



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE TLAXIACO

SEGURIDAD Y VIRTUALIZACIÓN

Proyecto final

Integrantes del Equipo:

Arnol Jesus Cruz Ortiz

Amilkar Vladimir Reyes Reyes

Rael Gabriel Bautista

Sandra Gabriela Velasco Guzmán

Docente:

Edward Osorio Salinas

Carrera:

Ingeniera en Sistemas Computacionales

Grupo: 7US

Semestre: Agosto – diciembre 2024

06/Diciembre/2024

PROYECTO FINAL

=VALIDACIÓN Y CREACIÓN DE CONTRASEÑAS SEGURAS=

ÍNDICE

OBJETIVO	3
INTRODUCCIÓN	3
“INICIO DEL PROYECTO”	4
Creación del proyecto final en Visual estudio code.....	4
Creación de la pagina principal	5
Creación de la hoja de estilos CSS	6
Creación de código para la creación de contraseñas seguras	8
Creación del código para la verificación de contraseñas.....	10
Creación de la validación de los formularios	12
Creación de código para los estilos de las tablas y formularios	14
Estilos globales.....	15
Estilos para los títulos.....	15
Formulario y tabla	16
Inputs y Botones	17
En la parte de la contraseña	18
Botones adicionales.....	19
Interfaz del inicio	20
Interfaz del Generador de contraseñas seguras	20
Interfaz de Verificación de Contraseña	21
PRUEBAS DE LA PAGINA WEB	21
CONCLUSIÓN	26

OBJETIVO

El objetivo de este proyecto es desarrollar una aplicación web segura denominada ProyectoSeguridad, que integre las funcionalidades de una generación de contraseñas y verificación de contraseñas utilizando buenas prácticas de seguridad como cifrado de contraseñas, autenticación y autorización. Este proyecto tiene como propósito crear un sistema confiable para la creación de contraseñas que pueda ser usado como base para aplicaciones más complejas en el futuro.

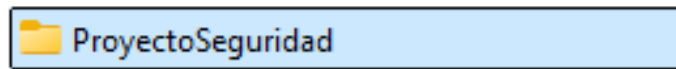
INTRODUCCIÓN

La seguridad en las aplicaciones web es un aspecto crítico en el desarrollo de software moderno, especialmente con el creciente número de ataques a datos o contraseñas sensibles. En este proyecto se propone el diseño y desarrollo de ProyectoSeguridad, una aplicación web basada en tecnologías frontend como HTML, CSS y JavaScript, junto con un backend en Node.js. El sistema permitirá crear contraseñas seguras con nuestros datos personales, y también poder validar esa contraseña si es segura o no, Además, se aplicarán técnicas de cifrado para proteger contraseñas y garantizar la seguridad de la información de contraseñas. Este proyecto no solo busca proporcionar un entorno seguro, sino también un diseño modular y extensible para futuras implementaciones.

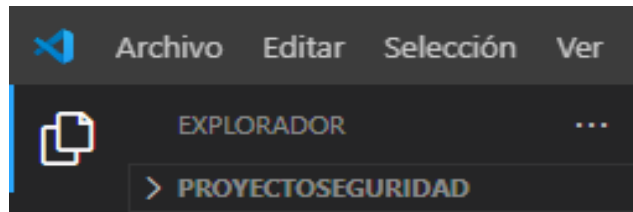
“INICIO DEL PROYECTO”

Creación del proyecto final en Visual estudio code

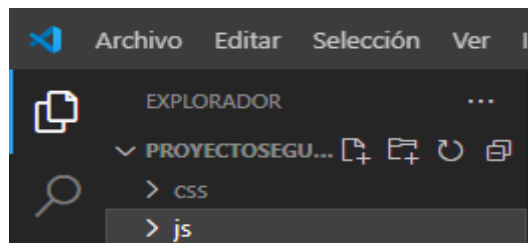
Para la creación de nuestro proyecto nos dirigimos al almacenamiento de nuestro equipo y ahí buscamos en este orden Windows/ Xampp/ htdocs y ahí creamos una carpeta con el nombre ProyectoSeguridad.



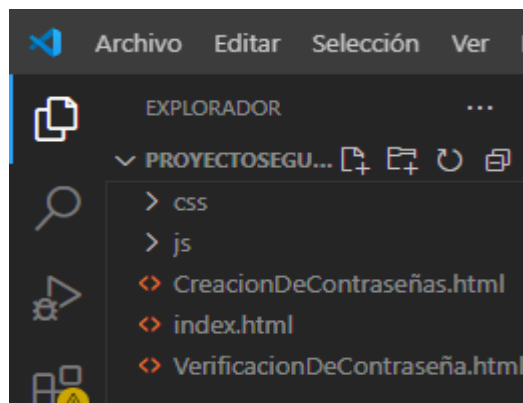
Ahora para abrir esa carpeta en visual code solo arrastramos la carpeta a una pestaña de Visual.



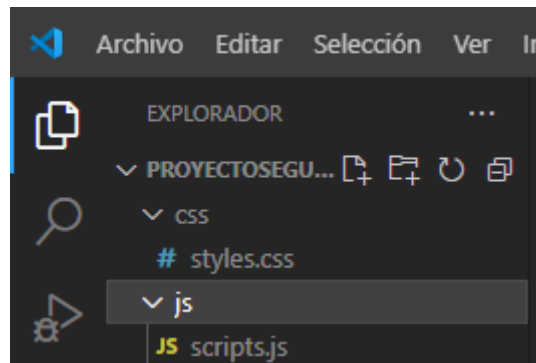
Ahora en la carepta de ProyectoSeguridad creamos dos carpetas una para css que será para los estilos de nuestra pagina web y js que será donde validará y reconocerá las contraseñas seguras.



Ahora crearemos los archivos Creación de contraseñas, Verificación de Contraseñas y un index todos con .html.



Ahora dentro de la carpeta css voy a crear un archivo llamado styles.css y también en la carpeta js creamos un archivo llamado scripts.js.



Con esta estructura que creamos será suficiente para realizar nuestra página web.

Una vez creado nuestra estructura de trabajo vamos a crear la interfaz (Frontend) Primeramente se creará el index.html que será la página principal. Después crearemos el archivo Creación de contraseñas que será el formulario en donde crearemos contraseñas, también crearemos el archivo verificación de contraseña en donde solo va a verificar si la contraseña es segura o no.



Creación de la página principal

Ahora vamos a empezar a crear el código de la página principal llamada index.html. La estructura de este código decidimos que tuviera un encabezado en donde muestre el propósito del proyecto, la navegación que proporciona enlaces a dos funcionalidades, un enlace para creación de contraseñas seguras y otro enlace de verificación de contraseñas seguras.

