

#### 

#### Índice:

[Índice:](#_64sjm11idq5n)

[1. Introducción](#_saayrnm6j800)

[2. Selección del Algoritmo de Hashing](#_at20nlw7fzrx)

[3. Almacenamiento Seguro de Contraseñas](#_4qjzkrylk5b)

[4. Pruebas de Seguridad](#_czr9sgc0ntak)

[5. Verificación de Contraseñas](#_nvw6n8809ztp)

[6. Funcionalidad de Cambio de Contraseña](#_56c1euxinf0c)

[7. Generación de Contraseñas con IA en Google Colab](#_65tcvzl86zum)

[7.1 Flujo de Trabajo con la API para la IA](#_cdfpyr1u6wx4)

[8. Arquitectura del Sistema](#_l74lenpgkf5o)

[9. Mantenimiento y Actualización](#_dv8lqytkuiku)

[10. Evaluador de Contraseñas:](#_9bonx134hftp)

[Descripción del Sistema](#_x7xglzmx979s)

[Arquitectura del Sistema](#_nrf3197c60v8)

[Especificaciones Técnicas](#_iyt6tbafu4kj)

[Pruebas de Seguridad Realizadas](#_dyop6nz3j2vf)

[11. Gestor de Contraseñas:](#_rh01nxo3uzpa)

[Descripción del Sistema](#_3oricc41ihtv)

[Arquitectura del Sistema](#_2wmhoer22kkj)

[Especificaciones Técnicas](#_epcflfgi5fq4)

[Pruebas de Seguridad Realizadas](#_8zt32z763l0n)

[Código y Decisiones de Diseño](#_5jpxzzdr3tyu)

#### 

#### 

#### 

#### 

#### 

### 1. Introducción

* Objetivo del Proyecto: Desarrollar un sistema para el almacenamiento seguro de contraseñas que emplee el algoritmo bcrypt para evitar el almacenamiento en texto plano, garantizando la protección de la información de los usuarios.
* Descripción: El sistema permite crear cuentas, autenticar usuarios y gestionar contraseñas mediante el uso de bcrypt, un algoritmo diseñado específicamente para proteger contraseñas. Adicionalmente, se utiliza una IA en Google Colab para la generación de contraseñas seguras, que luego se envían por correo electrónico al usuario.

### 2. Selección del Algoritmo de Hashing

* Algoritmo Elegido: bcrypt
  + Motivo de Selección: bcrypt es ampliamente utilizado en sistemas de almacenamiento de contraseñas por su capacidad de incrementar la seguridad mediante el ajuste de su "cost" o número de rondas de hashing.
  + Ventajas de bcrypt:
    - Protección Contra Ataques de Fuerza Bruta: bcrypt permite ajustar el "cost" o factor de trabajo, incrementando el tiempo requerido para cada intento de descifrado.
    - Resistencia a Ataques de Hardware: bcrypt fue diseñado para ser computacionalmente costoso, dificultando los ataques de fuerza bruta, especialmente mediante hardware especializado.
  + Comparativa con otros Algoritmos:
    - scrypt: Aunque scrypt es seguro, su alto consumo de memoria no es necesario para los requisitos actuales de esta aplicación.
    - Argon2: Otro algoritmo seguro y moderno, aunque bcrypt es ampliamente adoptado y cumple con los requisitos del sistema.

### 3. Almacenamiento Seguro de Contraseñas

* Estrategia de Almacenamiento:
  + Las contraseñas de los usuarios se almacenan únicamente en su forma hasheada mediante bcrypt, y ninguna contraseña se guarda en texto plano.
  + Salting: Cada contraseña se hashea junto a un valor de salt único y aleatorio, que bcrypt genera automáticamente, incrementando así la seguridad contra ataques de diccionario.
  + Proceso de Hasheo y Almacenamiento:
    1. El usuario introduce una contraseña.
    2. bcrypt genera automáticamente un salt y aplica el proceso de hashing.
    3. El hash resultante se almacena de manera segura en la base de datos.

### 4. Pruebas de Seguridad

* Pruebas Realizadas:
  + Validación de Autenticación Segura: Se verificó que el sistema autentique a los usuarios únicamente cuando el hash coincide, probando múltiples combinaciones de entradas incorrectas para confirmar la seguridad.
  + Pruebas de Entrega de Contraseñas: La funcionalidad de envío de contraseñas generadas por la IA se evaluó para garantizar que los correos fueran enviados de manera segura y confiable.
* Resultados:
  + La configuración de bcrypt y la integración de la IA para la generación de contraseñas cumplen con los estándares de seguridad requeridos, garantizando resistencia a ataques de fuerza bruta.

