



FACULTAD DE CIENCIAS DE LA VIDA Y TECNOLOGÍAS

TAREA 3: Formulario de Registro de Clientes

Nombre:

Winter Manuel Povea Bravo

Docente:

Ing. Patricia Alexandra Quiroz Palma

Asignatura:

Aplicaciones Web I

Curso:

5to "A"

Periodo:

2025-2

Contenido

1. Introducción	3
2. Estructura del proyecto	3
3. Desarrollo del HTML	3
3.1. Encabezado del documento.....	4
3.2. Cuerpo del documento	4
3.3. Campos del formulario	5
3.4. Botones de acción.....	6
3.5. Script vinculado	6
4. Diseño con CSS	7
4.1. Estilos generales.....	7
4.2. Contenedor Principal	7
4.3. Sección del Mensaje.....	8
4.4. Formulario	8
4.5. Botones.....	9
4.6. Diseño responsivo	9
5. Lógica con JavaScript	10
5.1. Estructura y funcionamiento.....	10
5.1.1. Inicialización	10
5.1.2. Funciones Auxiliares	10
5.1.3. Limpieza automática de errores.....	11
5.1.4. Validación para enviar el formulario.....	11
5.2. Validaciones implementadas	12
5.2.1. Cédula y teléfono.....	12
5.2.2. Nombres y Apellido	12
5.2.3. Sexo	12
5.2.4. Dirección	13

5.2.5. Fecha de nacimiento	13
5.2.6. Correo Electrónico.....	14
5.3. Envío exitoso	14
6. Pruebas realizadas	15
7. Conclusión	15
8. Link al repositorio en Git Hub	16

Índice de Ilustraciones:

Ilustración 1. Encabezado del HTML	4
Ilustración 2. Contenedor principal en el body	4
Ilustración 3. Formulario.....	5
Ilustración 4. Botones de Acción	6
Ilustración 5. Vinculación con el Script.....	6
Ilustración 6. Estilos generales.....	7
Ilustración 7. Contenedor Principal.....	7
Ilustración 8. Sección del mensaje	8
Ilustración 9. Formulario CSS 2.....	8
Ilustración 10. Botones.....	9
Ilustración 11. Diseño responsivo	9
Ilustración 12. Inicialización JS	10
Ilustración 13. Funciones Auxiliares.....	10
Ilustración 14. Limpieza automática de errores	11
Ilustración 15. Validación para enviar el formulario	11
Ilustración 16. Validación Cédula y Teléfono	12
Ilustración 17. Validación Nombres y Apellidos	12
Ilustración 18. Validación sexo	13
Ilustración 19. Validación Dirección	13
Ilustración 20. Validación fecha de nacimiento	13
Ilustración 21. Validación Correo Electrónico.....	14
Ilustración 22. Envío exitoso.....	14
Ilustración 23. Comprobando validaciones	15

1. Introducción

El presente documento describe el desarrollo de un formulario web interactivo para el registro de clientes del banco SHARK, una entidad financiera de estilo premium. El objetivo principal de este proyecto es implementar una interfaz moderna y funcional que permita al usuario ingresar sus datos personales de forma clara y segura, mediante un sistema de validaciones que garantice la correcta captura de la información.

El formulario ha sido diseñado con un enfoque estético elegante, utilizando una paleta de colores oscuros con detalles dorados, y una estructura modular compuesta por archivos HTML, CSS y JavaScript, lo que facilita su mantenimiento y escalabilidad.

2. Estructura del proyecto

El proyecto se realizó de manera modular es decir se separaron los tipos de archivo según su función en el proyecto de manera que:

- **Formulario.html:** contiene la estructura principal del formulario.
- **Estilo.css:** define los estilos visuales, colores y disposición de los elementos.
- **Validación.js:** controla las validaciones de los campos y la interacción del formulario.