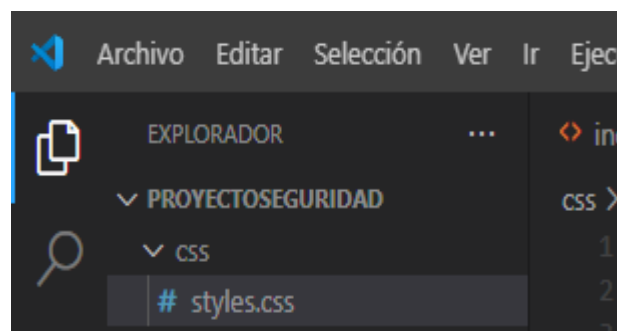
```
<> index.html X
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>ProyectoSeguridad</title>
7      <link rel="stylesheet" href="css/styles.css">
8  </head>
9  <body>
10
11      <h1>Validacion y creacion de contraseñas</h1>
12      <nav>
```

Ahora en esta sesión vamos a crear una sección de navegación con `<nav>` y después con `<a>` definiremos los enlaces que permiten al usuario navegar entre páginas. Ahora con `href="CreacionDeContraseñas.html"` indica que el enlace redirige al archivo `CreacionDeContraseñas.html`. y `href="VerificacionDeContraseña.html"` indica que el enlace redirige al archivo `VerificacionDeContraseña.html`.

```
13      <a href="CreacionDeContraseñas.html">Creacion De Contraseñas seguras</a>
14
15      <a href="VerificacionDeContraseña.html">Verificación de contraseñas seguras</a>
16  </nav>
17 </body>
18 </html>
```

Creación de la hoja de estilos CSS

Ahora en la carpeta `css` que creamos vamos a crear un nuevo archivo llamado `styles.css`

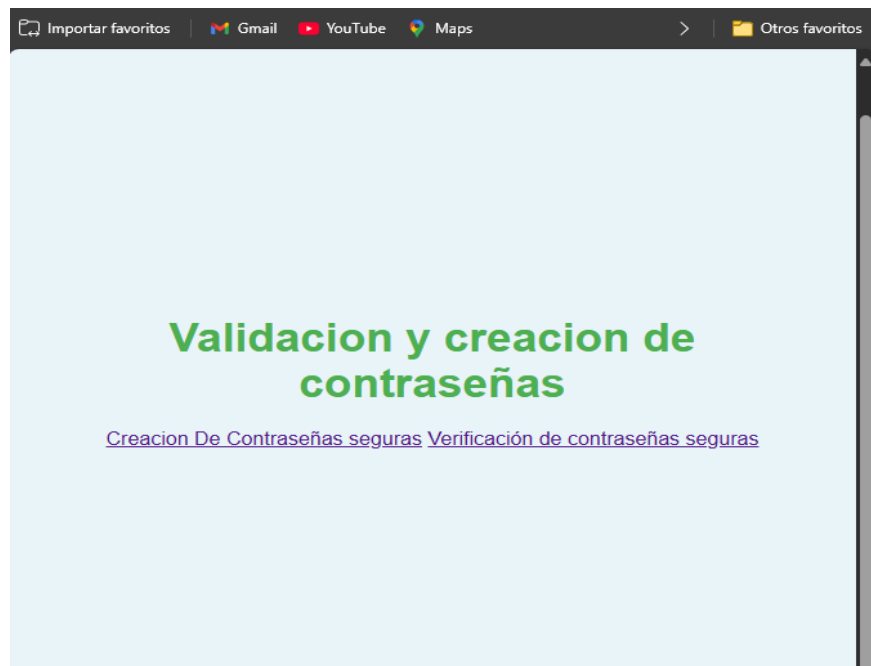


Y creamos el código para darle estilo a la pagina de inicio y queda de la siguiente manera.

```
<> index.html # styles.css X
css > # styles.css > body
1  /* Estilos globales */
2  body {
3      font-family: 'Arial', sans-serif;
4      background-color: #e8f4f8;
5      color: #333;
6      padding: 20px;
7      margin: 0;
8      display: flex;
9      justify-content: center;
10     align-items: center;
11     height: 100vh;
12     flex-direction: column;
13 }
14
15 /* Títulos */
16 h1 {
17     font-size: 2.2em; /* Aumenta el tamaño del título principal */
18     color: #4CAF50;
19     margin-bottom: 20px;
20     text-align: center;
21 }
22
23 h3 {
24     font-size: 1.8em; /* Aumenta el tamaño del subtítulo */
25     color: #333;
26     margin-top: 15px;
27     text-align: center;
28 }
29
```

Y bueno el propósito de este código es de que crea una pagina que sirve como punto de entrada para un proyecto relacionado con la validación y creación de contraseñas seguras. Usa un diseño limpio con un título y dos enlaces para navegar entre funcionalidades clave.

Y nuestro inicio queda de la siguiente manera



Creación de código para la creación de contraseñas seguras

Ahora vamos a pasar a crear el código de la creación de contraseñas seguras.

En la primera parte del código se establece la estructura inicial de una página web en español para un generador de contraseñas seguras. También utilizamos `<html lang="es">` que indica que el contenido está en español. Al igual que `<head>` contiene metadatos, como la codificación UTF-8, el diseño responsivo y el título de la página. También enlaza un archivo CSS para estilos. Y por último en esta parte `<body>` inicia el contenido visible, mostrando un encabezado principal (`<h1>`) y un formulario (`<form>`) para capturar información que será usada para generar una contraseña.

```
<> CreacionDeContraseñas.html X
<> CreacionDeContraseñas.html > html > body > h1
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Generador de Contraseña Segura</title>
7      <link rel="stylesheet" href="css/styles.css">
8  </head>
9  <body>
10     <h1>Generador de Contraseña Segura</h1>
11     <form>
12         <table>
```


Después del código anterior vamos hacer un formulario para la entrada de datos. Primero vamos a organizar los campos en filas con el comando (<tr>) y columnas (<td>), haciendo que se alineen de manera clara, para después crear los campos de entrada con <input>, utilizaremos type="text" que permite ingresar texto (para el nombre completo). type="number" para aceptar números (para el año, mes y día de nacimiento). También utilizamos placeholder para muestra un texto indicativo dentro del campo hasta que el usuario comience a escribir. Y por último el id que asocia un campo de entrada con una etiqueta <label> para facilitar el acceso.

```
<tr>
  <td><label for="nombre">Nombre Completo:</label></td>
  <td><input type="text" id="nombre" placeholder="Ingresa tu nombre completo"></td>
</tr>
<tr>
  <td><label for="anoNacimiento">Año de Nacimiento:</label></td>
  <td><input type="number" id="anoNacimiento" placeholder="Ingresa tu año"></td>
</tr>
<tr>
  <td><label for="mesNacimiento">Mes de Nacimiento:</label></td>
  <td><input type="number" id="mesNacimiento" placeholder="Ingresa tu mes"></td>
</tr>
<tr>
  <td><label for="diaNacimiento">Día de Nacimiento:</label></td>
  <td><input type="number" id="diaNacimiento" placeholder="Ingresa tu día"></td>
</tr>
</table>
<button type="button" onclick="generarContraseña()">Generar Contraseña</button>
</form>
```

