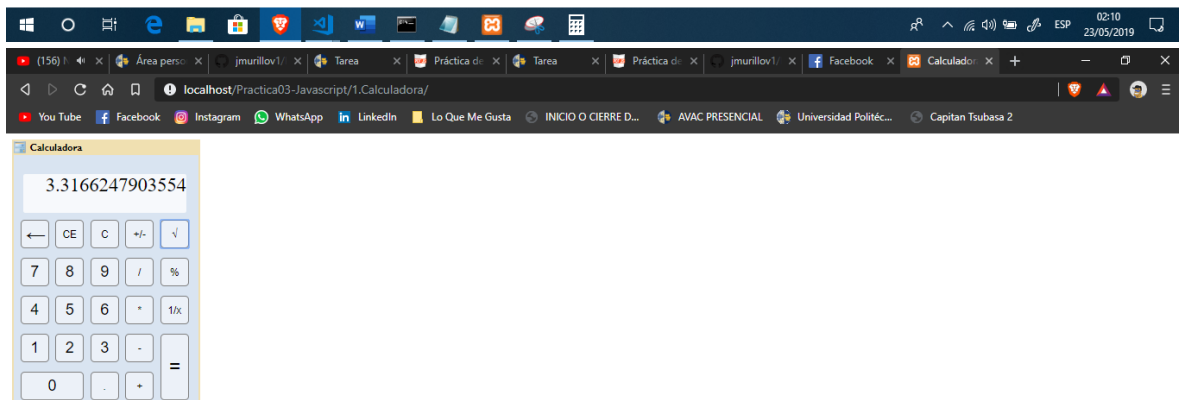
Resultados:


Sitio Web Completo: Google Chrome

1.Calculadora

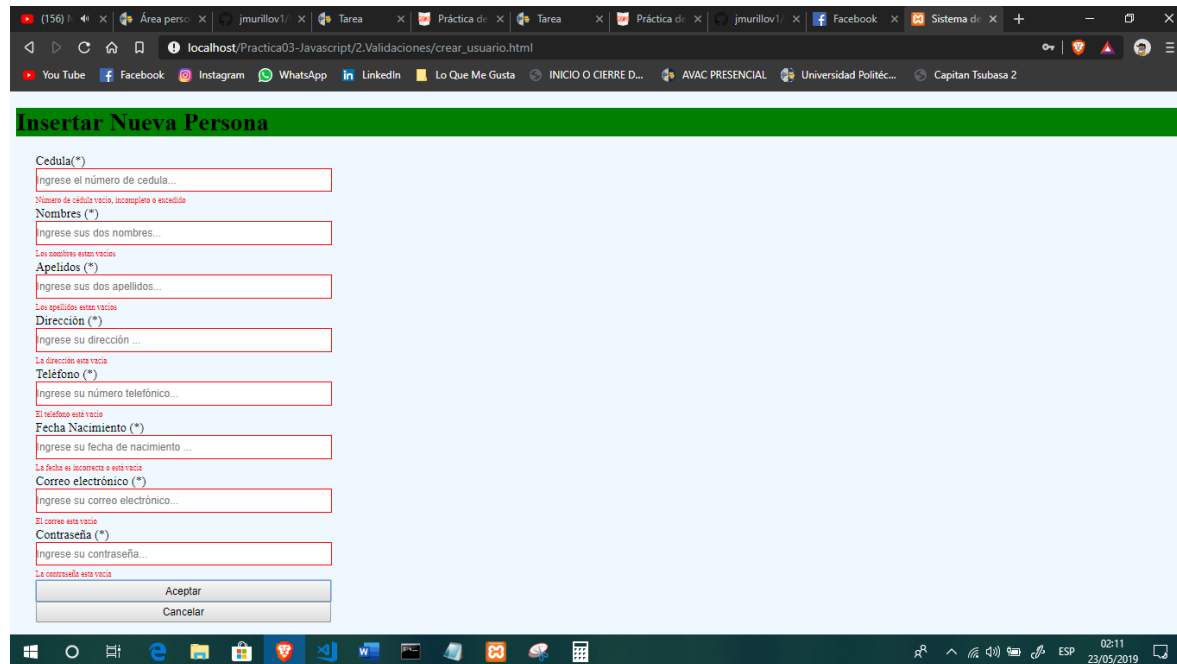


| | | |
|---|------------------------|------------------------|
|  | VICERRECTORADO DOCENTE | Código: GUIA-PRL-001 |
| | CONCEJO ACADÉMICO | Aprobación: 2016/04/06 |
| Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación | | |



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

2. Validaciones



Insertar Nueva Persona

Cedula(*)
Ingrese el número de cedula...