Representación gráfica:

Formulario con validaciones

|--- Estilo.css

|---Formulario.html

|---sl_030220_28570_39.jpg

|---Validación.js

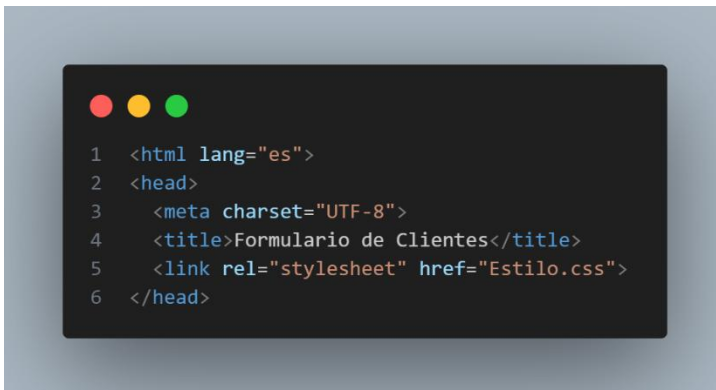
3. Desarrollo del HTML

El archivo Formulario.html define la estructura principal del formulario de registro de clientes para SHARK BANK. Este documento está dividido en dos secciones principales: un panel informativo y el formulario de registro.

3.1. Encabezado del documento

En la sección `<head>` se define el idioma, la codificación y se enlaza la hoja de estilos externa:

Ilustración 1. Encabezado del HTML

A screenshot of a code editor with a dark background and light-colored text. The code is HTML markup for the head section of a document. It includes the language attribute, the head tag, a meta tag for charset, a title tag, and a link tag for an external stylesheet.

```
1 <html lang="es">
2 <head>
3   <meta charset="UTF-8">
4   <title>Formulario de Clientes</title>
5   <link rel="stylesheet" href="Estilo.css">
6 </head>
```

3.2. Cuerpo del documento

Dentro del `<body>` se encuentra un contenedor principal que divide la página en dos secciones:

Ilustración 2. Contenedor principal en el body

A screenshot of a code editor with a dark background and light-colored text. The code is HTML markup for the body section of a document. It includes the body tag, a container div, a comment for the left side, a message div with a welcome message and a paragraph, a comment for the right side, and a form tag.

```
1 <body>
2   <div class="contenedor">
3     <!-- Lado izquierdo -->
4     <div class="mensaje">
5       <h1>¡Bienvenido a <strong>SHARK BANK</strong>! </h1>
6       <p>Por favor, complete todos los campos con sus datos personales y acceda
7         a la comunidad bancaria más <strong>EXCLUSIVA</strong>.
8     </p>
9     </div>
10
11    <!-- Lado derecho -->
12    <form id="formulario">
```

- Lado izquierdo ‘.mensaje’: contiene un mensaje de bienvenida con el nombre del banco.
- Lado derecho (`<form>`): contiene todos los campos del formulario de registro.

3.3. Campos del formulario

El formulario incluye los siguientes campos para la recolección de datos:

- Cédula
- Nombres
- Apellidos
- Sexo (mediante botones de radio)
- Dirección
- Fecha de nacimiento
- Correo electrónico
- Teléfono

Cada campo tiene una etiqueta (<label>) asociada y un elemento <div class="error"> para mostrar mensajes de validación.

Ilustración 3. Formulario

```

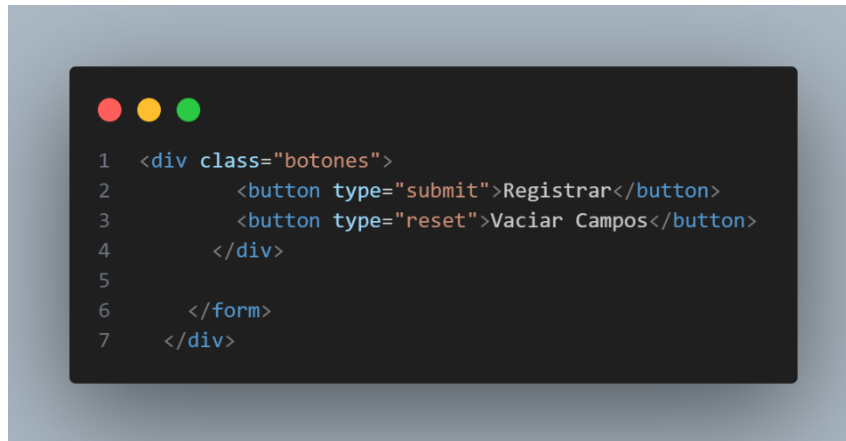
1  <!-- Lado derecho -->
2  <form id="formulario">
3    <h2>Registro Clientes</h2>
4
5    <label for="cedula">Cédula:</label>
6    <input type="text" id="cedula" name="cedula" maxlength="10">
7    <div class="error" id="errorCedula"></div>
8
9    <label for="nombres">Nombres:</label>
10   <input type="text" id="nombres" name="nombres">
11   <div class="error" id="errorNombres"></div>
12
13   <label for="apellidos">Apellidos:</label>
14   <input type="text" id="apellidos" name="apellidos">
15   <div class="error" id="errorApellidos"></div>
16
17   <label>Sexo:</label>
18   <div class="sexo-opciones">
19     <input type="radio" id="masculino" name="sexo" value="Masculino">
20     <label for="masculino">Masculino</label>
21
22     <input type="radio" id="femenino" name="sexo" value="Femenino">
23     <label for="femenino">Femenino</label>
24   </div>
25   <div class="error" id="errorSexo"></div>
26
27   <label for="direccion">Dirección:</label>
28   <input type="text" id="direccion" name="direccion" maxlength="100">
29   <div class="error" id="errorDireccion"></div>
30
31   <label for="fechaNacimiento">Fecha de nacimiento:</label>
32   <input type="date" id="fechaNacimiento" name="fechaNacimiento">
33   <div class="error" id="errorFechaNacimiento"></div>
34
35   <label for="correo">Correo:</label>
36   <input type="email" id="correo" name="correo">
37   <div class="error" id="errorCorreo"></div>
38
39   <label for="telefono">Teléfono:</label>
40   <input type="text" id="telefono" name="telefono" maxlength="10">
41   <div class="error" id="errorTelefono"></div>