Ahora vamos a crear un botón para la creación de contraseñas. Para eso generamos un type="button" el cual vita que el botón envíe el formulario automáticamente. Después con el comando onclick="generarContraseña()": Llama a la función JavaScript generarContraseña() cuando se hace clic en el botón.

```
</table>
<button type="button" onclick="generarContraseña()">Generar Contraseña</button>
</form>
```

En la sección de salida utilizamos el <h3>: Encabezado que indica dónde se mostrará la contraseña generada. También utilizamos el comando <p id="contraseña">, un párrafo vacío donde la contraseña generada será insertada dinámicamente mediante JavaScript.

```
<h3>Contraseña Recomendada:</h3>
<p id="contraseña"></p>
```

Ya que terminamos esa parte hacemos el enlace al archivo JavaScript el cual solo se vincula a un archivo JavaScript externo el cual es (scripts.js) que contiene la lógica para generar y mostrar la contraseña.

```
<script src="js/scripts.js"></script>
```

Y ya por último en este código colocamos enlaces de navegación uno para regresar al inicio, otro para validar contraseña.

```
<a href="index.html" class="button">Regresar al inicio</a>
<a href="VerificacionDeContraseña.html" class="button">Validar tu contraseña</a>
```

Creación del código para la verificación de contraseñas

El siguiente código que vamos a desarrollar será en HTML que corresponderá a una página de registro de usuario con un formulario para crear una cuenta, donde se incluye la validación de contraseñas

Bueno en la primera parte realizamos la estructura que ya sabemos la de HTML. Los pasos importantes aquí son `<html lang="en">` que define el idioma de la página como inglés o español. `<meta charset="UTF-8">` es el que establece la codificación de caracteres para manejar texto especial. Luego `<meta name="viewport" content="width=device-width, initial-scale=1.0">` es el que asegura que la página se vea bien en dispositivos móviles, ajustando su ancho. `<title>Register</title>` es el título que aparece en la pestaña del navegador. Y por último `<link rel="stylesheet" href="css/styles.css">` que vincula el archivo de estilos CSS para la apariencia de la página.

```
<> VerificacionDeContraseña.html X
<> VerificacionDeContraseña.html > html > body > script
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Register</title>
7   <link rel="stylesheet" href="css/styles.css">
8 </head>
```

Ahora vamos a crear el formulario de registro la cual vamos a utilizar el `<html lang="en">` que define el idioma de la página como inglés. Utilizaremos el `<meta charset="UTF-8">` que establece la codificación de caracteres para manejar texto especial. `<meta name="viewport" content="width=device-width, initial-scale=1.0">` se encargara de asegurar que la página se vea bien en dispositivos móviles, ajustando su ancho. `<title>Register</title>`: El título que aparece en la pestaña del navegador. `<link rel="stylesheet" href="css/styles.css">`: Vincula el archivo de estilos CSS para la apariencia de la página.

```
<form id="registerForm">
  <label for="username">Username:</label>
  <input type="text" id="username" required>
  <label for="password">Password:</label>
  <input type="password" id="password" required>
  <label for="confirmPassword">Confirm Password:</label>
  <input type="password" id="confirmPassword" required>
  <button type="submit">Register</button>
</form>
<script src="js/scripts.js"></script>
```

Y bueno al igual que la anterior pagina vamos a poner enlaces para regresar al inicio, y formular contraseña. Para ello utilizamos `` es el enlace para regresar a la página de inicio. `` enlace para redirigir al usuario a la página de creación de contraseñas.

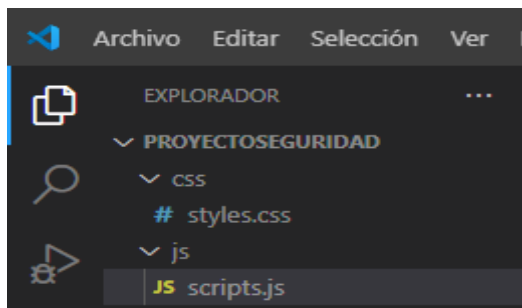
```
<a href="index.html" class="button">Regresar al inicio</a>

<a href="CreacionDeContraseñas.html" class="button">Formular tu contraseña</a>
</body>
</html>
```

El propósito del anterior código implementa un formulario de registro de usuario con campos para ingresar un nombre de usuario y una contraseña, además de confirmar la contraseña. El formulario está preparado para incluir validación de contraseñas y otros datos mediante JavaScript. También incluye enlaces para regresar al inicio o formular una contraseña en una página separada.

Creación de la validación de los formularios

Para hacer la validación de formularios en la carpeta llamada js vamos a crear otro archivo llamado scripts.js.



Este archivo scripts.js contiene dos partes principales que gestionan la validación y la generación de contraseñas en el formulario de registro, y el generador de contraseñas en la página de creación.

Lo primero que debemos de hacer es la validación de contraseñas para eso vamos a ocupar `e.preventDefault()` La cual evita que el formulario se envíe de inmediato, permitiendo que se realice la validación primero. Obtención de valores es el que extrae los valores ingresados en los campos de contraseña y confirmación de contraseña (`password` y `confirmPassword`). Verificación de coincidencia de contraseñas, si las contraseñas no coinciden, muestra un mensaje de alerta y detiene la ejecución. Expresión regular para validar la contraseña usa una expresión regular para comprobar que la contraseña tenga al menos 8 caracteres y que contenga al menos una letra minúscula (a-z), una mayúscula (A-Z), un número (\d) y un símbolo especial (@\$!%*?&). Y por último una alerta de éxito, si todas las validaciones son correctas, muestra un mensaje de "Formulario válido".

```
JS scripts.js X
js > JS scripts.js > addEventListener("submit") callback
1 document.getElementById("registerForm").addEventListener("submit", (e) => {
2   e.preventDefault();
3
4   const password = document.getElementById("password").value;
5   const confirmPassword = document.getElementById("confirmPassword").value;
6
7   if (password !== confirmPassword) {
8     alert("Las contraseñas no coinciden");
9     return;
10  }
11
12  if (!/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$/ .test(password)) {
13    alert("La contraseña debe tener al menos 8 caracteres e incluir mayúsculas, minúsculas, números y símbolos.");
14    return;
15  }
16
17  alert("El formulario es válido!");
18
19  });
```

Ahora en el paso dos vamos a hacer la generación de una contraseña segura

Aquí lo que se tiene que hacer es una obtención de datos del formulario. Se extrae el nombre completo, año de nacimiento, mes de nacimiento y día de nacimiento del formulario. También vamos a hacer la validación de campos vacíos por ejemplo si alguno de estos campos está vacío, muestra un mensaje de alerta pidiendo que se completen.

```
JS scripts.js X
js > JS scripts.js > addEventListener("submit") callback
22
23 function generarContraseña() {
24     const nombreCompleto = document.getElementById('nombre').value;
25     const añoNacimiento = document.getElementById('anoNacimiento').value;
26     const mesNacimiento = document.getElementById('mesNacimiento').value;
27     const diaNacimiento = document.getElementById('diaNacimiento').value;
28
29     if (!nombreCompleto || !añoNacimiento || !mesNacimiento || !diaNacimiento) {
30         alert("Por favor, llena todos los campos.");
31         return;
32     }
```