El número de cedula vacío, incompleto o excedido

Nombres (*)
Ingrese sus dos nombres...

Los nombres están vacíos

Apellidos (*)
Ingrese sus dos apellidos...

Los apellidos están vacíos

Dirección (*)
Ingrese su dirección ...

La dirección está vacía

Teléfono (*)
Ingrese su número telefónico...

El teléfono está vacío

Fecha Nacimiento (*)
Ingrese su fecha de nacimiento ...

La fecha es incorrecta o está vacía

Correo electrónico (*)
Ingrese su correo electrónico...

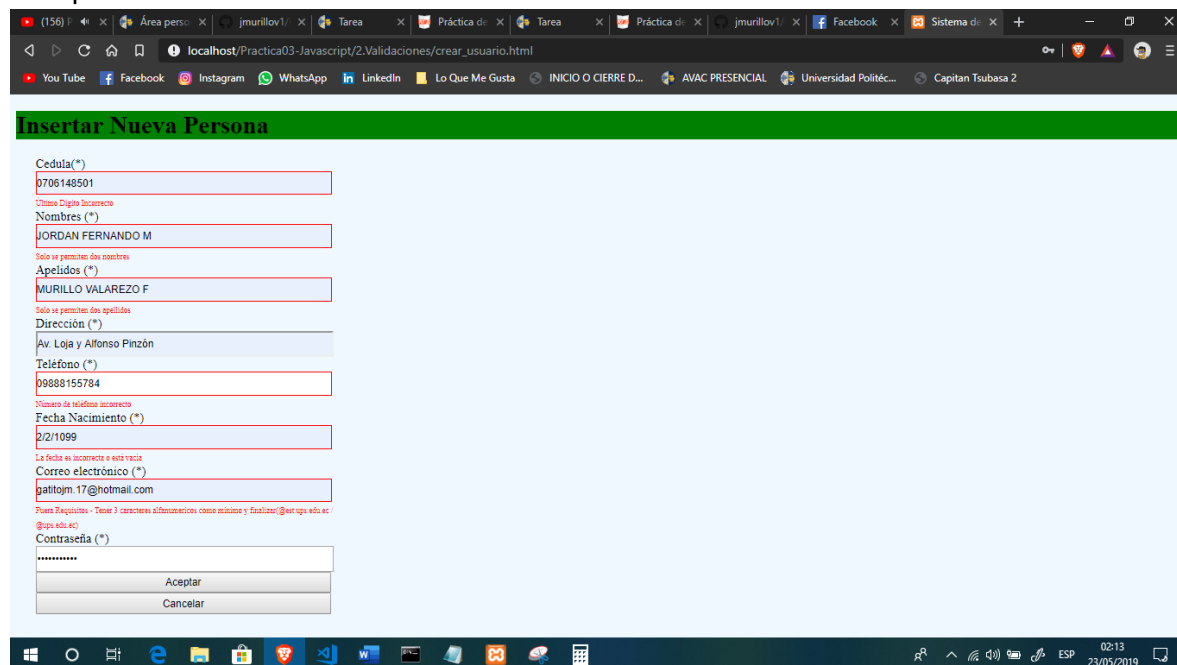
El correo está vacío

Contraseña (*)
Ingrese su contraseña...

La contraseña está vacía

Aceptar
Cancelar

Campos Vacios



Insertar Nueva Persona

Cedula(*)
0706148501

Último Dígito Incorrecto

Nombres (*)
JORDAN FERNANDO M

Solo se permiten dos nombres

Apellidos (*)
MURILLO VALAREZO F

Solo se permiten dos apellidos

Dirección (*)
Av. Loja y Alfonso Pinzón

Teléfono (*)
09888155784

Número de Teléfono incorrecto

Fecha Nacimiento (*)
2/2/1099


La fecha es incorrecta o está vacía

Correo electrónico (*)
gattojm.17@hotmail.com

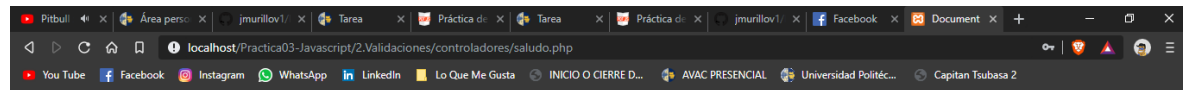
Puede Reemplazar - Tener 3 caracteres alfanuméricos como minimo y finalizar@ext.upa.edu.ec
(@upa.edu.ec)