```

3.4. Botones de acción

- Registrar: envía el formulario para su validación mediante JavaScript.
- Vaciar Campos: limpia todos los campos del formulario.

Ilustración 4. Botones de Acción



3.5. Script vinculado

Finalmente, se incluye el archivo Validacion.js, el cual contiene la lógica de validación de los datos ingresados:

Ilustración 5. Vinculación con el Script

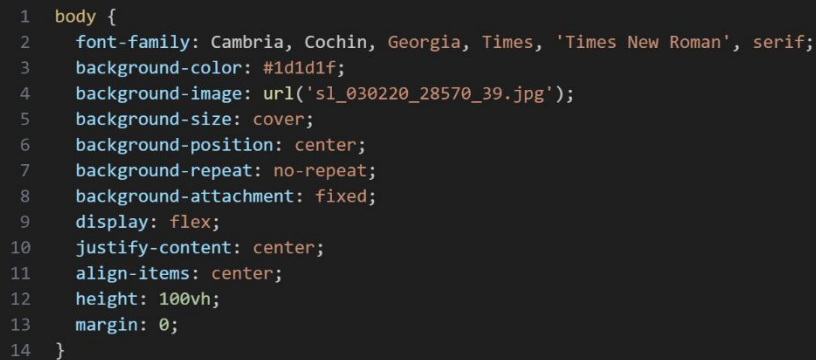


4. Diseño con CSS

4.1. Estilos generales

Se define una tipografía elegante (familia Cambria), un fondo oscuro con imagen centrada, y un diseño de tipo ‘flexbox’ para centrar el contenido tanto vertical como horizontalmente.

Ilustración 6. Estilos generales

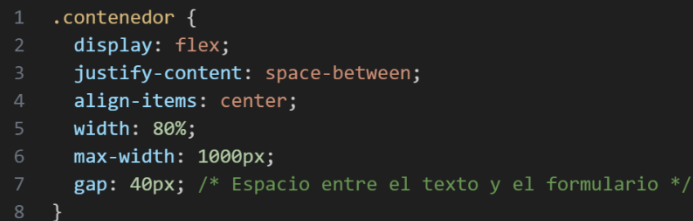
A screenshot of a code editor with a dark background and light-colored text. The code is CSS for the body element, using flexbox for centering and a dark background with a centered image.

```
1  body {  
2    font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;  
3    background-color: #1d1d1f;  
4    background-image: url('sl_030220_28570_39.jpg');  
5    background-size: cover;  
6    background-position: center;  
7    background-repeat: no-repeat;  
8    background-attachment: fixed;  
9    display: flex;  
10   justify-content: center;  
11   align-items: center;  
12   height: 100vh;  
13   margin: 0;  
14 }
```

4.2. Contenedor Principal

El contenedor ‘.contenedor’ divide la pantalla en dos secciones: el mensaje de bienvenida (izquierda) y el formulario (derecha).

Ilustración 7. Contenedor Principal

A screenshot of a code editor with a dark background and light-colored text. The code is CSS for the .contenedor class, using flexbox to divide the screen into two sections.

```
1  .contenedor {  
2    display: flex;  
3    justify-content: space-between;  
4    align-items: center;  
5    width: 80%;  
6    max-width: 1000px;  
7    gap: 40px; /* Espacio entre el texto y el formulario */  
8  }
```


4.3. Sección del Mensaje

La clase ‘.mensaje’ define el área de texto promocional, usando color blanco y espaciado cómodo para la lectura.