Para que pueda generar una contraseña segura debemos de poner nuestro primer nombre, apellido, y fecha de nacimiento.

```
JS scripts.js X
js > JS scripts.js > generarContraseña > nombre
23 function generarContraseña() {
34     let partesNombre = nombreCompleto.split(' ');
35     let nombre = partesNombre[0].charAt(0).toUpperCase(); // Primera letra del primer nombre
36     let apellido = partesNombre[partesNombre.length - 1].toUpperCase().slice(0, 3); // Primeras tres le
37
38     let año = añoNacimiento.toString().slice(-2); // Últimos dos dígitos del año
39     let mes = (mesNacimiento < 10 ? '0' + mesNacimiento : mesNacimiento); // Mes con dos dígitos
40     let día = (díaNacimiento < 10 ? '0' + díaNacimiento : díaNacimiento); // Día con dos dígitos
41
42     // Crear una contraseña compleja de al menos 12 caracteres
43     let contraseñaBase = nombre + apellido + año + mes + día;
44     while (contraseñaBase.length < 12) {
45         contraseñaBase += Math.floor(Math.random() * 10); // Añadir números aleatorios si es necesario
46     }
```

Después para la generación de contraseña lo que haces combinar la información del nombre, apellido y fecha de nacimiento. Si la longitud de la contraseña es menor a 12 caracteres, se añaden números aleatorios. Se agrega un símbolo aleatorio de una lista predefinida de caracteres especiales. Se modifica aleatoriamente la mayúscula o minúscula de algunos caracteres de la contraseña. Mostrar la contraseña generada: Finalmente, la contraseña segura generada se muestra en la página en el elemento con el id="contraseña".

```

JS scripts.js x
js > JS scripts.js > generarContraseña > [e] nombre
23 function generarContraseña() {
47
48 // Mezclar la contraseña con caracteres especiales, mayúsculas, minúsculas y números
49 let caracteresEspeciales = ['!', '@', '#', '$', '%', '&', '*', '(', ')'];
50 let contraseñaSegura = contraseñaBase;
51
52 // Incluir un símbolo aleatorio al final
53 let simboloAleatorio = caracteresEspeciales[Math.floor(Math.random() * caracteresEspeciales.length)];
54 contraseñaSegura += simboloAleatorio;
55
56 // Convertir algunos caracteres a minúsculas o mayúsculas aleatoriamente
57 contraseñaSegura = contraseñaSegura.split('').map((char, index) => {
58   if (Math.random() > 0.5) {
59     return char.toUpperCase();
60   }
61   return char.toLowerCase();
62 }).join('');
63
64 // Mostrar la contraseña recomendada
65 document.getElementById('contraseña').textContent = contraseñaSegura;

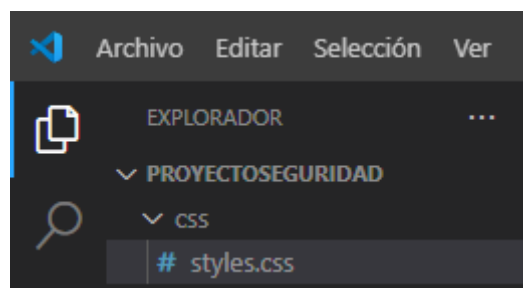
```

En este anterior código JavaScript incluye dos importantes funcionalidades, la primera es poder hacer la validación de un formulario de registro, asegurando que las contraseñas coincidan y cumplan con ciertos criterios de seguridad. La segunda es la generación de una contraseña segura, utilizando información personal del usuario (como nombre y fecha de nacimiento) y añadiendo elementos aleatorios para cumplir con requisitos de complejidad. Ambas funcionalidades ayudan a mejorar la seguridad del sistema y facilitar la creación de contraseñas fuertes.

Creación de código para los estilos de las tablas y formularios

Y ya para terminar con los códigos procedemos a realizar el estilo de cada página y los formularios.

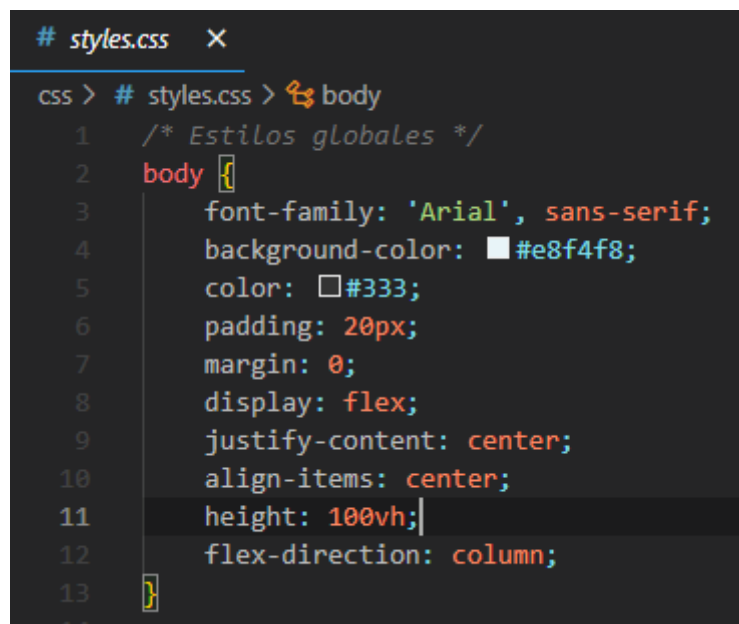
Para eso se creó desde el principio una carpeta llamada css y dentro de ella se creó un archivo llamado styles.css.



Esta hoja de estilos CSS define el diseño visual de una página web que incluye formularios, tablas y botones.

Estilos globales

Bueno vamos a empezar creando estilos globales para todo el cuerpo de la página. Para eso empezamos con un font-family: 'Arial', sans-serif este aplica la fuente Arial a todo el contenido, con una fuente alternativa sans-serif en caso de que no esté disponible. Después con el background-color: #e8f4f8 se establece un color de fondo suave para toda la página. Después padding: 20px el añade espacio dentro del body, alrededor del contenido. Con el margin: 0 se elimina cualquier margen predeterminado del body. El display: flex: Utiliza Flexbox para alinear los elementos de manera flexible. El justify-content: center: Centra el contenido horizontalmente. El align-items: center entra el contenido verticalmente. height: 100vh: Hace que el alto del body sea igual al 100% de la altura de la ventana del navegador. flex-direction: column: Organiza los elementos en una columna.

