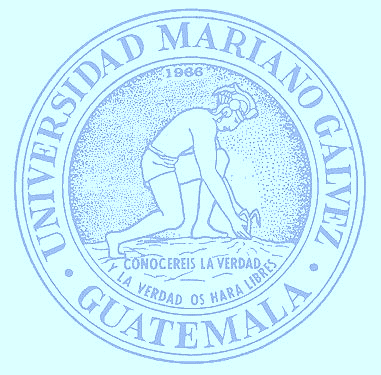
UNIVERSIDAD MARIANO GALVEZ DE GUATEMALA

INGENIERIA EN SISTEMAS DE LA INFORMACION Y CIENCIAS DE LA COMPUTACION

PROYECO FINAL

Sistema de Gestión de Reservas de Hotel

Brayan Ariel Ixcotoyac Lem Carnet: 9941-23-120

Raúl Antonio Morales Hernández Carnet: 9941 23 5977

Carlos Sebasthian Garcia Grajeda Carnet: 9941 23 11796

Guatemala, noviembre de 2024

**Tabla de Contenidos**

1. [Backend - Spring Boot](#backend---spring-boot)
2. Introduccion
3. Objetivos
   * [Estructura General](#estructura-general)
   * [Configuración de Archivos](#configuraci%C3%B3n-de-archivos)
   * [Explicación Detallada de Clases](#explicaci%C3%B3n-detallada-de-clases)
   * [Controladores](#controladores)
   * [Servicios](#servicios)
4. [Frontend - HTML, CSS, JavaScript](#frontend---html-css-javascript)
   * [Estructura General](#estructura-general-frontend)
   * [Páginas HTML](#p%C3%A1ginas-html)
   * [Estilos CSS](#estilos-css)
   * [JavaScript](#javascript)
5. [Conclusión](#conclusi%C3%B3n)

INTRODUCCION

Este proyecto es un sistema de gestión de usuarios que permite a las personas registrarse e iniciar sesión de manera segura. Para entenderlo, pensemos en cómo funcionan las cuentas de redes sociales o de correo electrónico. Cuando quieras entrar a tu cuenta, necesitas un nombre de usuario y una contraseña. Este sistema hace algo similar, permitiendo a los usuarios crear una cuenta y luego usar sus credenciales (nombre de usuario y contraseña) para acceder a su perfil.

**Tecnologías utilizadas**

Este sistema está construido usando dos partes principales:

1. **Backend (parte trasera del sistema)** : Este es el "cerebro" del sistema, que se encarga de manejar la lógica interna. Utiliza un framework llamado **Spring Boot** , que permite crear y gestionar funciones complejas de una manera organizada. Aquí es donde se almacenan los datos de los usuarios, como el nombre de usuario y la contraseña, en una base de datos.
2. **Frontend (parte frontal del sistema)** : Esta es la "cara" del sistema, la interfaz que el usuario ve y con la que interactúa, hecha con HTML, CSS y JavaScript. Por ejemplo, al abrir el sitio web, los usuarios verán botones y formularios donde pueden escribir su nombre de usuario y contraseña para iniciar sesión.

**¿Cómo funciona?**

El sistema funciona de la siguiente manera:

* **Registro** : Los usuarios pueden crear una cuenta introduciendo su nombre, usuario y contraseña. Esta información se guarda de forma segura en la base de datos.
* **Inicio de sesión** : Los usuarios pueden introducir su nombre de usuario y contraseña para ingresar a su cuenta. Si los datos coinciden con los guardados en la base de datos, el sistema les permite acceder.
* **Cierre de sesión** : Los usuarios pueden cerrar su sesión cuando terminan, para que otra persona no pueda acceder a su cuenta desde el mismo dispositivo.

Este proyecto cómo trabajar con las dos partes esenciales de cualquier aplicación web: el backend, que procesa los datos, y el frontend, que se muestra al usuario

Objetivos

**Objetivo general**

Desarrollar un sistema de gestión de usuarios que permita el registro e inicio de sesión de manera segura, utilizando tecnologías de backend y frontend que ofrezcan una estructura organizada y funcional para la autenticación básica de usuarios.

**Objetivos específicos**

1. **Implementar la funcionalidad de registro de usuarios** : Permitir que nuevos usuarios creen una cuenta proporcionando un nombre de usuario y contraseña. Esta información debe ser almacenada de forma segura en una base de datos.
2. **Desarrollar el sistema de inicio de sesión** : Crear una interfaz de inicio de sesión donde los usuarios puedan ingresar sus credenciales (nombre de usuario y contraseña) y verificar su identidad para acceder al sistema.
3. **Configurar el cierre de sesión seguro** : Permitir que los usuarios cierren sesión de forma segura para proteger sus datos personales y evitar el acceso no autorizado a sus cuentas.
4. **Proporcionar una interfaz intuitiva y amigable para el usuario** : Cree una interfaz frontend clara y sencilla, utilizando HTML, CSS y JavaScript, para que los usuarios puedan navegar fácilmente por el sistema.
5. **Implementar una base de datos para el almacenamiento de usuarios** : Configurar una base de datos que almacene de forma segura la información de los usuarios, permitiendo la recuperación de datos para las funciones de autenticación.
6. **Crear una documentación detallada del sistema** : Proporcionar una guía completa del proyecto que explica su estructura, el funcionamiento del código y las tecnologías utilizadas, dirigida a personas sin experiencia en programación para futuras revisiones y mantenimiento.