Contraseña (*)

Aceptar
Cancelar

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

Valores Falsos

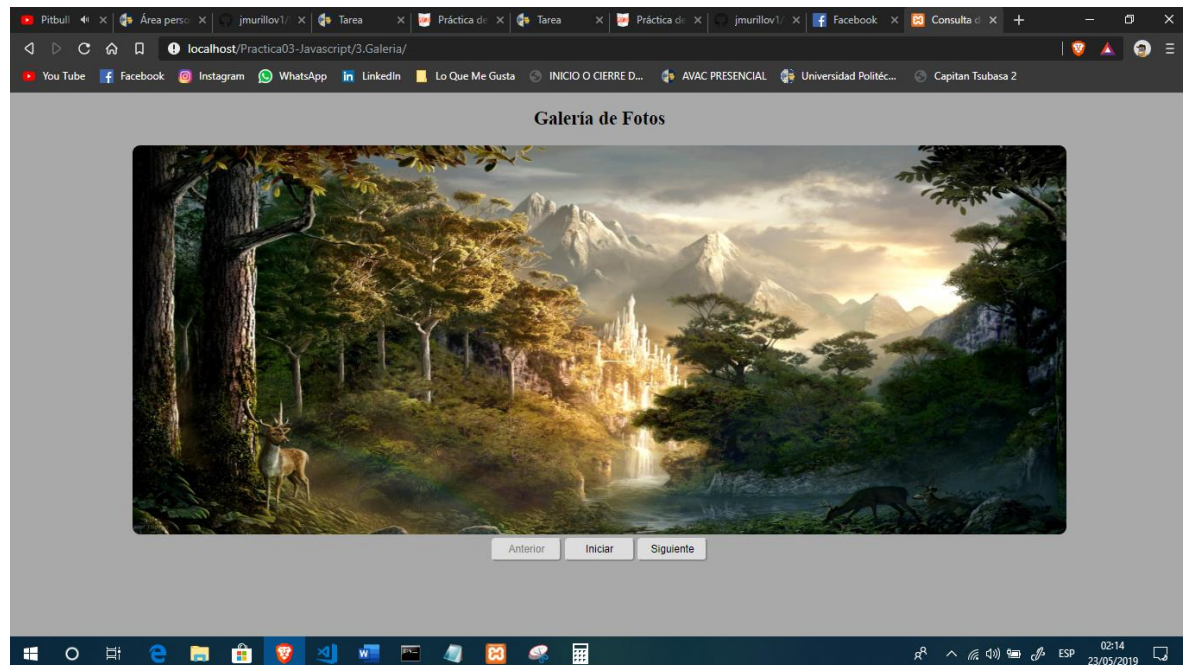


Bienvenido, pasaste las validaciones JORDAN FERNANDO




Todo bien.