### 5. Verificación de Contraseñas

* Proceso:
  + Durante el inicio de sesión, el sistema toma la contraseña ingresada, aplica el proceso de hashing bcrypt y compara el resultado con el hash almacenado.
* Pruebas:
  + Se realizaron pruebas exhaustivas para asegurar que el sistema valide correctamente las contraseñas y rechace cualquier ingreso incorrecto.

#### 6. Funcionalidad de Cambio de Contraseña

El cambio de contraseña se implementa para brindar a los usuarios la opción de actualizar sus contraseñas de manera segura. Aquí se describe cómo funciona:

1. Validación de Usuario y Contraseña Actual:
   * Al recibir la solicitud, el sistema verifica que el nombre de usuario ingresado exista en la base de datos. Luego, compara la contraseña actual ingresada por el usuario con el hash de la contraseña almacenada, utilizando password\_verify. Este proceso asegura que solo el propietario de la cuenta pueda cambiar la contraseña.
2. Hashing Seguro con Bcrypt:
   * La nueva contraseña se cifra utilizando bcrypt, ajustado con un costo de 12 (definido como BCRYPT\_COST). Este valor proporciona una buena resistencia contra ataques de fuerza bruta. El hash de la nueva contraseña se almacena en la base de datos reemplazando el hash anterior.
3. Proceso de Actualización:
   * Una vez que la nueva contraseña ha sido cifrada, se actualiza el registro en la base de datos. Si el proceso es exitoso, el sistema envía al usuario a la pantalla de inicio de sesión con una notificación de que la contraseña se cambió correctamente.
4. Manejo de Errores y Mensajes al Usuario:
   * En caso de que el nombre de usuario no se encuentre en la base de datos o la contraseña actual ingresada no coincide con el hash almacenado, el sistema envía un mensaje de error específico para ayudar al usuario a resolver el problema.

### 7. Generación de Contraseñas con IA en Google Colab

Como una innovación adicional, el sistema utiliza una Inteligencia Artificial (IA) desarrollada en Google Colab para generar contraseñas seguras que se envían a los usuarios por correo electrónico. Sin embargo, esta IA aún no está integrada directamente en la aplicación web, ya que se encuentra en una fase de entrenamiento y ajuste. Este entrenamiento es necesario para que la IA aprenda a manejar distintos patrones y caracteres en la generación de contraseñas seguras. Además, cuestiones de seguridad y permisos en el correo electrónico han retrasado su implementación completa.

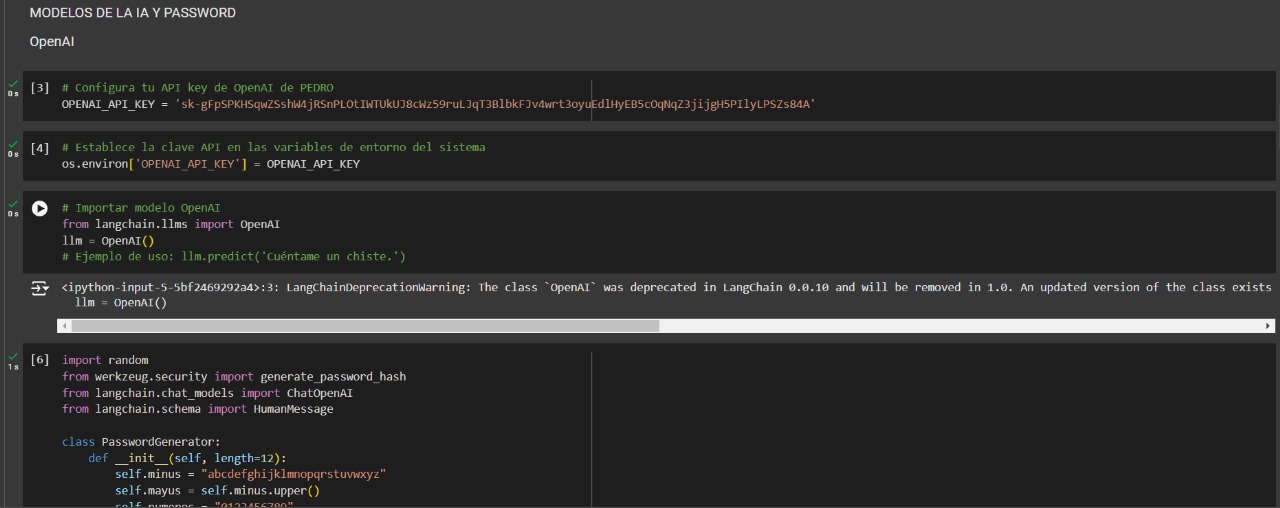
* Uso de IA para Generación de Contraseñas:
  + Se ha implementado un modelo de IA en Google Colab para generar contraseñas seguras con configuraciones avanzadas de seguridad.
  + Proceso:
    - La IA crea una contraseña segura, integrando parámetros como mayúsculas, minúsculas, números y caracteres especiales según las preferencias del usuario.
    - Una vez generada, la contraseña es enviada al usuario por correo electrónico de manera segura.
  + Ventajas:
    - Automatización: La IA permite la creación rápida y segura de contraseñas robustas.
    - Seguridad: La contraseña generada es aleatoria y cumple con estándares avanzados, reforzando la protección del usuario.