A screenshot of a code editor window titled "# styles.css". The editor shows CSS code for the body element. The code is as follows:

```
css > # styles.css > body
1  /* Estilos globales */
2  body {
3      font-family: 'Arial', sans-serif;
4      background-color: #e8f4f8;
5      color: #333;
6      padding: 20px;
7      margin: 0;
8      display: flex;
9      justify-content: center;
10     align-items: center;
11     height: 100vh;
12     flex-direction: column;
13 }
```

Estilos para los títulos

Esta parte del código define los estilos para los elementos de encabezado h1 y h3. h1: Se aumenta el tamaño de la fuente a 2.2 veces el tamaño base, se cambia el color del texto a un verde (#4CAF50), se agrega un margen inferior de 20 píxeles para separar el encabezado de los elementos siguientes, y se centra el texto horizontalmente.

h3: Se aumenta el tamaño de la fuente a 1.8 veces el tamaño base, se aplica un color gris oscuro (#333), se añade un margen superior de 15 píxeles para separarlo de los elementos previos, y también se centra el texto horizontalmente.

En resumen, ambos estilos ajustan el tamaño, color, márgenes y alineación de los encabezados h1 y h3 para mejorar la presentación visual.

```
15  /* Títulos */
16  h1 {
17      font-size: 2.2em; /* Aumenta el tamaño del título principal */
18      color: #4CAF50;
19      margin-bottom: 20px;
20      text-align: center;
21  }
22
23  h3 {
24      font-size: 1.8em; /* Aumenta el tamaño del subtítulo */
25      color: #333;
26      margin-top: 15px;
27      text-align: center;
28  }
```

Formulario y tabla

Esta parte del código define los estilos para el formulario (form), la tabla (table), las celdas de la tabla (td) y las etiquetas (label): form: Establece un fondo blanco, con un padding de 20 píxeles para crear espacio interno, bordes redondeados de 10 píxeles, y una sombra sutil para dar un efecto de profundidad. Además, tiene un ancho máximo de 500 píxeles y un margen superior de 20 píxeles para separarlo de los elementos previos. Table: Configura la tabla para que ocupe todo el ancho disponible y elimina los espacios entre las celdas con border-collapse: collapse. td: Aumenta el padding de las celdas a 12 píxeles y alinea verticalmente el contenido al centro. label: Ajusta el tamaño de la fuente a 1.1 veces el tamaño base, la pone en negrita y la hace un bloque, lo que significa que cada etiqueta ocupará su propia línea.


```

30  /* Tabla en el formulario */
31  form {
32      background: #fff;
33      padding: 20px; /* Incrementa el padding interno */
34      border-radius: 10px;
35      box-shadow: 0 6px 12px rgba(0, 0, 0, 0.2);
36      width: auto;
37      max-width: 500px; /* Ajusta el ancho máximo */
38      margin-top: 20px;
39  }
40
41  table {
42      width: 100%;
43      border-collapse: collapse;
44  }
45
46  td {
47      padding: 12px; /* Incrementa el espacio entre los elementos */
48      vertical-align: middle;
49  }
50
51  label {
52      font-size: 1.1em; /* Incrementa el tamaño de las etiquetas */
53      font-weight: bold;
54      display: block;
55  }

```

Inputs y Botones

Esta parte del código define los estilos para los elementos input y button: input, button: Ambos elementos ocupan el 100% del ancho disponible, tienen un padding de 12 píxeles, un tamaño de fuente de 1.1em, un borde gris claro, bordes redondeados de 8 píxeles y un fondo gris claro (#f9f9f9). El box-sizing: border-box asegura que el padding no afecte el tamaño total del elemento.

button: Se le da un fondo verde (#4CAF50), texto blanco, se elimina el borde y se cambia el cursor a "pointer" para indicar que es un botón clickeable. Además, se le añade un padding de 14 píxeles y un margen superior de 10 píxeles.

button:hover: Cuando el usuario pasa el mouse sobre el botón, el fondo cambia a un verde más oscuro (#45a049).

```
# styles.css X
css > # styles.css > body
56
57 /* Inputs y botones */
58 input, button {
59     width: 100%;
60     padding: 12px; /* Incrementa el tamaño interno de Los campos */
61     font-size: 1.1em; /* Incrementa el tamaño de fuente de Los campos */
62     border: 1px solid #ccc;
63     border-radius: 8px;
64     box-sizing: border-box;
65     background-color: #f9f9f9;
66 }
67
68 button {
69     background-color: #4CAF50;
70     color: white;
71     border: none;
72     cursor: pointer;
73     font-size: 1.1em; /* Incrementa el tamaño del texto del botón */
74     padding: 14px; /* Ajusta el padding del botón */
75     margin-top: 10px;
76 }
77
78 button:hover {
79     background-color: #45a049;
80 }
```

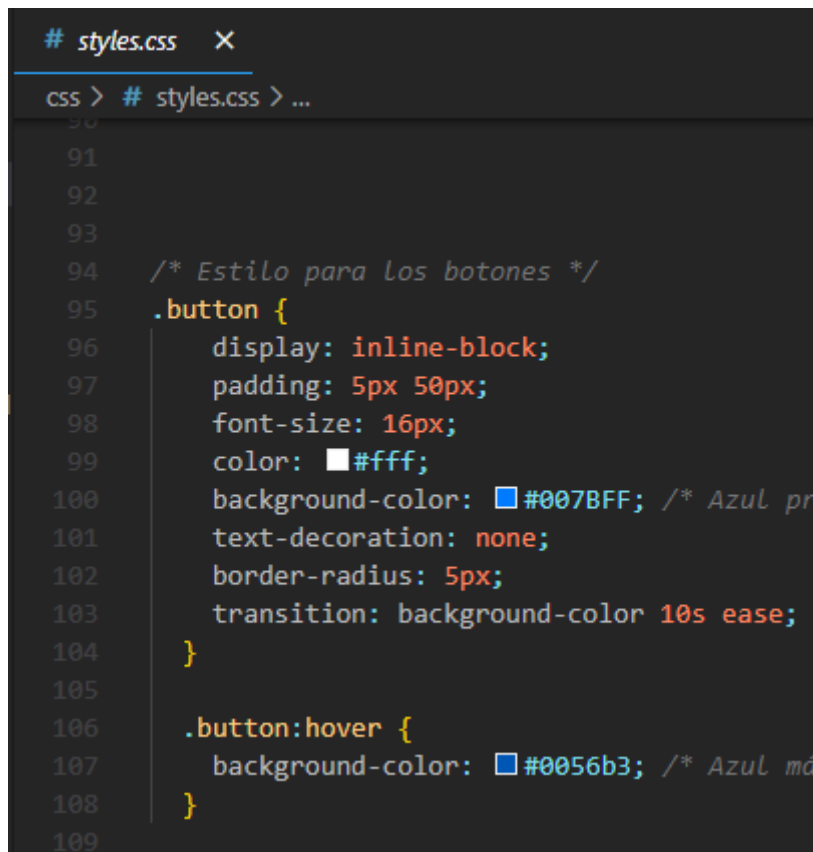
En la parte de la contraseña

Aquí solamente define el estilo del elemento con id contraseña, ajustando su tamaño de fuente, color, haciéndolo en negrita, centrando el texto y añadiendo un margen superior de 15 píxeles.

```
# styles.css X
css > # styles.css > ...
80 }
81 |
82 /* Contraseña */
83 #contraseña {
84     font-size: 1.1em; /*
85     color: #333;
86     font-weight: bold;
87     text-align: center;
88     margin-top: 15px;
89 }
90
91
```