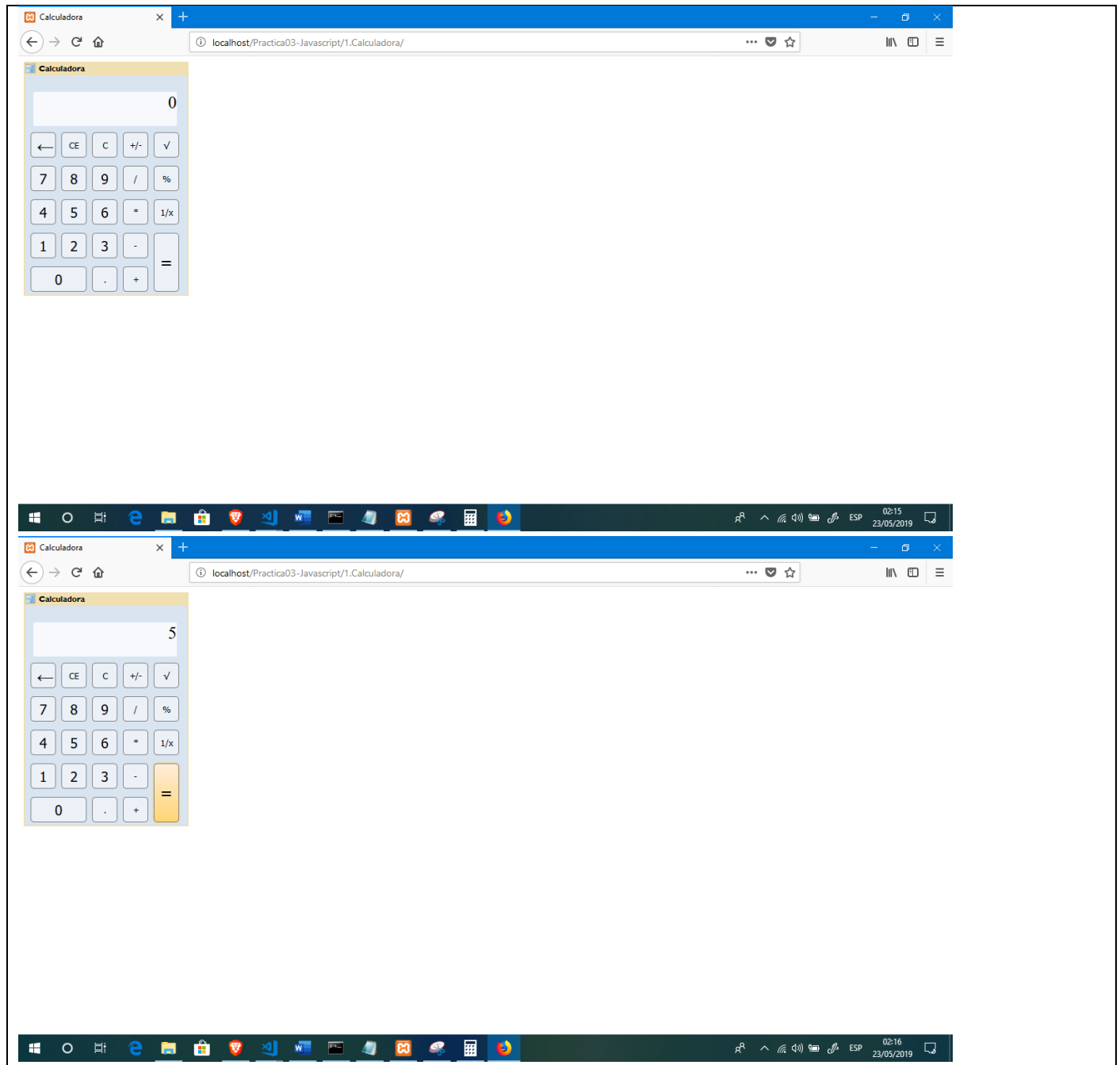
3.Galeria




Sitio Web Completo: Firefox

1. Calculadora

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



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

Calculadora

localhost/Practica03-Javascript/1.Calculadora/

Calculadora

2.23606797749979

←

CE

C

+/-

√

7

8

9

/

%

4

5

6

*

1/x

1

2

3

-

=

0

.

+

2. Validaciones

Sistema de Gestion de Usuarios

localhost/Practica03-Javascript/2.Validaciones/crear_usuario.html

Cedula(*)

Ingrese el número de cedula...

Numero de cédula vacío, incompleto o excedido

Nombres (*)

Ingrese sus dos nombres...

Los nombres estan vacíos

Apellidos (*)

Ingrese sus dos apellidos...

Los apellidos estan vacíos

Dirección (*)

Ingrese su dirección ...

La dirección esta vacía

Teléfono (*)

Ingrese su número telefónico...

El telefono está vacío

Fecha Nacimiento (*)

Ingrese su fecha de nacimiento ...

La fecha es incorrecta o está vacía

Correo electrónico (*)

Ingrese su correo electrónico...

El correo esta vacío


Contraseña (*)

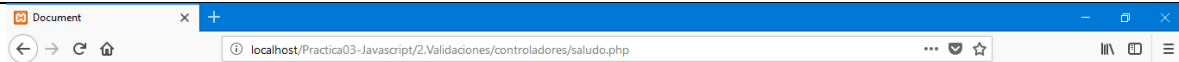
Ingrese su contraseña...