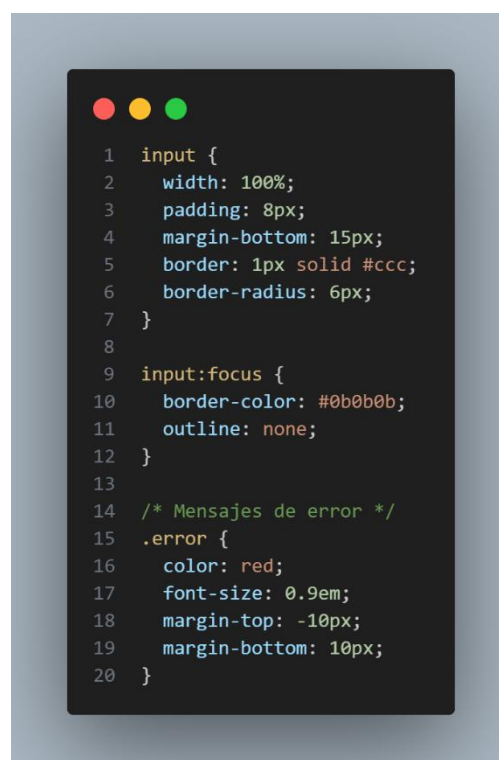
Ilustración 8. Sección del mensaje



4.4. Formulario

El formulario posee fondo claro, bordes redondeados y sombra para resaltar sobre el fondo oscuro. Cada campo (label + input) está correctamente espaciado y los errores se muestran en rojo mediante ‘.error’.

Ilustración 9. Formulario CSS 2

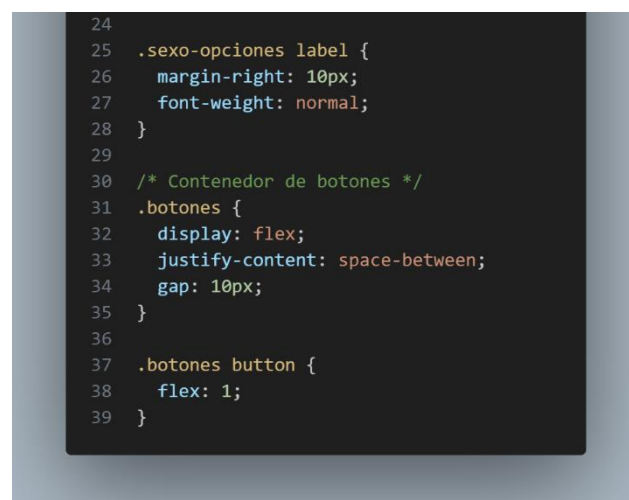


4.5. Botones

Se aplican estilos consistentes a los botones:

- Color de fondo negro con texto blanco.
- Efecto ‘hover’ que cambia el color a dorado (#c9bf0f).
- Diseño adaptado con ‘.botones’ para mantener ambos botones (Registrar / Vaciar Campos) del mismo tamaño.

Ilustración 10. Botones



4.6. Diseño responsivo

Mediante una regla ‘@media’ se asegura que el diseño se adapte en pantallas pequeñas, apilando el texto y el formulario verticalmente para mejorar la legibilidad en dispositivos móviles.

Ilustración 11. Diseño responsivo



5. Lógica con JavaScript

El archivo Validacion.js contiene la lógica de validación del formulario de registro de clientes del SHARK BANK, desarrollada en JavaScript puro (sin librerías externas). Su función principal es garantizar que los datos ingresados por el usuario cumplan con los formatos establecidos antes de permitir el envío del formulario.

El script se ejecuta cuando el contenido del documento HTML ha cargado completamente gracias al evento ‘DOMContentLoaded’.

5.1. Estructura y funcionamiento

5.1.1. Inicialización

- Se asegura de que el código solo se ejecute una vez que el formulario esté disponible en el ‘DOM’.
- Si el elemento no existe, el script se detiene.

