

**Departamento de Ciencias de la
Computación (DCCO)**

Carrera de Ingeniería de Software

Curso de Aplicaciones Distribuidas

Diseño de Seguridad con OAuth 2.0

Presentado por: Marlyn Almeida, Sebastian Bolaños,
Nicole Lara, Axel Pullaguari

Tutor: Morales, Dario.

Ciudad: Sangolquí, Ecuador

Fecha: 01/02/2025

Contenido

1. Introducción	3
2. Objetivo	3
3. Implementación de la Capa de Presentación	3
3.1. Estructura del Proyecto	3
3.2. Código Fuente Documentado	3
4. Manual de Usuario	4
4.1. Inicio de Sesión	4
4.2. Gestión de Empleados	4
4.3. Registro de Asistencia	4
4.4. Evaluación de Desempeño	5
5. Diagramas de Navegación	5
6. Conclusión	5

1. Introducción

La capa de presentación es el punto de interacción entre los usuarios y el sistema de gestión de recursos humanos. Se ha desarrollado utilizando **Angular** para proporcionar una interfaz intuitiva y funcional que facilite la gestión de empleados, asistencia y evaluaciones de desempeño. Este documento detalla el código fuente de la interfaz de usuario e incluye un manual de usuario con instrucciones claras para su utilización.

2. Objetivo

Implementar una interfaz de usuario accesible, intuitiva y eficiente que permita a los usuarios gestionar recursos humanos de manera efectiva, integrándose con los microservicios a través de APIs RESTful.

3. Implementación de la Capa de Presentación

La interfaz de usuario se ha desarrollado en Angular, utilizando Bootstrap para la estilización y HTTPClient para la comunicación con los microservicios.

3.1. Estructura del Proyecto

El proyecto sigue la estructura estándar de Angular:

```
Fronend/  
├── src/  
│   ├── app/  
│   │   ├── components/  
│   │   │   ├── login/  
│   │   │   ├── dashboard/  
│   │   │   ├── employees/  
│   │   │   ├── attendance/  
│   │   │   └── evaluations/  
│   │   ├── services/  
│   │   │   ├── auth.service.ts  
│   │   │   ├── employee.service.ts  
│   │   │   ├── attendance.service.ts  
│   │   │   └── evaluation.service.ts  
│   │   └── app.module.ts  
│   ├── assets/  
│   └── environments/  
├── angular.json  
└── package.json
```

3.2. Código Fuente Documentado

Ejemplo de código para la autenticación de usuarios:

```
/**  
 * Servicio de autenticación para gestionar el inicio y cierre de  
 sesión.  
 */  
@Injectable({ providedIn: 'root' })
```

```
export class AuthService {
  private apiUrl = 'https://api.example.com/auth';

  constructor(private http: HttpClient) {}

  /**
   * Inicia sesión con las credenciales del usuario.
   * @param credentials { username: string, password: string }
   */
  login(credentials: { username: string; password: string }):
  Observable<any> {
    return this.http.post(`${this.apiUrl}/login`, credentials);
  }
}
```

4. Manual de Usuario

4.1. Inicio de Sesión

1. Acceder a la página de inicio del sistema.
2. Ingresar las credenciales (usuario y contraseña).
3. Presionar el botón "Iniciar Sesión".
4. Si las credenciales son correctas, se redirige al panel de administración.

4.2. Gestión de Empleados

1. Navegar a la sección "Gestión de Empleados" desde el menú lateral.
2. Para agregar un empleado:
 - Hacer clic en "Agregar Empleado".
 - Llenar el formulario con los datos del empleado.
 - Presionar "Guardar".
3. Para editar un empleado:
 - Seleccionar un empleado de la lista.
 - Modificar los datos y presionar "Actualizar".
4. Para eliminar un empleado:
 - Seleccionar un empleado y hacer clic en "Eliminar".

4.3. Registro de Asistencia

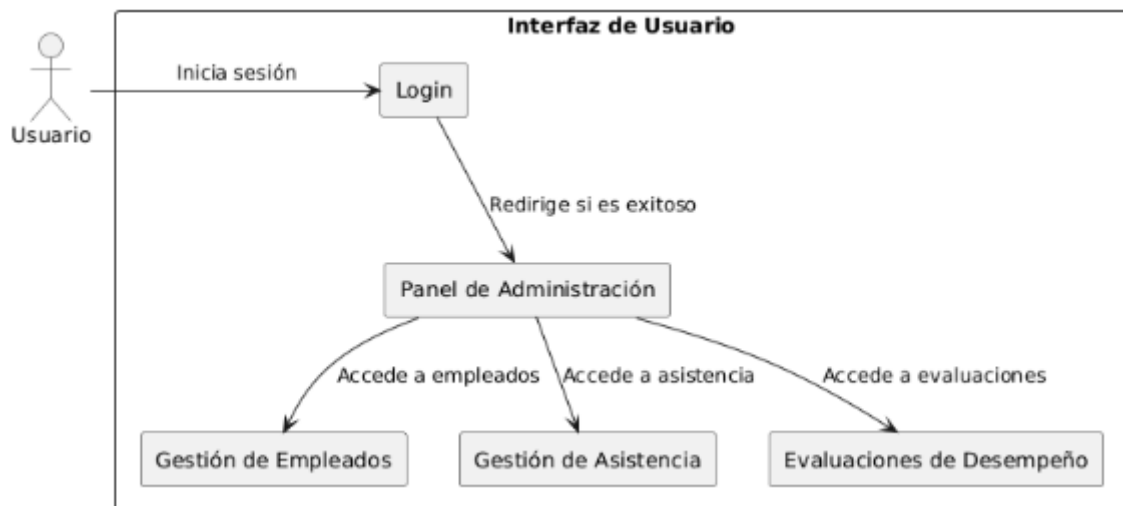
1. Ir a la pestaña "Asistencia" en el panel de administración.
2. Seleccionar un empleado y registrar su entrada/salida.
3. Verificar los registros en la tabla de asistencia.

4.4. Evaluación de Desempeño

1. Acceder a la sección "Evaluaciones".
2. Seleccionar un empleado para evaluar.
3. Llenar los criterios de evaluación y asignar una calificación.
4. Guardar la evaluación y generar un reporte.

5. Diagramas de Navegación

El siguiente diagrama representa la navegación dentro de la interfaz:



6. Conclusión

La interfaz de usuario ha sido implementada con una estructura modular en Angular, facilitando la navegación y la experiencia del usuario. La integración con los microservicios se ha logrado utilizando HTTPClient para consumir las APIs. El manual de usuario proporciona instrucciones detalladas para cada funcionalidad clave, garantizando un uso intuitivo y accesible del sistema.