La contraseña esta vacía

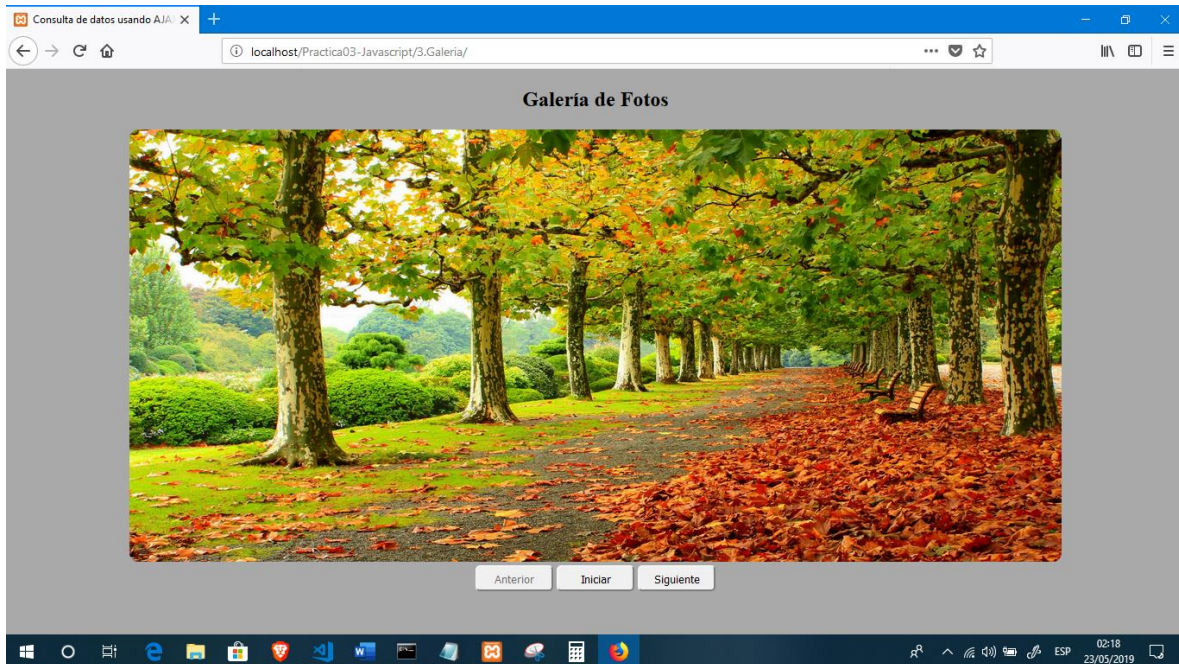
Aceptar


Cancelar

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



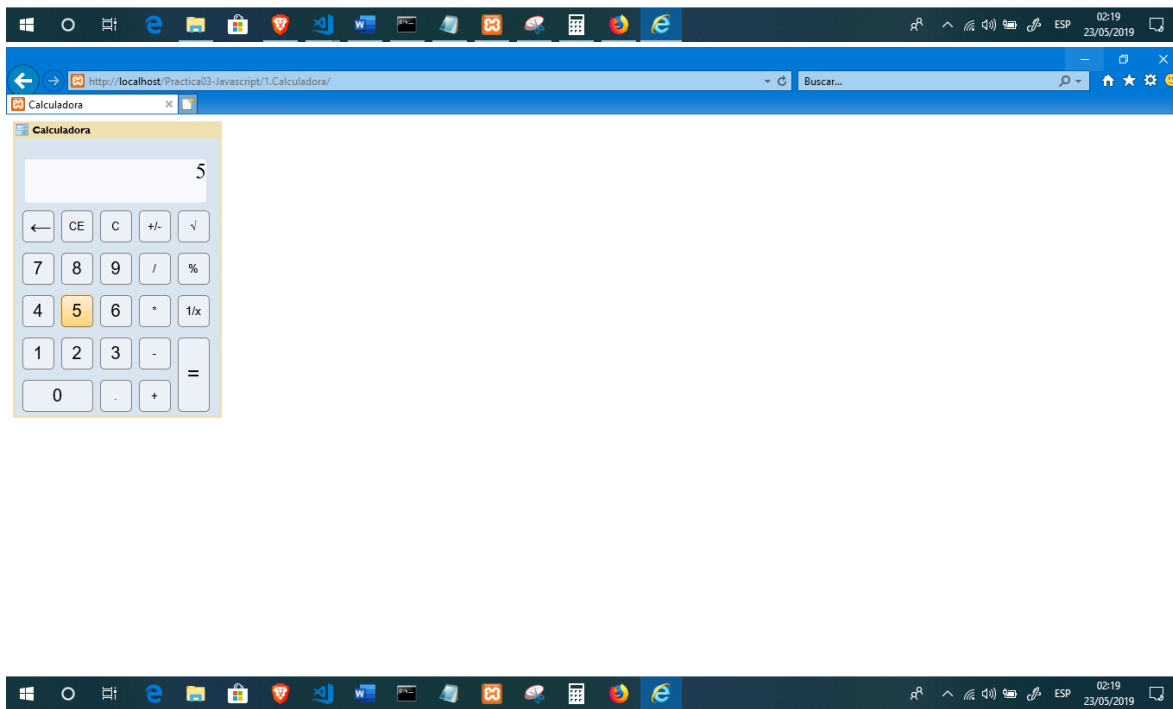
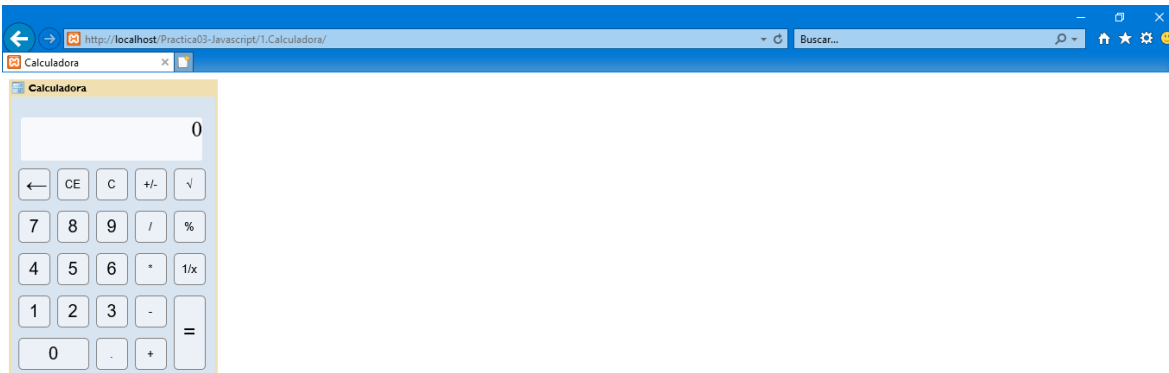
3. Galería