Botones adicionales

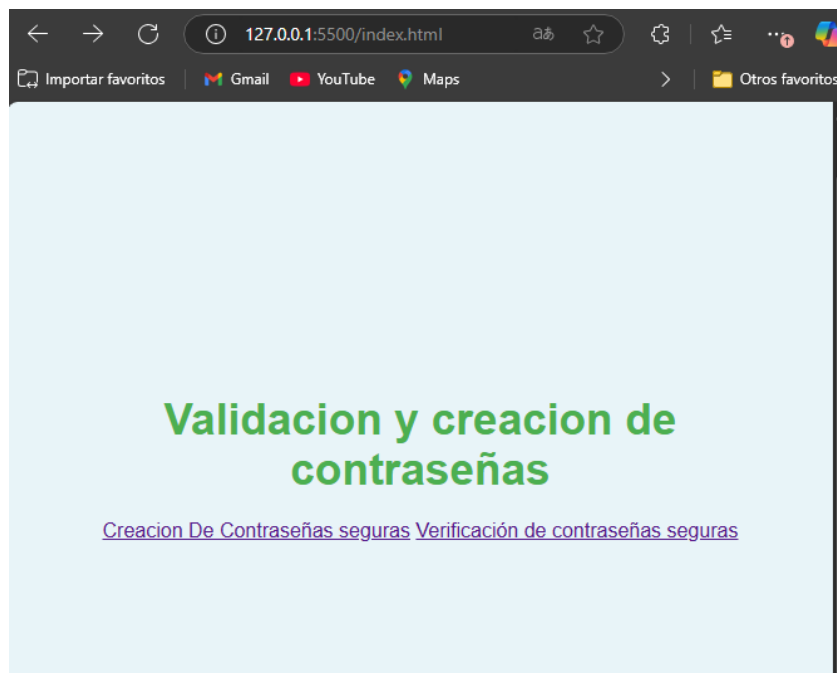
El código estiliza los botones. button, dándoles un fondo azul, texto blanco, bordes redondeados y un efecto de cambio de color al pasar el mouse. Además, aplica una transición lenta al color de fondo.

A screenshot of a code editor window titled "# styles.css" with a close button (X) in the top right corner. The editor shows CSS code for styling buttons. The code is as follows:

```
90
91
92
93
94  /* Estilo para los botones */
95  .button {
96      display: inline-block;
97      padding: 5px 50px;
98      font-size: 16px;
99      color: #fff;
100     background-color: #007BFF; /* Azul pr
101     text-decoration: none;
102     border-radius: 5px;
103     transition: background-color 10s ease;
104 }
105
106 .button:hover {
107     background-color: #0056b3; /* Azul m
108 }
109
```

Y pues bueno en resumen la hoja de estilo css proporciona estilos para un formulario de registro y una página de generación de contraseñas seguras. Define la apariencia de elementos como el fondo, los textos, las tablas, los botones y los campos de entrada. Utiliza colores suaves, bordes redondeados y sombras para un diseño limpio y moderno. Los botones cambian de color al pasar el mouse, y se asegura que los elementos estén alineados y centrados en la pantalla.

Interfaz del inicio



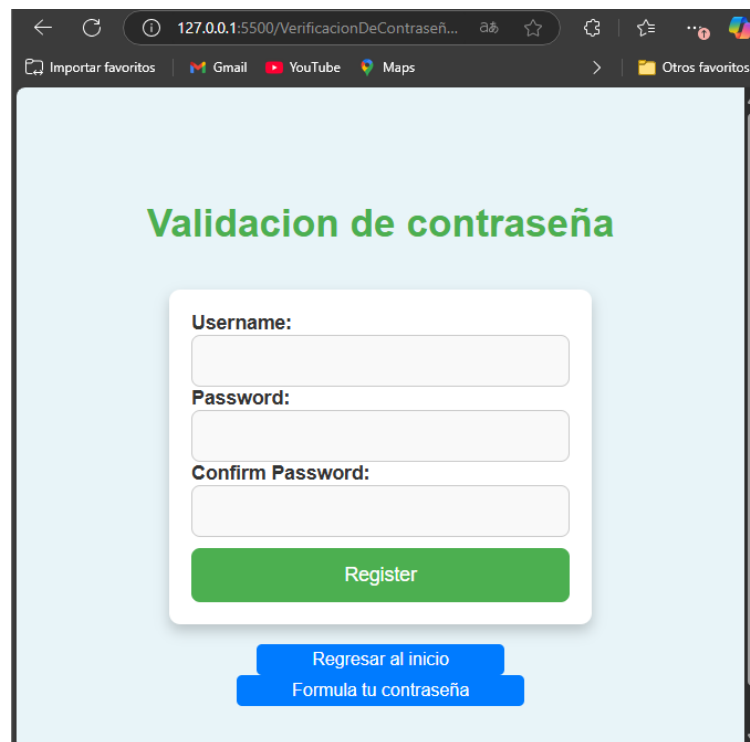
Interfaz del Generador de contraseñas seguras

A screenshot of a web browser displaying the secure password generator interface. The browser's address bar shows the URL `127.0.0.1:5500/CreacionDeContraseñas.html`. The page has a light blue background and features the main heading **Generador de Contraseña Segura** in green text. Below the heading, there is a white form with the following fields and labels:

- Nombre Completo:**
- Año de Nacimiento:**
- Mes de Nacimiento:**
- Día de Nacimiento:**

Below the form, there is a green button labeled **Generar Contraseña**. At the bottom of the page, there is a section labeled **Contraseña Recomendada:**.

