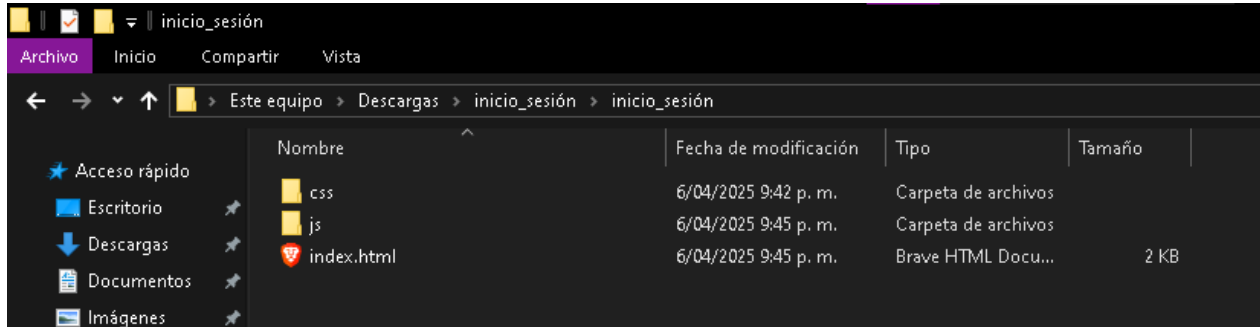


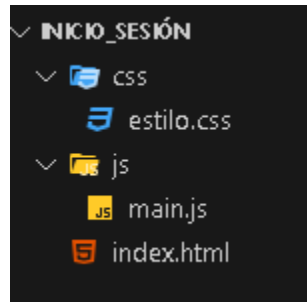
# Manual del programa de Inicio Sesión

## Hecho por: Juan José Ardila Castiblanco

1. Crear la estructura del proyecto.



2. Crear los archivos de cada parte del proyecto: index.html, styles.css y script.js.



3. Cambiamos el lenguaje del html y le colocamos un título.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Registro e Inicio de Sesión</title>
  <link rel="stylesheet" href="css/estilo.css">
</head>
```

4. Después comenzamos a escribir nuestra maqueta en el archivo HTML, que es la base de la estructura visual de la aplicación. Dentro de cada uno de los formularios, usamos la etiqueta <h2> para mostrar un título claro que indique la función del formulario. En el primer formulario se coloca el título "Registro", y en el segundo "Inicio de Sesión". Luego, insertamos una serie de campos de entrada para que el usuario pueda escribir su información.

```
<div class="container" id="registerContainer">
  <h2>Registro</h2>
  <form id="registerForm">
    <input type="text" id="regUsername" placeholder="Usuario" required>
    <input type="password" id="regPassword" placeholder="Contraseña" required>
    <input type="password" id="regConfirmPassword" placeholder="Confirmar Contraseña" required>
    <button type="submit">Registrarse</button>
  </form>
```

```
  <h2>Inicio de Sesión</h2>
  <form id="loginForm">
    <input type="text" id="loginUsername" placeholder="Usuario" required>
    <input type="password" id="loginPassword" placeholder="Contraseña" required>
    <button type="submit">Ingresar</button>
  </form>
```

5. Cada formulario incluye un botón que al hacer clic ejecuta la función correspondiente definida en JavaScript para registrar o iniciar sesión. Debajo de los formularios hay enlaces con clase "toggle" que permiten cambiar entre el registro y el inicio de sesión al hacer clic. Finalmente, todo está envuelto en <div> con clase "container" y id distintos.

```
<div class="container" id="registerContainer">
  <h2>Registro</h2>
  <form id="registerForm">
    <input type="text" id="regUsername" placeholder="Usuario" required>
    <input type="password" id="regPassword" placeholder="Contraseña" required>
    <input type="password" id="regConfirmPassword" placeholder="Confirmar Contraseña" required>
    <button type="submit">Registrarse</button>
  </form>
  <div class="toggle" id="toLogin">¿Ya tienes cuenta? Iniciar sesión</div>
</div>

<!-- Formulario de Inicio de Sesión -->
<div class="container" id="loginContainer" style="display: none;">
  <h2>Inicio de Sesión</h2>
  <form id="loginForm">
    <input type="text" id="loginUsername" placeholder="Usuario" required>
    <input type="password" id="loginPassword" placeholder="Contraseña" required>
    <button type="submit">Ingresar</button>
  </form>
  <div class="toggle" id="toRegister">¿No tienes cuenta? Regístrate</div>
</div>
```

6. Lo siguiente es linkear nuestro css donde el estilo aplicado al body centra todo el contenido en pantalla, además de aplicar un fondo gris claro y quitar los márgenes por defecto. Los formularios tienen un diseño limpio con fondo blanco, bordes redondeados, una sombra suave para dar profundidad, y un ancho fijo de 300px.