Ilustración 12. Inicialización JS

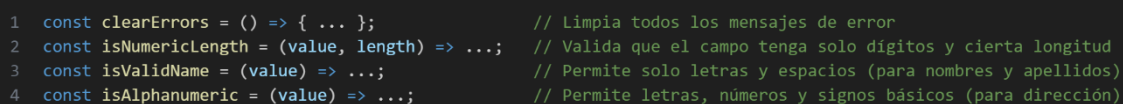


```
1 document.addEventListener('DOMContentLoaded', function() {
2   const form = document.getElementById('formulario');
3   if (!form) return;
```

5.1.2. Funciones Auxiliares

El código incluye tres funciones reutilizables que simplifican las validaciones:

Ilustración 13. Funciones Auxiliares



```
1 const clearErrors = () => { ... }; // Limpia todos los mensajes de error
2 const isNumericLength = (value, length) => ...; // Valida que el campo tenga solo dígitos y cierta longitud
3 const isValidName = (value) => ...; // Permite solo letras y espacios (para nombres y apellidos)
4 const isAlphanumeric = (value) => ...; // Permite letras, números y signos básicos (para dirección)
```

Estas funciones mejoran la legibilidad y evitan repetir código en cada validación.

5.1.3. Limpieza automática de errores

- Al presionar el botón "Vaciar Campos", se eliminan los mensajes de error del formulario.
- Se usa un 'setTimeout' corto para permitir que el evento 'reset' se ejecute primero.

Ilustración 14. Limpieza automática de errores

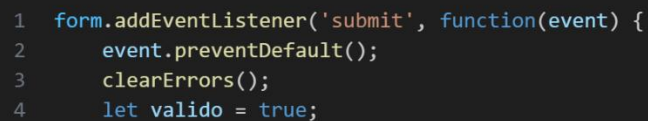


```
1 form.addEventListener('reset', () => setTimeout(clearErrors, 0));
```

5.1.4. Validación para enviar el formulario

- Se evita el envío automático del formulario (preventDefault).
- Se limpia cualquier error previo.
- Se declara la variable 'valido' como bandera general:
 - true → formulario correcto
 - false → existen errores que impiden el envío.

Ilustración 15. Validación para enviar el formulario



```
1 form.addEventListener('submit', function(event) {  
2   event.preventDefault();  
3   clearErrors();  
4   let valido = true;
```

5.2. Validaciones implementadas

5.2.1. Cédula y teléfono

Verifica que la cédula y el numero de teléfono tenga exactamente 10 dígitos numéricos.

Ilustración 16. Validación Cédula y Teléfono

```
1 // Validar cédula
2 let cedula = document.getElementById('cedula').value.trim();
3 if (!isNumericLength(cedula, 10)) {
4   document.getElementById('errorCedula').textContent = 'Ingrese una cédula válida (10 números).';
5   valido = false;
6 }
7
8 // Validar teléfono
9 let telefono = document.getElementById('telefono').value.trim();
10 if (!isNumericLength(telefono, 10)) {
11   document.getElementById('errorTelefono').textContent = 'Ingrese un teléfono válido (10 números).';
12   valido = false;
13 }
```

5.2.2. Nombres y Apellido

Acepta únicamente letras (incluyendo tildes y ñ) y espacios.

Ilustración 17. Validación Nombres y Apellidos

```
1 // Validar nombres
2 let nombres = document.getElementById('nombres').value.trim();
3 if (!isValidName(nombres)) {
4   document.getElementById('errorNombres').textContent = 'Solo se permiten letras y espacios.';
5   valido = false;
6 }
7
8 // Validar apellidos
9 let apellidos = document.getElementById('apellidos').value.trim();
10 if (!isValidName(apellidos)) {
11   document.getElementById('errorApellidos').textContent = 'Solo se permiten letras y espacios.';
12   valido = false;
13 }
```

5.2.3. Sexo

Comprueba que el usuario haya seleccionado una de las opciones (Masculino o Femenino).

Ilustración 18. Validación sexo

```

1  // Validar sexo
2  let sexo = document.querySelector('input[name="sexo"]:checked');
3  if (!sexo) {
4      document.getElementById('errorSexo').textContent = 'Seleccione su sexo.';
5      valido = false;
6  }

```

5.2.4. Dirección

Valida que el campo no esté vacío y solo contenga caracteres permitidos (letras, números, punto, coma, numeral o guion).

Ilustración 19. Validación Dirección

```

1  // Validar dirección (alfanumérica)
2  let direccion = document.getElementById('direccion').value.trim();
3  if (direccion === '' || !isAlphanumeric(direccion)) {
4      document.getElementById('errorDireccion').textContent = 'Ingrese una dirección válida (solo letras, números y signos básicos).';
5      valido = false;
6  }

```

5.2.5. Fecha de nacimiento

Calcula la edad exacta y exige que el usuario tenga al menos 18 años para registrarse.