Por ahora, las credenciales de correo electrónico y la contraseña generada para los usuarios se encuentran directamente en el código fuente, hasta que la IA se integre completamente en la aplicación. Esto garantiza que, temporalmente, el sistema funcione sin interrupciones mientras se termina de entrenar y conectar la IA a la interfaz web.

#### 7.1 Flujo de Trabajo con la API para la IA

Para generar contraseñas seguras a través de IA, el sistema realiza los siguientes pasos mediante una API:

1. Solicitud desde el Código: Cuando el sistema necesita generar una nueva contraseña, llama a una función password\_with\_generator, que crea un mensaje con las instrucciones detalladas para la IA (cómo debe ser la contraseña, los requisitos de longitud, los caracteres permitidos, etc.).
2. Envío de la Solicitud: La función envía estas instrucciones a los servidores de OpenAI utilizando una solicitud HTTP POST, incluyendo la clave API en el encabezado para autenticarse.
3. Respuesta del Servidor: OpenAI procesa la solicitud y devuelve una respuesta que contiene la contraseña generada.
4. Procesamiento de la Respuesta: El sistema recibe la contraseña generada y, en este caso, la imprime en la consola y la envía al correo electrónico del usuario.



### 8. Arquitectura del Sistema

* Flujo del Proceso:
  + Registro de Usuario: Los usuarios introducen sus datos y configuran una contraseña segura (manualmente o con la contraseña generada por la IA).
  + Hasheo de contraseña: La contraseña se hashea usando bcrypt, y el hash se almacena en la base de datos.
  + Inicio de sesión: El usuario introduce sus credenciales, el sistema hashea la contraseña ingresada y la compara con el hash almacenado para autenticar.
  + Cambio de contraseña: Los usuarios pueden actualizar su contraseña mediante el mismo proceso de hasheo y almacenamiento seguro.
* Tecnologías Utilizadas:
  + Lenguaje de Programación: PHP para la lógica del backend.
  + Base de Datos: MySQL para almacenar los datos de usuario y las contraseñas hasheadas.
  + Algoritmo de Hashing: bcrypt para el almacenamiento seguro de contraseñas.
  + Google Colab: Utilizado para la generación de contraseñas con IA.

### 9. Mantenimiento y Actualización

* Revisión Periódica de Configuración de bcrypt: Se recomienda incrementar el cost periódicamente para asegurar que el sistema mantenga su resistencia contra el hardware moderno.
* Pruebas de Seguridad Anuales: Para asegurar que el sistema siga cumpliendo con los estándares de seguridad.

### 10. Evaluador de Contraseñas:

### Descripción del Sistema

El evaluador de contraseñas analiza las contraseñas ingresadas por los usuarios para determinar su nivel de seguridad según los siguientes criterios:

* Longitud: Aumenta la puntuación a medida que crece.
* Complejidad: Evalúa la presencia de caracteres en mayúsculas, minúsculas, números y símbolos especiales.