Backend - Spring Boot

# Estructura General

El backend de este proyecto se basa en el framework Spring Boot, el cual permite gestionar peticiones y manejar operaciones relacionadas con el usuario. Se han dividido las funcionalidades en varios paquetes y clases organizados de la siguiente manera:

plaintext

Copiar código

src/

├── main/

├── java/

├── com.example.demo/

├── controller/

├── service/

├── repository/

├── entity/

├── resources/

├── application.properties

# Configuración de Archivos

**application.properties**  
En este archivo se configura la conexión a la base de datos, así como otros ajustes de Spring Boot. Ejemplo de configuración:

* properties

**spring.datasource.url=jdbc:mysql://localhost:3306/usuarios\_db**

**spring.datasource.username=root**

**spring.datasource.password=root**

**spring.jpa.hibernate.ddl-auto=update**

# Explicación Detallada de Clases

**1. Entidad Usuario**

Define la estructura de datos del usuario, incluyendo atributos como nombre, usuario y contraseña:

* java

**@Entity**

**public class Usuario {**

**@Id**

**@GeneratedValue(strategy = GenerationType.IDENTITY)**

**private Long id;**

**private String nombre;**

**private String usuario;**

**private String contrasena;**

**}**

**2. Repositorio UsuarioRepository**

Extiende JpaRepository y se encarga de gestionar las operaciones CRUD para la entidad Usuario.

* java

**public interface UsuarioRepository extends JpaRepository<Usuario, Long> {**

**Optional<Usuario> findByUsuario(String usuario);**

**}**

**3. Servicio UsuarioService**

Contiene la lógica del negocio. Valida las credenciales y registra nuevos usuarios:

java

**@Service**

**public class UsuarioService {**

**public boolean validarCredenciales(String usuario, String contrasena) { ... }**

**}**

# Controladores

**UsuarioController**

Este controlador maneja las peticiones HTTP. Los principales endpoints son:

* java

**@RestController**

**public class UsuarioController {**

**@PostMapping("/login")**

**public ResponseEntity<?> login(@RequestBody LoginRequest request) { ... }**

**@PostMapping("/registro")**

**public ResponseEntity<?> registro(@RequestBody Usuario usuario) { ... }**

**}**

# Servicios

* **validarCredenciales**: Verifica las credenciales del usuario en el sistema.
* **registro**: Permite registrar nuevos usuarios en la base de datos.

Frontend - HTML, CSS, JavaScript

# Estructura General (Frontend)

La estructura del frontend es la siguiente:

plaintext

Copiar código

frontend\_project/

├── assets/

├── css/

├── Inicio.css

├── Inicio\_Sesion.css

├── images/

├── bg4.jpg

├── Icono.png

├── js/

├── Cerrar\_Sesion.js

├── Inicio.js

├── Inicio\_Sesion.js

├── Inicio.html

├── Inicio\_Sesion.html

# Páginas HTML

**Inicio.html**

Esta es la página de bienvenida:

* html

Copiar código

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<title>Inicio</title>

<link rel="stylesheet" href="assets/css/Inicio.css">

</head>

<body>

<header>

<img src="assets/images/Icono.png" alt="Icono">

<h1>Bienvenido a Nuestro Sitio</h1>

</header>

<main>

<a href="Inicio\_Sesion.html" class="btn">Iniciar Sesión</a>

</main>

<script src="assets/js/Inicio.js"></script>

</body>

</html>

**Inicio\_Sesion.html**

Página para que los usuarios inicien sesión:

* html

Copiar código

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<title>Inicio de Sesión</title>

<link rel="stylesheet" href="assets/css/Inicio\_Sesion.css">

</head>

<body>

<section class="login-container">

<form id="loginForm">

<label for="username">Usuario:</label>

<input type="text" id="username" required>

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

<input type="password" id="password" required>

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

</form>

<p id="error-message"></p>

</section>

<script src="assets/js/Inicio\_Sesion.js"></script>

</body>

</html>

# Estilos CSS

**Inicio.css**

Define el estilo de la página de inicio.

* css

Copiar código

body {

background-image: url('../images/bg4.jpg');

}

**Inicio\_Sesion.css**

Aplica el estilo del formulario de inicio de sesión.

* css

Copiar código

body {

background-color: #f0f0f0;

}

.login-container {

background: white;

}

# JavaScript

**Inicio.js**

Muestra un mensaje de bienvenida.

* javascript

Copiar código

document.addEventListener('DOMContentLoaded', function() {

console.log("Página cargada.");

});

**Inicio\_Sesion.js**

Valida los datos de inicio de sesión:

* javascript

Copiar código

document.getElementById('loginForm').addEventListener('submit', function(event) {

event.preventDefault();

if (username === "usuario" && password === "12345") {

window.location.href = "Inicio.html";

} else {

document.getElementById('error-message').innerText = "Usuario o contraseña incorrectos.";

}

});

**Cerrar\_Sesion.js**

Permite al usuario cerrar sesión y redirige a la página de inicio de sesión.

* javascript

Copiar código

function cerrarSesion() {

alert("Sesión cerrada correctamente.");

window.location.href = "Inicio\_Sesion.html";

}

Conclusión

Este sistema de gestión de usuarios proporciona una arquitectura simple pero efectiva para la autenticación básica de usuarios. La combinación de Spring Boot en el backend y HTML/CSS/JavaScript en el frontend crea una experiencia de usuario fluida y permite un desarrollo y mantenimiento organizado del sistema.