

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

**Carrera de Ingeniería de Software**

**Curso de Aplicaciones Distribuidas**

Diseño de Microservicios

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

Tutor: Morales, Dario.

Ciudad: Sangolquí, Ecuador

Fecha: 01/02/2025

## Contenido

<b>1. Introducción.....</b>	<b>3</b>
<b>2. Objetivo .....</b>	<b>3</b>
<b>3. Especificaciones de API.....</b>	<b>3</b>
<b>3.1. API Gateway .....</b>	<b>3</b>
<b>3.1.1. Endpoints y Métodos HTTP.....</b>	<b>3</b>
<b>3.1.2. Ejemplo de Petición y Respuesta .....</b>	<b>3</b>
<b>3.2. Microservicio de Empleados .....</b>	<b>4</b>
<b>3.2.1. Endpoints y Métodos HTTP.....</b>	<b>4</b>
<b>3.2.2. Esquema de Datos.....</b>	<b>4</b>
<b>3.3. Microservicio de Asistencia.....</b>	<b>4</b>
<b>3.3.1. Endpoints y Métodos HTTP.....</b>	<b>4</b>
<b>3.3.2. Esquema de Datos.....</b>	<b>4</b>
<b>3.4. Microservicio de Evaluación de Desempeño .....</b>	<b>5</b>
<b>3.4.1. Endpoints y Métodos HTTP.....</b>	<b>5</b>
<b>3.4.2. Esquema de Datos.....</b>	<b>5</b>
<b>3.5. Microservicio de Usuarios y Roles.....</b>	<b>5</b>
<b>3.5.1. Endpoints y Métodos HTTP.....</b>	<b>5</b>
<b>3.5.2. Esquema de Datos.....</b>	<b>5</b>

# Diseño de Microservicios - Especificaciones de API

## 1. Introducción

El presente documento describe el diseño de los microservicios del sistema de gestión de recursos humanos. Cada microservicio está diseñado bajo la arquitectura RESTful, permitiendo una comunicación eficiente y desacoplada. Se presentan las especificaciones de las APIs, incluyendo los endpoints, métodos HTTP y esquemas de datos.

## 2. Objetivo

Definir la estructura de las APIs RESTful de los microservicios del sistema, especificando los puntos de acceso, operaciones permitidas y formatos de datos intercambiados. Esto garantizará una integración eficiente entre servicios y una interfaz clara para los clientes.

## 3. Especificaciones de API

### 3.1. API Gateway

#### 3.1.1. Endpoints y Métodos HTTP

Endpoint	Método	Descripción
/api/auth/login	POST	Autenticación de usuarios y generación de token JWT.
/api/auth/register	POST	Registro de nuevos usuarios.
/api/auth/validate	GET	Validación de token JWT.

#### 3.1.2. Ejemplo de Petición y Respuesta

Petición:

```
{
  "username": "admin",
  "password": "123456"
}
```

Respuesta:

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm91dCI6ImFkbWlnIiwiaWF0IjoxNjU0OTY0MDA0In0.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm91dCI6ImFkbWlnIiwiaWF0IjoxNjU0OTY0MDA0In0"
}
```

## 3.2. Microservicio de Empleados

### 3.2.1. Endpoints y Métodos HTTP

Endpoint	Método	Descripción
/api/employees	GET	Obtiene la lista de empleados.
/api/employees/{id}	GET	Obtiene los detalles de un empleado.
/api/employees	POST	Crea un nuevo empleado.
/api/employees/{id}	PUT	Actualiza los datos de un empleado.
/api/employees/{id}	DELETE	Elimina un empleado.

### 3.2.2. Esquema de Datos

```
{
  "id": 1,
  "nombre": "Juan Pérez",
  "correo": "juan.perez@example.com",
  "cargo": "Analista de Sistemas"
}
```

## 3.3. Microservicio de Asistencia

### 3.3.1. Endpoints y Métodos HTTP

Endpoint	Método	Descripción
/api/attendance	GET	Obtiene la lista de registros de asistencia.
/api/attendance/{id}	GET	Obtiene un registro de asistencia específico.
/api/attendance	POST	Registra una nueva asistencia.

### 3.3.2. Esquema de Datos

```
{
  "id": 10,
  "empleadoId": 1,
  "fecha": "2024-03-01",
  "horaEntrada": "08:00:00",
  "horaSalida": "17:00:00"
}
```

### 3.4. Microservicio de Evaluación de Desempeño

#### 3.4.1. Endpoints y Métodos HTTP

Endpoint	Método	Descripción
/api/evaluations	GET	Obtiene la lista de evaluaciones.
/api/evaluations/{id}	GET	Obtiene una evaluación específica.
/api/evaluations	POST	Registra una nueva evaluación.

#### 3.4.2. Esquema de Datos

```
{
  "id": 5,
  "empleadoId": 1,
  "fecha": "2024-03-01",
  "puntaje": 85,
  "comentarios": "Buen desempeño en el último trimestre."
}
```

### 3.5. Microservicio de Usuarios y Roles

#### 3.5.1. Endpoints y Métodos HTTP

Endpoint	Método	Descripción
/api/users	GET	Obtiene la lista de usuarios.
/api/users/{id}	GET	Obtiene los detalles de un usuario.
/api/users	POST	Crea un nuevo usuario.
/api/users/{id}	PUT	Actualiza un usuario.
/api/users/{id}	DELETE	Elimina un usuario.

#### 3.5.2. Esquema de Datos

```
{
  "id": 2,
  "username": "usuario123",
  "rol": "Administrador"
}
```