### Arquitectura del Sistema

1. Frontend:
   * Interfaz que permite al usuario ingresar una contraseña.
   * Muestra la puntuación y recomendaciones de mejora.
2. Backend:
   * Algoritmo para evaluar contraseñas según reglas predefinidas.
   * Devuelve la puntuación y advertencias específicas al usuario.
3. Escala de Puntuación:
   * 0-50: Baja seguridad (insegura).
   * 51-80: Media seguridad (mejorable).
   * 81-100: Alta seguridad (segura).

### Especificaciones Técnicas

* Criterios de Complejidad:
  + Longitud mínima: 12 caracteres.
  + Puntos adicionales:
  + Uso de mayúsculas y minúsculas.
  + Inclusión de números y símbolos especiales.

### Pruebas de Seguridad Realizadas

1. Simulaciones de Entrada:
   * Las contraseñas simples puntuaron adecuadamente bajo.
   * Las contraseñas complejas alcanzaron puntajes altos.

### 11. Gestor de Contraseñas:

### Descripción del Sistema

El gestor de contraseñas almacena y organiza credenciales de manera segura. Garantiza la privacidad mediante cifrado y ofrece accesibilidad mediante funciones como autocompletado seguro.

### Arquitectura del Sistema

1. Frontend:
   * Interfaz para añadir, modificar o eliminar credenciales.
   * Vista para listar credenciales almacenadas.
2. Backend:
   * Almacena contraseñas cifradas en la base de datos.
   * Gestiona el proceso de autenticación
3. Base de Datos:
   * Cifrado de datos sensibles
   * Uso de tablas para almacenar servicios, usuarios y contraseñas.

### Especificaciones Técnicas

1. Cifrado de Contraseñas:
   * Algoritmo: bycript para el registro de cifrado
   * Clave: aleatoria para mayor seguridad.
2. Autenticación Multifactor (MFA):
   * Generación de OTP (One-Time Password).
   * Validación del código antes de otorgar acceso al gestor.
3. Funciones Esenciales:
   * Integración con navegadores para autocompletado seguro.

### Pruebas de Seguridad Realizadas

1. Resistencia al Descifrado:
   * Las contraseñas cifradas no pudieron ser descifradas sin la clave adecuada.
2. Validación
   * Rechazó intentos de acceso con códigos incorrectos.

### Código y Decisiones de Diseño

registro:

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<title>Registro de Usuario</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f6f9;

display: flex;

justify-content: center;

align-items: flex-start;

min-height: 100vh;

margin: 0;

padding: 20px;

}

.container {

width: 100%;

max-width: 400px;

padding: 30px;

background-color: #ffffff;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h2 {

text-align: center;

margin-bottom: 20px;

color: #2c3e50;

}

label {

display: block;

margin-top: 15px;

color: #34495e;

}

input[type="text"],

input[type="email"],

input[type="password"],

input[type="number"],

select {

width: 100%;

padding: 10px;

margin-top: 5px;

box-sizing: border-box;

border: 1px solid #bdc3c7;

border-radius: 4px;

}

button {

width: 100%;

padding: 12px;

margin-top: 20px;

background-color: #1f9fea;

color: #ffffff;

border: none;

border-radius: 4px;

cursor: pointer;

font-size: 16px;

}

button:hover {

background-color: #2c3e50;

}

.contrasena-generada {

margin-top: 20px;

padding: 15px;

background-color: #f1f1f1;

border: 1px solid #bdc3c7;

border-radius: 4px;

text-align: center;

display: none;

}

.prompt-section {

margin-top: 10px;

display: none;

color: #34495e;

}

.prompt-section label {

font-weight: bold;

}

.input-container {

position: relative;

}

.password-toggle {

position: absolute;

right: 10px;

top: 50%;

transform: translateY(-50%);

cursor: pointer;

font-size: 18px;

color: #7f8c8d;

}

.password-toggle:hover {

color: #2c3e50;

}

.recomendaciones {

color: #e74c3c;

margin-top: 5px;

font-size: 14px;

}

</style>

<script src="script.js" defer></script>

</head>

<body>

<div class="container">

<h2>Registro de Usuario</h2>

<form action="procesar\_registro.php" method="POST">

<label for="nombre\_usuario">Nombre de Usuario:</label>

<input type="text" id="nombre\_usuario" name="nombre\_usuario" required>

<label for="email">Correo Electrónico:</label>

<input type="email" id="email" name="email" required>

<label for="nombre">Nombre:</label>

<input type="text" id="nombre" name="nombre" placeholder="Tu nombre">

<label for="apellido">Apellido:</label>

<input type="text" id="apellido" name="apellido" placeholder="Tu apellido">

<label for="fecha\_nacimiento">Fecha de Nacimiento (DDMMYYYY):</label>

<input type="text" id="fecha\_nacimiento" name="fecha\_nacimiento" placeholder="Ejemplo: 01011990">