Ilustración 20. Validación fecha de nacimiento

```

1  // Validar fecha de nacimiento (mayor de 18 años)
2  let fechaNacimiento = document.getElementById('fechaNacimiento').value;
3  if (!fechaNacimiento) {
4      document.getElementById('errorFechaNacimiento').textContent = 'Seleccione su fecha de nacimiento.';
5      valido = false;
6  } else {
7      let hoy = new Date();
8      let fechaNac = new Date(fechaNacimiento);
9      let edad = hoy.getFullYear() - fechaNac.getFullYear();
10     let m = hoy.getMonth() - fechaNac.getMonth();
11     if (m < 0 || (m === 0 && hoy.getDate() < fechaNac.getDate())) edad--;
12     if (edad < 18) {
13         document.getElementById('errorFechaNacimiento').textContent = 'Debe tener al menos 18 años.';
14         valido = false;
15     }
16 }

```

5.2.6. Correo Electrónico

Verifica el formato de correo. Solo acepta los dominios más comunes y el institucional: gmail.com, outlook.com, hotmail.com, live.com, yahoo.com, live.uleam.edu.ec.

Ilustración 21. Validación Correo Electrónico

```
1 // Validar correo
2 let correo = document.getElementById('correo').value.trim();
3 if (!/^[^\s@]+@(gmail\.com|outlook\.com|hotmail\.com|live\.com|yahoo\.com|live\.uleam\.edu\.ec)$/i.test(correo)) {
4     document.getElementById('errorCorreo').textContent =
5     'Ingrese un correo válido (gmail, outlook, hotmail, live, yahoo o live.uleam.edu.ec).';
6     valido = false;
7 }
```

5.3. Envío exitoso

Si todas las validaciones son correctas, se muestra un mensaje de confirmación. Luego se limpia el formulario automáticamente.

Ilustración 22. Envío exitoso

```
1 if (valido) {
2     alert('Formulario enviado correctamente.');
```

```
3     form.reset();
```

6. Pruebas realizadas

Después de tener habilitadas todas las validaciones las cuales se fueron modificando a la marcha, nos da como resultado lo siguiente:

Si los campos están vacíos o tienen datos que no corresponden aparece el mensaje rojo debajo de los mismos:

Ilustración 23. Comprobando validaciones

The image shows a registration form for SHARK BANK. On the left, a dark banner with gold diagonal stripes and a starry background contains the text: **¡Bienvenido a SHARK BANK!** and *Por favor, complete todos los campos con sus datos personales y acceda a la comunidad bancaria más EXCLUSIVA.*

On the right, the registration form fields and their validation messages are as follows:

- Cédula:** 1317059960
- Nombres:** Winter Manuel
- Apellidos:** Povea 13214
Solo se permiten letras y espacios.
- Sexo:** ☐ Masculino ☐ Femenino
Seleccione su sexo.
- Dirección:** Barrio Abdón Calderón
- Fecha de nacimiento:** 25/10/2022
Debe tener al menos 18 años.
- Correo:** bravopwinter@correo.com
Ingrese un correo válido (gmail, outlook, hotmail, live, yahoo o live.uileam.edu.ec).
- Teléfono:** 09395309as
Ingrese un teléfono válido (10 números).

Observando detalladamente la imagen, nos damos cuenta de los mensajes de error mientras que en los campos en los que hay información correcta simplemente se mantienen con los datos.

Así mismo se ofrece la posibilidad de vaciar todos los campos junto con sus errores o por su contraparte el uso del botón ‘Registrar’ una vez los campos sean aptos por su validación.

7. Conclusión

La realización de esta actividad logró su cometido y es que no simplemente fue lograr un formulario con validaciones, sino que, todo en su conjunto contempla; el diseño de paginas web, la temática que se le puede brindar a una de estas páginas y los tipos de datos, que en un aspecto profesional merecen de validaciones más exactas.

8. Link al repositorio en Git Hub

[https://github.com/WinterPovea/Aplicaciones-Web-
I/tree/2dec76f620530409e0b728e9ff4db4c7dd7a3065/Formulario%20con%20validacio
nes](https://github.com/WinterPovea/Aplicaciones-Web-
I/tree/2dec76f620530409e0b728e9ff4db4c7dd7a3065/Formulario%20con%20validacio
nes)