Interfaz de Verificación de Contraseña



Validacion de contraseña

Username:

Password:

Confirm Password:

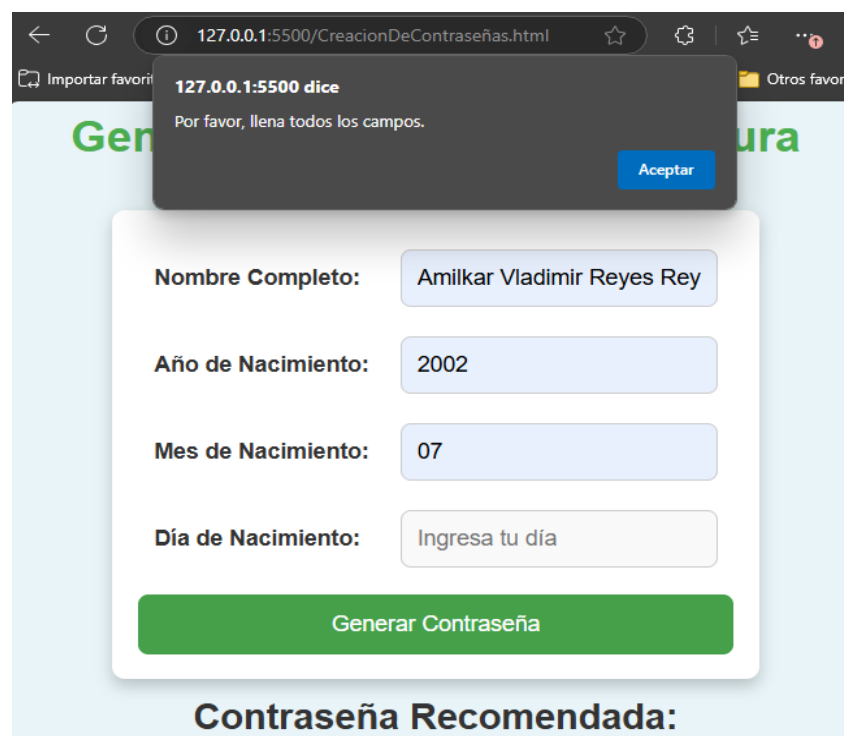
Register

Regresar al inicio

Formula tu contraseña

PRUEBAS DE LA PAGINA WEB

Al poner tu nombre completo, años de nacimiento, mes de nacimiento, pero si no rellenamos un campo nos mandara un mensaje de que por favor llenar todos los campos.



127.0.0.1:5500 dice

Por favor, llena todos los campos.

Aceptar

Nombre Completo: Amilkar Vladimir Reyes Rey

Año de Nacimiento: 2002

Mes de Nacimiento: 07

Día de Nacimiento: Ingresa tu día

Generar Contraseña

Contraseña Recomendada:

Ahora rellenamos todos los campos como debe de ser y le damos en crear contraseña y efectivamente nos crea la contraseña segura.



Nombre Completo: Amilkar Vladimir Reyes Rey

Año de Nacimiento: 2002

Mes de Nacimiento: 07

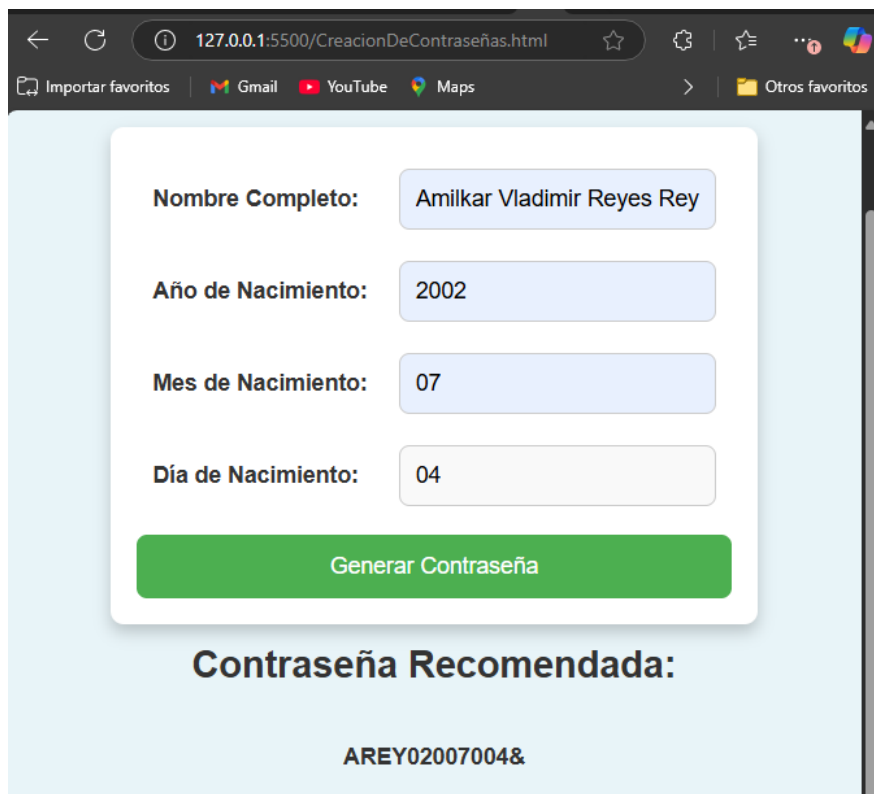
Día de Nacimiento: 04

Generar Contraseña

Contraseña Recomendada:

aREy02007004\$

Pero si nosotros queremos mas contraseñas simplemente le damos de nuevo en generar contraseña y nos dará la que queramos



Nombre Completo: Amilkar Vladimir Reyes Rey

Año de Nacimiento: 2002

Mes de Nacimiento: 07

Día de Nacimiento: 04

Generar Contraseña

Contraseña Recomendada:

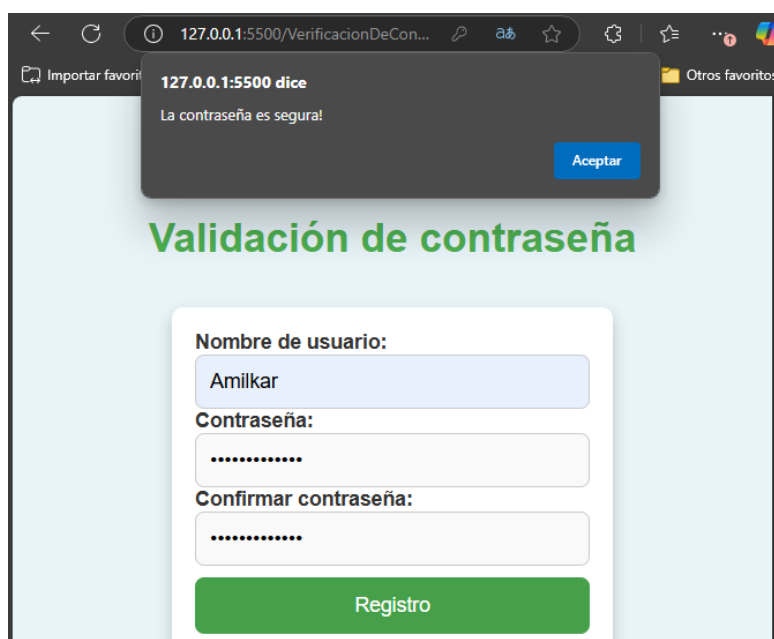
AREY02007004&

Ahora si queremos comprobar que esa contraseña es segura nos vamos a verificar contraseña.