<label for="rol">Selecciona tu rol:</label>

<select id="rol" name="rol" required>

<option value="">--Selecciona un rol--</option>

<option value="cliente">Cliente</option>

<option value="agente">Agente</option>

<option value="admin">Administrador</option>

</select>

<label for="contraseña">Contraseña:</label>

<div class="input-container">

<input type="password" id="contraseña" name="contraseña" required oninput="evaluarYMostrarRecomendaciones()">

<span class="password-toggle" onclick="togglePassword()">👁️</span>

</div>

<div id="recomendaciones" class="recomendaciones"></div>

<button type="button" onclick="mostrarOpciones()">Configurar Contraseña</button>

<div id="prompt-section" class="prompt-section">

<label>Configuración de Contraseña:</label><br>

Longitud (mínimo 8): <input type="number" id="longitud" min="8" value="12"><br>

<input type="checkbox" id="mayusculas"> Mayúsculas

<input type="checkbox" id="minusculas" checked> Minúsculas

<input type="checkbox" id="numeros" checked> Números

<input type="checkbox" id="especiales"> Caracteres especiales

<input type="checkbox" id="diccionario"> Usar palabras del diccionario

<button type="button" onclick="crearContrasena()">Generar Contraseña</button>

</div>

<div class="contrasena-generada" id="contrasena-generada">

<h3>Contraseña Generada:</h3>

<input type="text" id="contrasena-copy" readonly>

<button type="button" onclick="copiarContraseña()">Copiar Contraseña</button>

</div>

<button type="submit">Registrarse</button>

</form>

<p>¿Ya tienes una cuenta? <a href="Iniciodesecion.php">Inicia Sesión</a></p>

</div>

</body>

</html>

script:

const bancoDePalabras = [

"sol", "luna", "estrella", "cielo", "mar", "montaña", "río", "bosque", "flor",

"árbol", "nube", "viento", "fuego", "agua", "tierra", "catarata", "brisa", "noche",

"día", "amanecer", "atardecer", "cascada", "lago", "isla", "valle", "desierto",

"ciudad", "aldea", "puente", "sendero", "caminata", "rayo", "relámpago", "nieve",

"lluvia", "tormenta", "huracán", "tornado", "calor", "frío", "templo", "castillo",

"pirámide", "túnel", "puerta", "ventana", "silla", "mesa", "cama", "libro", "pájaro",

"pez", "elefante", "tigre", "león", "gato", "perro", "serpiente", "cocodrilo", "jirafa", "mono"

];

function evaluarContrasena(contrasena) {

let puntuacion = 0;

const recomendaciones = [];

if (contrasena.length >= 12) {

puntuacion += 2;

} else if (contrasena.length >= 8) {

puntuacion += 1;

} else {

recomendaciones.push("La contraseña debe tener al menos 8 caracteres.");

}

const tieneMayusculas = /[A-Z]/.test(contrasena);

const tieneMinusculas = /[a-z]/.test(contrasena);

const tieneNumeros = /\d/.test(contrasena);

const tieneEspeciales = /[!@#$%^&\*(),.?":{}|<>]/.test(contrasena);

if (tieneMayusculas) puntuacion++;

if (tieneMinusculas) puntuacion++;

if (tieneNumeros) puntuacion++;

if (tieneEspeciales) puntuacion++;

const patronesComunes = ["12345678", "password", "qwerty", "abc123"];

if (patronesComunes.some(patron => contrasena.includes(patron))) {

recomendaciones.push("Evita utilizar contraseñas comunes o patrones fáciles de adivinar.");

}

let nivel;

if (puntuacion <= 2) {

nivel = "Baja";

} else if (puntuacion <= 4) {

nivel = "Media";

} else {

nivel = "Alta";

}

return { nivel, recomendaciones };

}

function evaluarYMostrarRecomendaciones() {

const contrasena = document.getElementById('contraseña').value;

const evaluacion = evaluarContrasena(contrasena);

const recomendacionesEl = document.getElementById('recomendaciones');

recomendacionesEl.innerHTML = `Nivel de Seguridad: ${evaluacion.nivel}. ${evaluacion.recomendaciones.join(' ')}`;