```
body {
  background: #f2f2f2;
  font-family: Arial, sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
}

.container {
  background: #fff;
  width: 300px;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);
  margin: 10px;
}

h2 {
  text-align: center;
  margin-bottom: 20px;
}
```

7. Los títulos se alinean al centro y tienen un margen inferior para separar visualmente el encabezado del resto del contenido. Los botones tienen un diseño atractivo con fondo azul, texto blanco y bordes redondeados. Además, incluyen un efecto hover que oscurece el color al pasar el cursor. Para terminar, el elemento con clase `.toggle`, que sirve como enlace para cambiar entre formularios, se muestra centrado, con color azul y cursor de tipo puntero para indicar que es clickeable.

**Registro**

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

8. En el Js el script comienza accediendo a los elementos principales del HTML, como los contenedores de los formularios y los botones que permiten cambiar entre registro e inicio de sesión. Luego, se programan los eventos para alternar entre ambos formularios: al hacer clic en "¿Ya tienes cuenta?" se oculta el formulario de registro y se muestra el de inicio, y viceversa.

```
document.addEventListener("DOMContentLoaded", function() {  
  // Referencias a contenedores y enlaces para alternar formularios  
  const registerContainer = document.getElementById('registerContainer');  
  const loginContainer = document.getElementById('loginContainer');  
  const toLogin = document.getElementById('toLogin');  
  const toRegister = document.getElementById('toRegister');  
  
  // Cambio entre formularios  
  toLogin.addEventListener('click', function() {  
    registerContainer.style.display = 'none';  
    loginContainer.style.display = 'block';  
  });  
  
  toRegister.addEventListener('click', function() {  
    loginContainer.style.display = 'none';  
    registerContainer.style.display = 'block';  
  });  
});
```

9. En la parte del registro, se captura el evento de envío del formulario. Primero, se validan los campos para asegurarse de que no estén vacíos y que las contraseñas coinciden. Si todo está correcto, se almacenan el nombre de usuario y la contraseña en el localStorage, se muestra un mensaje de éxito, y se limpia el formulario mostrando automáticamente el de inicio de sesión.

```
const username = document.getElementById('regUsername').value.trim();  
const password = document.getElementById('regPassword').value.trim();  
const confirmPassword = document.getElementById('regConfirmPassword').value.trim();  
  
// Validación de campos vacíos  
if (username === '' || password === '' || confirmPassword === '') {  
  alert('Por favor, complete todos los campos.');
```

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

```
  return;  
}  
  
// Guardar credenciales en localStorage  
localStorage.setItem('username', username);  
localStorage.setItem('password', password);  
  
alert('Registro exitoso. Ahora puedes iniciar sesión.');
```

10. Por último, en el formulario de inicio de sesión, se validan también los campos ingresados y se comparan con los valores guardados en localStorage. Si

coinciden, se muestra un mensaje indicando que el inicio fue exitoso. En caso contrario, se notifica que el usuario o la contraseña son incorrectos.

```
const loginForm = document.getElementById('loginForm');
loginForm.addEventListener('submit', function(event) {
  event.preventDefault();

  const username = document.getElementById('loginUsername').value.trim();
  const password = document.getElementById('loginPassword').value.trim();

  if (username === '' || password === '') {
    alert('Por favor, complete todos los campos.');
```

11. El resultado final debe ser una aplicación web permite a los usuarios registrarse con un nombre de usuario y contraseña, y posteriormente iniciar sesión con esos mismos datos. La información del usuario se almacena en el localStorage del navegador, por lo que no se necesita una base de datos externa.

The screenshot displays a web application interface. At the top, a dark notification box contains the text "127.0.0.1:5500 dice" and "Registro exitoso. Ahora puedes iniciar sesión.", with an "Aceptar" button. Below this is a white registration form titled "Registro". The form includes three input fields: the first contains the text "ii", and the other two contain a single dot ".". A blue "Registrarse" button is positioned below the inputs. At the bottom of the form, there is a link that reads "¿Ya tienes cuenta? Iniciar sesión".

127.0.0.1:5500 dice

Inicio de sesión exitoso.

Aceptar

## Inicio de Sesión

Ingresar

[¿No tienes cuenta? Regístrate](#)