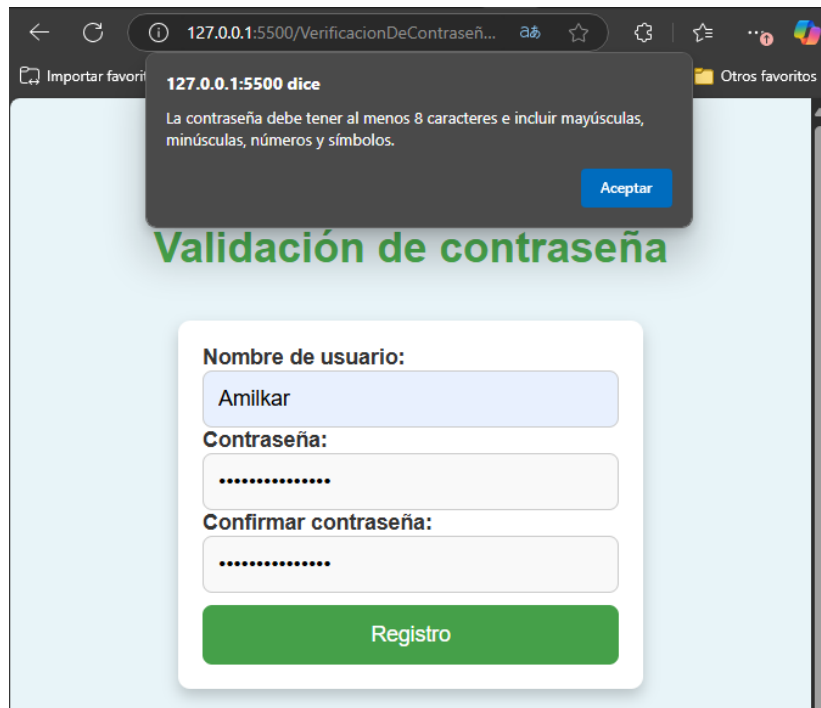


Estando ahí ponemos nuestro nombre y copiamos la contraseña que nos creamos anteriormente.

Ahora le damos registro y nos dirá que la contraseña es segura.

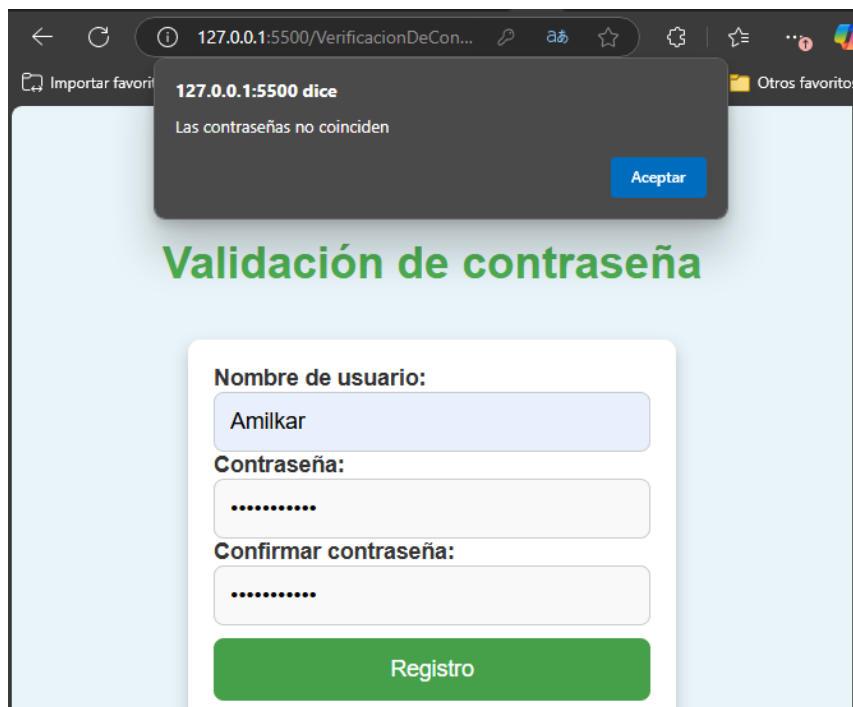


Ahora si quiero saber si una contraseña que yo mismo cree pero solo tiene puras mayúscula o puros números o símbolo me marcara un error en donde dice que la contraseña debe de tener al menos 8 caracteres e incluir mayúsculas, minúsculas, números y símbolos.



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5500/VerificacionDeContraseña...". A dark grey error message box is overlaid on the page, stating: "127.0.0.1:5500 dice: La contraseña debe tener al menos 8 caracteres e incluir mayúsculas, minúsculas, números y símbolos." with an "Aceptar" button. The background page is titled "Validación de contraseña" in green. Below the title is a registration form with the following fields: "Nombre de usuario:" (containing "Amilkar"), "Contraseña:" (containing "....."), and "Confirmar contraseña:" (containing "....."). A green "Registro" button is at the bottom of the form.

También si ponemos mal las contraseñas nos dice que no coinciden.



The screenshot shows the same web browser window as before, but the error message box now states: "127.0.0.1:5500 dice: Las contraseñas no coinciden" with an "Aceptar" button. The background page remains the same, titled "Validación de contraseña" in green, with the registration form containing "Amilkar" for the username and two identical "....." for the password fields. The "Registro" button is still present at the bottom.

También debemos de llenar todos los campos sino nos marcara error y dira que llenemos todos los campos.

Validación de contraseña

Nombre de usuario:

Contraseña:

Confirmar contraseña:

 Rellena este campo.

[Regresar al inicio](#)
[Formula tu contraseña](#)

Y si ponemos todo correcto nos dice que la contraseña es segura

127.0.0.1:5500/VerificacionDeCon...
127.0.0.1:5500 dice
La contraseña es segura!
[Aceptar](#)

Validacion de contraseña

Username:

Password:

Confirm Password:

[Register](#)

CONCLUSIÓN

Nuestro proyecto "Validación y Creación de Contraseñas Seguras" busca proporcionar una aplicación web robusta y segura para la generación y verificación de contraseñas. A través de una interfaz intuitiva, permite a los usuarios crear contraseñas utilizando sus datos personales, asegurando que estas cumplan con los estándares de seguridad mediante la validación y el uso de cifrado.

El desarrollo se llevó a cabo utilizando tecnologías como HTML, CSS, JavaScript, y Node.js, lo que permitió construir una aplicación modular y extensible, con un frontend sencillo y un backend eficiente. La implementación de validaciones tanto para el registro de usuarios como para la creación de contraseñas, junto con la generación automática de contraseñas seguras, mejora la seguridad general de los sistemas que dependen de contraseñas robustas.

En cuanto al diseño visual, se cuidaron aspectos de usabilidad y accesibilidad, con un estilo limpio y profesional, optimizado para diferentes dispositivos. Las pruebas realizadas confirmaron la funcionalidad del sistema, mostrando que la aplicación es capaz de generar contraseñas seguras y validar las contraseñas ingresadas por el usuario de manera efectiva. Este proyecto no solo mejora la seguridad en el manejo de contraseñas, sino que también establece una base sólida para futuros desarrollos en la creación de aplicaciones web seguras.