}

function mostrarOpciones() {

const section = document.getElementById('prompt-section');

const button = document.querySelector("button[onclick='mostrarOpciones()']");

if (section.style.display === 'none' || section.style.display === '') {

section.style.display = 'block';

button.textContent = "Ocultar Opciones";

} else {

section.style.display = 'none';

button.textContent = "Configurar Contraseña";

}

}

function togglePassword() {

const passwordInput = document.getElementById('contraseña');

passwordInput.type = passwordInput.type === "password" ? "text" : "password";

}

function crearContrasena() {

const longitud = parseInt(document.getElementById('longitud').value);

if (longitud < 8) {

alert("La longitud debe ser igual o mayor a 8.");

return;

}

const mayusculas = document.getElementById('mayusculas').checked;

const minusculas = document.getElementById('minusculas').checked;

const numeros = document.getElementById('numeros').checked;

const especiales = document.getElementById('especiales').checked;

const diccionario = document.getElementById('diccionario').checked;

let caracteres = '';

if (mayusculas) caracteres += 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';

if (minusculas) caracteres += 'abcdefghijklmnopqrstuvwxyz';

if (numeros) caracteres += '0123456789';

if (especiales) caracteres += '!@#$%^&\*()-\_=+[]{}|;:,.<>?';

let contrasena = '';

if (diccionario) {

let numPalabras = Math.ceil(longitud / 5);

for (let i = 0; i < numPalabras; i++) {

const palabraAleatoria = bancoDePalabras[Math.floor(Math.random() \* bancoDePalabras.length)];

contrasena += palabraAleatoria;

if (contrasena.length >= longitud) break;

}

contrasena = contrasena.slice(0, longitud);

} else {

if (caracteres === '') {

alert('Selecciona al menos una opción.');

return;

}

for (let i = 0; i < longitud; i++) {

contrasena += caracteres.charAt(Math.floor(Math.random() \* caracteres.length));

}

}

document.getElementById('contrasena-copy').value = contrasena;

document.getElementById('contrasena-generada').style.display = 'block';

document.getElementById('contraseña').value = contrasena;

const evaluacion = evaluarContrasena(contrasena);

document.getElementById('recomendaciones').innerText = `Nivel de Seguridad: ${evaluacion.nivel}. ${evaluacion.recomendaciones.join(' ')}`;

}

function copiarContraseña() {

const contrasenaCopy = document.getElementById("contrasena-copy");

contrasenaCopy.select();

document.execCommand("copy");

alert("Contraseña copiada al portapapeles");

}

document.addEventListener('DOMContentLoaded', (event) => {

document.getElementById('contraseña').addEventListener('input', evaluarYMostrarRecomendaciones);

});

registro:   
<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<title>Registro de Usuario</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f6f9;

display: flex;

justify-content: center;

align-items: flex-start;

min-height: 100vh;

margin: 0;

padding: 20px;

}

.container {

width: 100%;

max-width: 400px;

padding: 30px;

background-color: #ffffff;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h2 {

text-align: center;

margin-bottom: 20px;

color: #2c3e50;

}

label {

display: block;

margin-top: 15px;

color: #34495e;

}

input[type="text"],

input[type="email"],

input[type="password"],

input[type="number"],

select {

width: 100%;

padding: 10px;

margin-top: 5px;

box-sizing: border-box;

border: 1px solid #bdc3c7;

border-radius: 4px;

}

button {

width: 100%;

padding: 12px;

margin-top: 20px;

background-color: #1f9fea;

color: #ffffff;

border: none;

border-radius: 4px;

cursor: pointer;

font-size: 16px;

}

button:hover {

background-color: #2c3e50;

}

.contrasena-generada {

margin-top: 20px;

padding: 15px;

background-color: #f1f1f1;

border: 1px solid #bdc3c7;

border-radius: 4px;

text-align: center;

display: none;

}

.prompt-section {

margin-top: 10px;

display: none;

color: #34495e;

}

.prompt-section label {

font-weight: bold;

}

.input-container {

position: relative;

}

.password-toggle {

position: absolute;

right: 10px;

top: 50%;

transform: translateY(-50%);

cursor: pointer;

font-size: 18px;

color: #7f8c8d;

}

.password-toggle:hover {

color: #2c3e50;