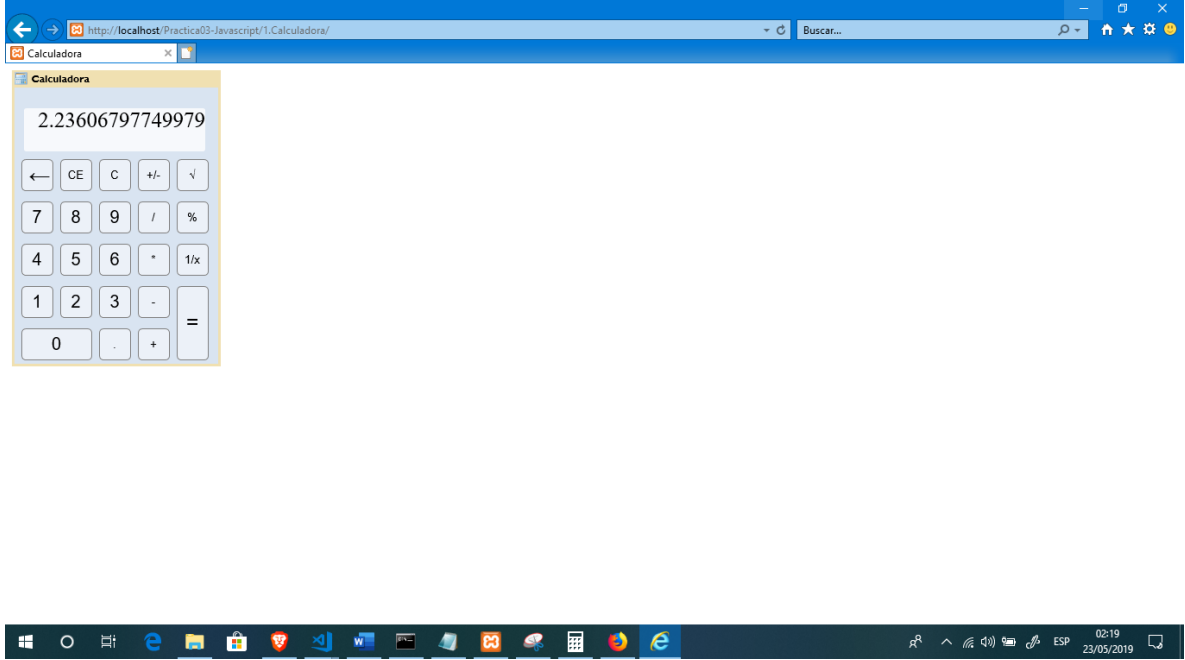
| | | |
|---|------------------------|------------------------|
|  | VICERRECTORADO DOCENTE | Código: GUIA-PRL-001 |
| | CONCEJO ACADÉMICO | Aprobación: 2016/04/06 |
| Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación | | |

Sitio Web Completo: Internet Explorer


1. Calculadora

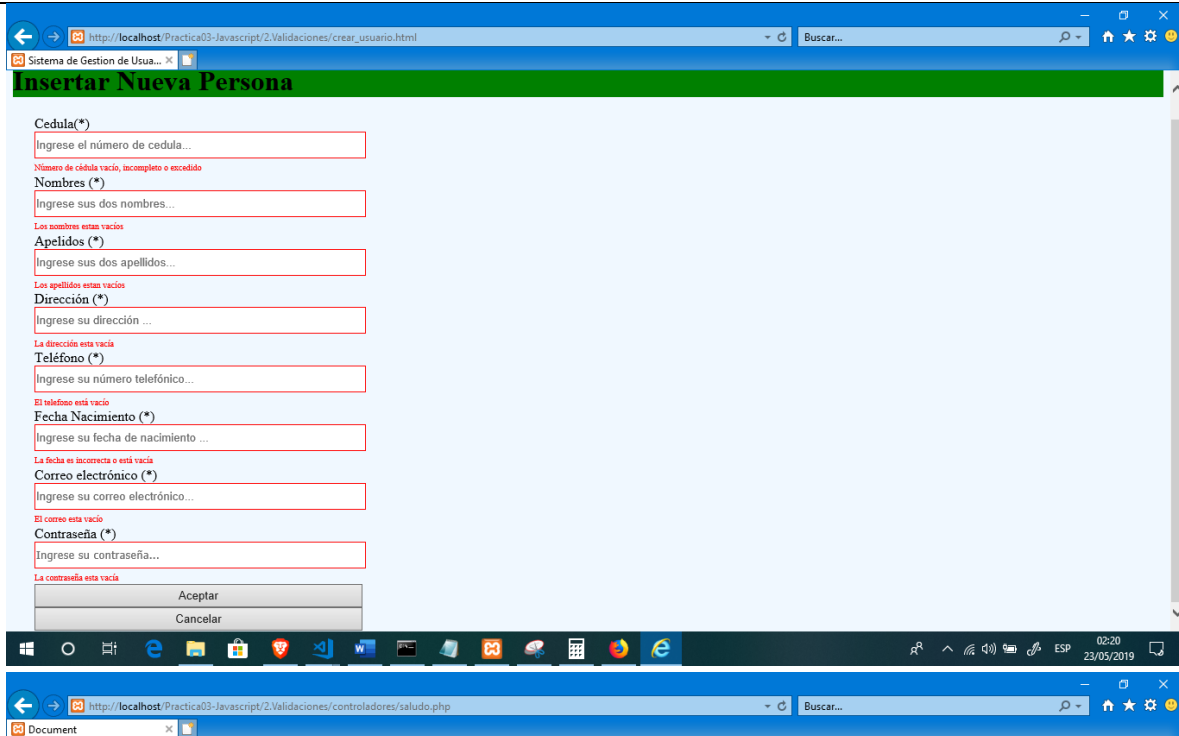


| | | |
|---|------------------------|------------------------|
|  | VICERRECTORADO DOCENTE | Código: GUIA-PRL-001 |
| | CONCEJO ACADÉMICO | Aprobación: 2016/04/06 |
| Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación | | |



2. Validaciones

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



Insertar Nueva Persona

Cedula(*)
Ingrese el número de cedula...
Número de cédula vacío, incompleto o excedido

Nombres (*)
Ingrese sus dos nombres...
Los nombres están vacíos

Apellidos (*)
Ingrese sus dos apellidos...
Los apellidos están vacíos

Dirección (*)
Ingrese su dirección ...
La dirección está vacía

Teléfono (*)
Ingrese su número telefónico...
El teléfono está vacío


Fecha Nacimiento (*)
Ingrese su fecha de nacimiento ...
La fecha es incorrecta o está vacía

Correo electrónico (*)
Ingrese su correo electrónico...
El correo está vacío

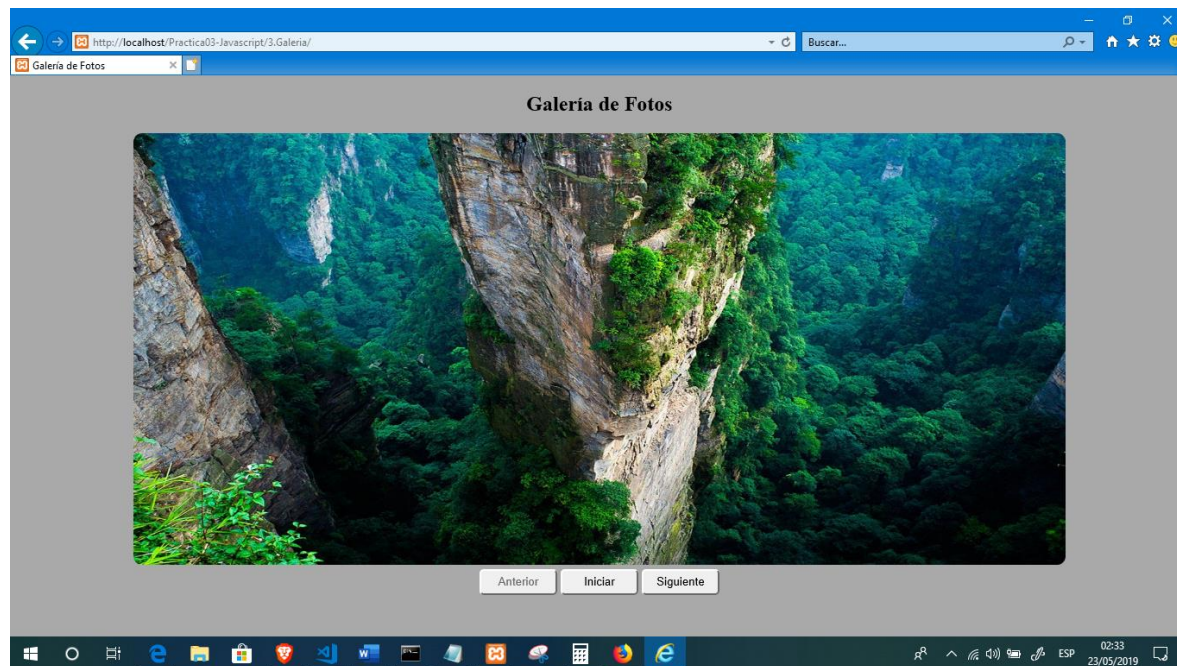
Contraseña (*)
Ingrese su contraseña...
La contraseña está vacía

Aceptar
Cancelar

Bienvenido, pasaste las validaciones Jordan Murillo

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

3. Galería



CONCLUSIONES:

- Los estudiantes podrán organizar sitios web basados en el lenguaje de etiquetado HTML, CSS y JavaScript.
- Que el uso de JavaScript juntamente con HTML y CSS ayuda mucho a desarrollar de mejor forma los sitios web.

RECOMENDACIONES:

- Probar la solución de la práctica en al menos tres navegadores web; Google Chrome, Firefox y Safari.
- Validar las páginas creadas con un software, como es el caso de: W3C Validator.
- Usar correctamente las etiquetas HTML y las formas de CSS y el código para JavaScript.

Nombre del estudiante: Jordan Fernando Murillo Valarezo

Firma del estudiante:

