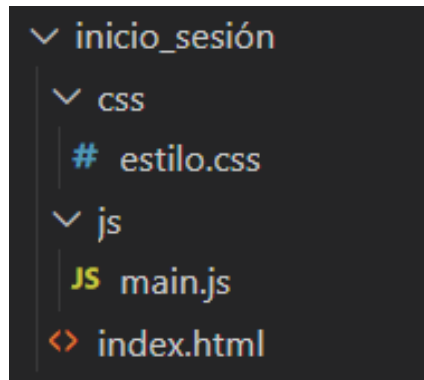


MANUAL DE USUARIO – INICIO DE SESIÓN

1. ESTRUCTURA DE ARCHIVOS

- Carpeta principal (inicio_sesión)
- Archivo index.html
- Subcarpeta CSS y dentro archivo estilo.css
- Subcarpeta JS y dentro archivo main.js



2. MAQUETADO HTML

• ESTRUCTURA BÁSICA

```
# estilo.css  index.html X
inicio_sesión > index.html > html > body
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>Registro e Inicio de Sesión</title>
7    <link rel="stylesheet" href="css/estilo.css">
8  </head>
9  <body>...
34 </body>
35 </html>
36
```

- <!DOCTYPE html>: Define el documento como HTML5.
- <html lang="es">: Idioma principal: español.
- <meta charset="UTF-8">: Soporta caracteres especiales (tildes, ñ).
- <meta name="viewport">: Adapta la pantalla a dispositivos móviles.
- <title>: Título de la pestaña del navegador.
- <link rel="stylesheet">: Vincula el archivo CSS (estilo.css).

- **FORMULARIO DE REGISTRO**

```
9 <body>
10 <!-- Formulario de Registro -->
11 <div class="container" id="registerContainer">
12   <h2>Registro</h2>
13   <form id="registerForm">
14     <input type="text" id="regUsername" placeholder="Usuario" required>
15     <input type="password" id="regPassword" placeholder="Contraseña" required>
16     <input type="password" id="regConfirmPassword" placeholder="Confirmar Contraseña" required>
17     <button type="submit">Registrarse</button>
18   </form>
19   <div class="toggle" id="toLogin">¿Ya tienes cuenta? Iniciar sesión</div>
20 </div>
21
```

- id="registerContainer": Contenedor principal del formulario de registro.
- Campos del formulario (registerForm):
 - regUsername: Campo de texto para nombre de usuario.
 - regPassword y regConfirmPassword: Campos de contraseña (con validación de coincidencia).
 - required: Obligatorio llenar estos campos.
- Botón "Registrarse": Envía el formulario.
- Enlace "Iniciar sesión" (id="toLogin"): Alterna entre formularios (funcionalidad en JS).

- **FORMULARIO DE INICIO DE SESIÓN**

```
21
22 <!-- Formulario de Inicio de Sesión -->
23 <div class="container" id="loginContainer" style="display: none;">
24   <h2>Inicio de Sesión</h2>
25   <form id="loginForm">
26     <input type="text" id="loginUsername" placeholder="Usuario" required>
27     <input type="password" id="loginPassword" placeholder="Contraseña" required>
28     <button type="submit">Ingresar</button>
29   </form>
30   <div class="toggle" id="toRegister">¿No tienes cuenta? Regístrate</div>
31 </div>
32
```

- style="display: none;": Oculta este formulario inicialmente.
- Campos:
 - loginUsername y loginPassword: Credenciales para iniciar sesión.
- Botón "Ingresar": Envía los datos.
- Enlace "Regístrate" (id="toRegister"): Muestra el formulario de registro.

- **VINCULAR JAVASCRIPT**

```
33 <script src="js/main.js"></script>
34 </body>
35 </html>
```

- <script src="js/main.js"></script>: Enlaza el archivo JavaScript ubicado en la carpeta js (se carga al final del <body> para mejor rendimiento).

3. DISEÑO CSS

- ESTILOS GENERALES

```
1
2      /* Estilos básicos para centrar
3  body {
4      background: #f2f2f2;
5      font-family: Arial, sans-serif;
6      display: flex;
7      justify-content: center;
8      align-items: center;
9      height: 100vh;
10     margin: 0;
11 }
12
```

- display: flex + justify-content/align-items: center: Centra el contenido vertical y horizontalmente.
- height: 100vh: Ocupa el 100% del alto de la pantalla.
- background: #f2f2f2: Color de fondo gris claro.

- CONTENEDOR DE FORMULARIOS

```
12
13     .container {
14         background: #fff;
15         width: 300px;
16         padding: 20px;
17         border-radius: 8px;
18         box-shadow: 0 2px 4px rgba(0,0,0,0.1);
19         margin: 10px;
20     }
21
```

- box-shadow: Sutil sombra para efecto de elevación.
- border-radius: 8px: Esquinas redondeadas.
- width: 300px: Ancho fijo para consistencia.

- TÍTULOS

```
21
22     h2 {
23         text-align: center;
24         margin-bottom: 20px;
25     }
26
```

- Text-align:center = centra el título
- Margin-bottom:20px = agrega espacio inferior

- CAMPOS DE ENTRADA (INPUT)

```
26
27   input[type="text"],
28   input[type="password"] {
29     width: 100%;
30     padding: 10px;
31     margin: 10px 0;
32     border: 1px solid #ccc;
33     border-radius: 4px;
34     box-sizing: border-box;
35   }
```

- width: 100%: Ocupa todo el ancho del contenedor.
- box-sizing: border-box: Incluye padding en el ancho total.
- border: 1px solid #ccc: Borde sutil para mejor visibilidad.

- BOTONES

```
36
37   button {
38     width: 100%;
39     padding: 10px;
40     background: #007BFF;
41     border: none;
42     border-radius: 4px;
43     color: #fff;
44     font-size: 16px;
45     cursor: pointer;
46   }
47
48   button:hover {
49     background: #0056b3;
50   }
```

- background: #007BFF: Color azul estándar.
- :hover: Oscurece el botón al pasar el mouse (feedback visual).
- cursor: pointer: Cambia el cursor a "mano" para indicar acción

- ENLACES DE ALTERNANCIA (.TOGGLE)

```
52   .toggle {
53     text-align: center;
54     margin-top: 10px;
55     cursor: pointer;
56     color: #007BFF;
57   }
```

- color: #007BFF: Coherencia con el color de los botones.
- cursor: pointer: Indica que es clickeable.

4. INTERACTIVIDAD JAVASCRIPT

- INICIALIZACIÓN Y REFERENCIAS A ELEMENTOS DEL DOM

```
# estilo.css JS main.js X
inicio_sesión > js > JS main.js > ...
1 document.addEventListener("DOMContentLoaded", function() {
2   // Referencias a contenedores y enlaces para alternar formularios
3   const registerContainer = document.getElementById('registerContainer');
4   const loginContainer = document.getElementById('loginContainer');
5   const toLogin = document.getElementById('toLogin');
6   const toRegister = document.getElementById('toRegister');
```

- Espera a que la página cargue completamente (DOMContentLoaded)
- Selecciona elementos HTML necesarios usando "getElementById"

- ALTERNAR ENTRE FORMULARIOS

```
7
8   // Cambio entre formularios
9   toLogin.addEventListener('click', function() {
10     registerContainer.style.display = 'none';
11     loginContainer.style.display = 'block';
12   });
13
14   toRegister.addEventListener('click', function() {
15     loginContainer.style.display = 'none';
16     registerContainer.style.display = 'block';
17   });
18
```

- Al hacer clic en "¿Ya tienes cuenta?": Oculta registro → Muestra login.
- Al hacer clic en "¿No tienes cuenta?": Oculta login → Muestra registro.

- FUNCIÓN PARA REGISTRO DE USUARIOS

```
19   // Función para registrar un usuario
20   const registerForm = document.getElementById('registerForm');
21   registerForm.addEventListener('submit', function(event) {
22     event.preventDefault();
```

- event.preventDefault(): Evita que el formulario se envíe de la manera tradicional (recargando la página).

```

23
24     const username = document.getElementById('regUsername').value.trim();
25     const password = document.getElementById('regPassword').value.trim();
26     const confirmPassword = document.getElementById('regConfirmPassword').value.trim();
27

```

- `document.getElementById('...').value`:
 - Obtiene el texto ingresado en los campos:
 - `regUsername`: Nombre de usuario.
 - `regPassword`: Contraseña.
 - `regConfirmPassword`: Confirmación de contraseña.
- `.trim()`: Elimina espacios en blanco al inicio/final del texto (para evitar errores).

```

27
28     // Validación de campos vacíos
29     if (username === '' || password === '' || confirmPassword === '') {
30         alert('Por favor, complete todos los campos.');
```

- Validación de Campos Vacíos:
 - Verifica si algún campo está vacío (`=== ''`).
 - Muestra un mensaje emergente al usuario (`alert()`) y detiene el proceso

```

34     // Verificar que las contraseñas coincidan
35     if (password !== confirmPassword) {
36         alert('Las contraseñas no coinciden.');
```

- Validación de Coincidencia de Contraseñas:
 - Condicional: Compara `password` y `confirmPassword`. Si son diferentes, muestra un error y detiene el proceso.

```

40     // Guardar credenciales en localStorage
41     localStorage.setItem('username', username);
42     localStorage.setItem('password', password);
43

```

- Almacenamiento en LocalStorage: Almacena datos en el navegador persistentemente (aunque se cierre la pestaña).

```

43
44     alert('Registro exitoso. Ahora puedes iniciar sesión.');
```

- Alerta de registro exitoso (alert()): notifica al usuario que el registro fue exitoso

```

45
46     // Limpiar campos y mostrar el formulario de inicio de sesión
47     registerForm.reset();
48     registerContainer.style.display = 'none';
49     loginContainer.style.display = 'block';
50 });
```

- Limpieza del formulario y cambio automático al formulario de Login:
 - .reset(): Vacía los campos del formulario (regUsername, regPassword, etc.).
 - Oculta el formulario de registro (registerContainer).
 - Muestra el formulario de inicio de sesión (loginContainer).
- Finalmente, la función completa para registrar un usuario, queda así:

```

19     // Función para registrar un usuario
20     const registerForm = document.getElementById('registerForm');
21     registerForm.addEventListener('submit', function(event) {
22         event.preventDefault();
23
24         const username = document.getElementById('regUsername').value.trim();
25         const password = document.getElementById('regPassword').value.trim();
26         const confirmPassword = document.getElementById('regConfirmPassword').value.trim();
27
28         // Validación de campos vacíos
29         if (username === '' || password === '' || confirmPassword === '') {
30             alert('Por favor, complete todos los campos.');
```

- FUNCIÓN PARA INICIAR SESIÓN

```
51
52 // Función para iniciar sesión
53 const loginForm = document.getElementById('loginForm');
54 loginForm.addEventListener('submit', function(event) {
55     event.preventDefault();
56
57     const username = document.getElementById('loginUsername').value.trim();
58     const password = document.getElementById('loginPassword').value.trim();
59
60     if (username === '' || password === '') {
61         alert('Por favor, complete todos los campos.');
```

- Inicialmente se hace el mismo proceso que con la función de registro: evitar que se recargue la página, obtener los valores del formulario y validar que los campos no estén vacíos.

```
64
65     const storedUsername = localStorage.getItem('username');
66     const storedPassword = localStorage.getItem('password');
```

- `localStorage.getItem()`: Recupera los valores almacenados durante el registro: ('username' y 'password').

```
67
68 // Validar credenciales
69 if (username === storedUsername && password === storedPassword) {
70     alert('Inicio de sesión exitoso.');
```

- condicional para comparar las credenciales:
 - si las credenciales coinciden, muestra una alerta con un mensaje de éxito.
 - si no coinciden, alerta con mensaje de error.

- Finalmente, la función completa de iniciar sesión queda así:

```
51
52 // Función para iniciar sesión
53 const loginForm = document.getElementById('loginForm');
54 loginForm.addEventListener('submit', function(event) {
55     event.preventDefault();
56
57     const username = document.getElementById('loginUsername').value.trim();
58     const password = document.getElementById('loginPassword').value.trim();
59
60     if (username === '' || password === '') {
61         alert('Por favor, complete todos los campos.');
```

```
62         return;
63     }
64
65     const storedUsername = localStorage.getItem('username');
66     const storedPassword = localStorage.getItem('password');
```

```
67
68     // Validar credenciales
69     if (username === storedUsername && password === storedPassword) {
70         alert('Inicio de sesión exitoso.');
```

```
71         // Aquí se puede redirigir a otra página o realizar otra acción
72     } else {
73         alert('Usuario o contraseña incorrectos.');
```

```
74     }
75 });
76 });
77
```

5. PRODUCTO FINAL

Registro

Usuario

Contraseña

Confirmar Contraseña

Registrarse

[¿Ya tienes cuenta? Iniciar sesión](#)