}

.recomendaciones {

color: #e74c3c;

margin-top: 5px;

font-size: 14px;

}

</style>

<script src="script.js" defer></script>

</head>

<body>

<div class="container">

<h2>Registro de Usuario</h2>

<form action="procesar\_registro.php" method="POST">

<label for="nombre\_usuario">Nombre de Usuario:</label>

<input type="text" id="nombre\_usuario" name="nombre\_usuario" required>

<label for="email">Correo Electrónico:</label>

<input type="email" id="email" name="email" required>

<label for="nombre">Nombre:</label>

<input type="text" id="nombre" name="nombre" placeholder="Tu nombre">

<label for="apellido">Apellido:</label>

<input type="text" id="apellido" name="apellido" placeholder="Tu apellido">

<label for="fecha\_nacimiento">Fecha de Nacimiento (DDMMYYYY):</label>

<input type="text" id="fecha\_nacimiento" name="fecha\_nacimiento" placeholder="Ejemplo: 01011990">

<label for="rol">Selecciona tu rol:</label>

<select id="rol" name="rol" required>

<option value="">--Selecciona un rol--</option>

<option value="cliente">Cliente</option>

<option value="agente">Agente</option>

<option value="admin">Administrador</option>

</select>

<label for="contraseña">Contraseña:</label>

<div class="input-container">

<input type="password" id="contraseña" name="contraseña" required oninput="evaluarYMostrarRecomendaciones()">

<span class="password-toggle" onclick="togglePassword()">👁️</span>

</div>

<div id="recomendaciones" class="recomendaciones"></div>

<button type="button" onclick="mostrarOpciones()">Configurar Contraseña</button>

<div id="prompt-section" class="prompt-section">

<label>Configuración de Contraseña:</label><br>

Longitud (mínimo 8): <input type="number" id="longitud" min="8" value="12"><br>

<input type="checkbox" id="mayusculas"> Mayúsculas

<input type="checkbox" id="minusculas" checked> Minúsculas

<input type="checkbox" id="numeros" checked> Números

<input type="checkbox" id="especiales"> Caracteres especiales

<input type="checkbox" id="diccionario"> Usar palabras del diccionario

<button type="button" onclick="crearContrasena()">Generar Contraseña</button>

</div>

<div class="contrasena-generada" id="contrasena-generada">

<h3>Contraseña Generada:</h3>

<input type="text" id="contrasena-copy" readonly>

<button type="button" onclick="copiarContraseña()">Copiar Contraseña</button>

</div>

<button type="submit">Registrarse</button>

</form>

<p>¿Ya tienes una cuenta? <a href="Iniciodesecion.php">Inicia Sesión</a></p>

</div>

</body>

</html>

cambio de contraseña:   
<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Cambiar Contraseña</title>

<style>

/\* Estilos CSS \*/

body {

font-family: Arial, sans-serif;

background-color: #f4f6f9;

display: flex;

justify-content: center;

align-items: flex-start;

min-height: 100vh;

margin: 0;

padding: 20px;

}

.container {

width: 100%;

max-width: 400px;

padding: 30px;

background-color: #ffffff;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h2 {

text-align: center;

margin-bottom: 20px;

color: #2c3e50;

}

label {

display: block;

margin-top: 15px;

color: #34495e;

font-weight: bold;

}

input[type="text"],

input[type="password"],

input[type="number"] {

width: 100%;

padding: 10px;

margin-top: 5px;

box-sizing: border-box;

border: 1px solid #bdc3c7;

border-radius: 4px;

font-size: 16px;

}

button {

width: 100%;

padding: 12px;

margin-top: 20px;

background-color: #1f9fea;

color: #ffffff;

border: none;

border-radius: 4px;

cursor: pointer;

font-size: 16px;

}

button:hover {

background-color: #2c3e50;

}

.message {

margin-top: 20px;

text-align: center;

font-size: 14px;

color: #2c3e50;

}

.error {

color: #e74c3c;

}

.success {

color: #2ecc71;

}

.input-container {

position: relative;

}

.password-toggle {

position: absolute;

right: 10px;

top: 50%;

transform: translateY(-50%);

cursor: pointer;

font-size: 18px;

color: #7f8c8d;

}

.password-toggle:hover {

color: #2c3e50;

}

.prompt-section {

margin-top: 10px;

display: none;

color: #34495e;

}

.prompt-section label {

font-weight: bold;

}

p {

text-align: center;

margin-top: 15px;

color: #7f8c8d;

}

a {

color: #1f9fea;

text-decoration: none;

}

.contrasena-generada {

margin-top: 20px;

padding: 15px;

background-color: #f1f1f1;

border: 1px solid #bdc3c7;

border-radius: 4px;

text-align: center;

display: none;

}

</style>

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

</head>

<body>

<div class="container">

<h2>Cambiar Contraseña</h2>

<form action="cambiar\_contrasena.php" method="POST">

<label for="nombre\_usuario">Nombre de Usuario:</label>

<input type="text" id="nombre\_usuario" name="nombre\_usuario" required>

<label for="contraseña\_actual">Contraseña Actual:</label>

<div class="input-container">

<input type="password" id="contraseña\_actual" name="contraseña\_actual" required>

<span class="password-toggle" onclick="togglePassword('contraseña\_actual')">👁️</span>

</div>

<label for="nueva\_contraseña">Nueva Contraseña:</label>

<div class="input-container">

<input type="password" id="nueva\_contraseña" name="nueva\_contraseña" required>

<span class="password-toggle" onclick="togglePassword('nueva\_contraseña')">👁️</span>

</div>

<button type="button" onclick="mostrarOpciones()">Configurar Contraseña</button>

<div id="prompt-section" class="prompt-section">

<label>Configuración de Contraseña:</label><br>

Longitud (mínimo 8): <input type="number" id="longitud" min="8" value="12"><br>

<input type="checkbox" id="mayusculas"> Mayúsculas

<input type="checkbox" id="minusculas" checked> Minúsculas

<input type="checkbox" id="numeros" checked> Números

<input type="checkbox" id="especiales"> Caracteres especiales

<button type="button" onclick="crearContrasena()">Generar Contraseña</button>

</div>

<div class="contrasena-generada" id="contrasena-generada">

<h3>Contraseña Generada:</h3>

<input type="text" id="contrasena-copy" readonly>

</div>

<!-- Agregar aquí los elementos para mostrar la evaluación de la contraseña -->

<div id="resultado-evaluacion" style="margin-top: 10px; color: #2c3e50;"></div>

<div id="recomendaciones" style="color: #e74c3c; margin-top: 5px;"></div>

<button id="enviar-correo" type="button" style="display: none;" onclick="enviarContrasenaPorCorreo()">Enviar Contraseña a Correo</button>

<button type="submit">Actualizar Contraseña</button>

</form>

<p>Regresar <a href="Iniciodesecion.php">Inicio de Seción</a></p>

<div class="message">

<?php

if (isset($\_SESSION['error\_message'])) {

echo "<p class='error'>" . $\_SESSION['error\_message'] . "</p>";

unset($\_SESSION['error\_message']);

}

if (isset($\_SESSION['success\_message'])) {

echo "<p class='success'>" . $\_SESSION['success\_message'] . "</p>";

unset($\_SESSION['success\_message']);

}

?>

</div>

</div>

</body>

</html>

enviar contraseña:

<?php

session\_start();

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\Exception;

require 'vendor/autoload.php'; // Asegúrate de que el autoload de Composer está incluido

header('Content-Type: application/json');

$data = json\_decode(file\_get\_contents('php://input'), true);

$email = $data['email'];

$contrasena = $data['contrasena'];

// Crear una instancia de PHPMailer

$mail = new PHPMailer(true);

try {

// Configuración del servidor SMTP

$mail->isSMTP();

$mail->Host = 'smtp.gmail.com'; // Cambia esto a tu servidor SMTP

$mail->SMTPAuth = true;

$mail->Username = 'dianayazminhernandezvargas@gmail.com'; // Tu email

$mail->Password = 'plwc tjfk somm ecnl'; // Tu contraseña del email o una contraseña de aplicación

$mail->SMTPSecure = 'tls';

$mail->Port = 587;

// Configuración del correo

$mail->setFrom('dianayazminhernandezvargas@gmail.com', 'Diana Hernandez');

$mail->addAddress($email);

$mail->isHTML(true);

$mail->Subject = 'Nueva Contraseña Generada';

$mail->Body = "Tu nueva contraseña generada es: <strong>$contrasena</strong>";

$mail->send();

echo json\_encode(['success' => true]);

} catch (Exception $e) {

echo json\_encode(['success' => false, 'error' => $mail->ErrorInfo]);

}

?>