

## Plataforma de Gestión Empleados

### Descripción del Proyecto

La plataforma de gestión de empleados tiene como objetivo mejorar la eficiencia y la efectividad de la administración de recursos humanos dentro de una organización. Esta plataforma permite gestionar de manera integral la información de los empleados, desde su contratación hasta su salida de la empresa, pasando por evaluaciones de desempeño, formación, y gestión de beneficios.

- 

### 2. Diagrama de Arquitectura

El diagrama de arquitectura ilustra cómo se organizan los microservicios y cómo interactúan entre sí.

#### Microservicio de Gestión de Empleados:

- **Responsabilidad:** Maneja la información de los empleados (datos personales, historial laboral, etc.).
- **Interacción:** Se comunica con el microservicio de Roles y Permisos para asignar roles a los empleados.

#### 2. Microservicio de Roles y Permisos:

- **Responsabilidad:** Gestiona los roles y permisos de los empleados.
- **Interacción:** Se comunica con el microservicio de Gestión de Empleados para actualizar permisos.

### 3. Microservicio de Seguimiento del Rendimiento:

- **Responsabilidad:** Administra evaluaciones de rendimiento, objetivos y logros.
- **Interacción:** Consulta el microservicio de Gestión de Empleados para obtener la información relevante de los empleados.

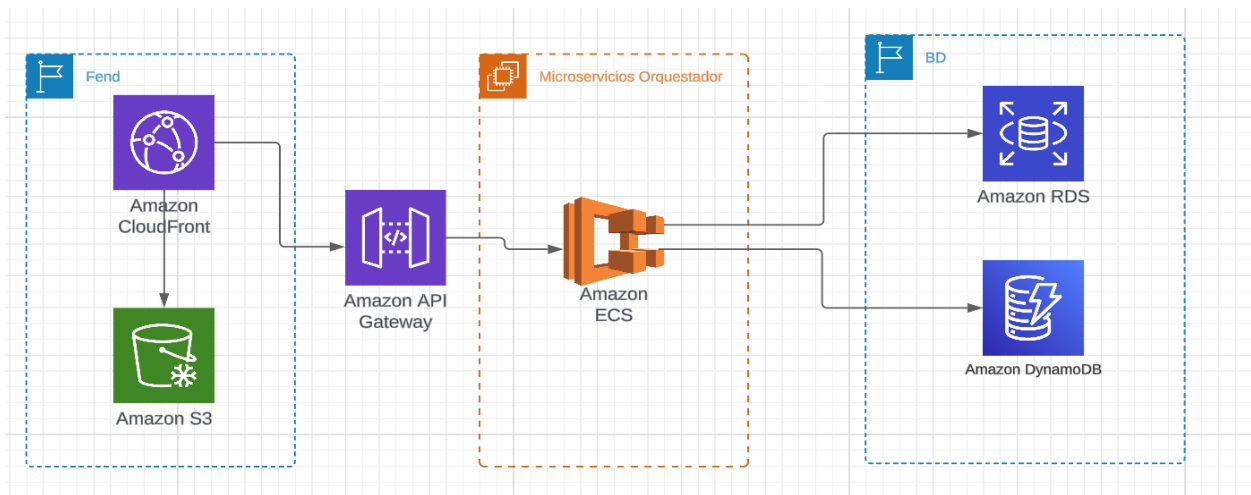
### 4. Microservicio de Administración de Nómina:

- **Responsabilidad:** Calcula y gestiona sueldos, deducciones y beneficios.
- **Interacción:** Obtiene información del microservicio de Gestión de Empleados para procesar la nómina.

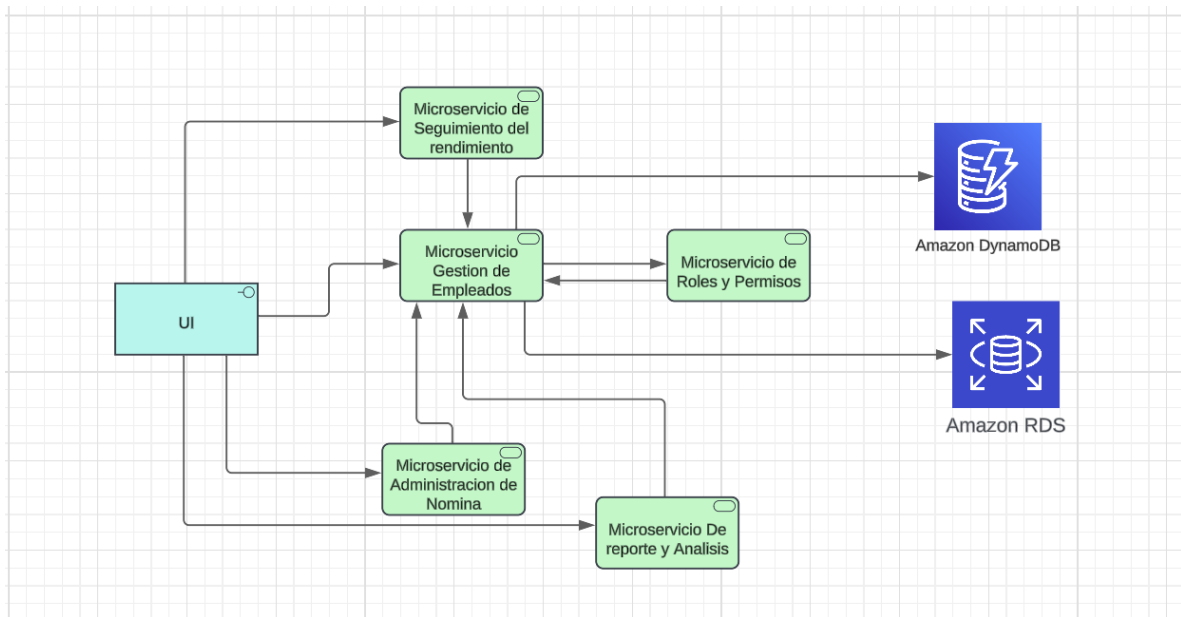
### 5. Microservicio de Reportes y Análisis:

- **Responsabilidad:** Genera reportes y análisis sobre empleados y su rendimiento.
- **Interacción:** Recolecta datos de los otros microservicios para generar reportes completos.

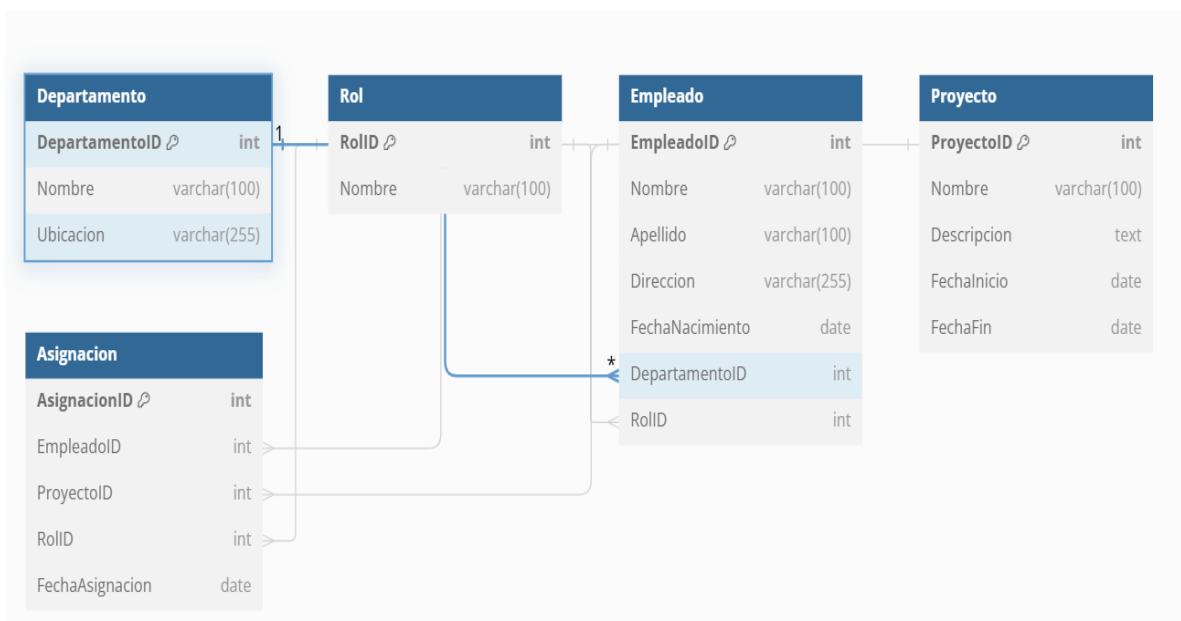
Arquitectura:



## Diseño de Microservicio



## Diagrama ER.



```

-- Crear tabla de Departamento
CREATE TABLE Departamento (
    DepartamentoID INT PRIMARY KEY,
    Nombre NVARCHAR(100) NOT NULL,
    Ubicacion NVARCHAR(255)
);

-- Crear tabla de Rol
CREATE TABLE Rol (
    RolID INT PRIMARY KEY,
    Nombre NVARCHAR(100) NOT NULL
);

-- Crear tabla de Empleado
CREATE TABLE Empleado (
    EmpleadoID INT PRIMARY KEY,
    Nombre NVARCHAR(100) NOT NULL,
    Apellido NVARCHAR(100) NOT NULL,
    Direccion NVARCHAR(255),
    FechaNacimiento DATE,
    DepartamentoID INT,
    RolID INT,
    FOREIGN KEY (DepartamentoID) REFERENCES Departamento(DepartamentoID),
    FOREIGN KEY (RolID) REFERENCES Rol(RolID)
);

-- Crear tabla de Proyecto
CREATE TABLE Proyecto (
    ProyectoID INT PRIMARY KEY,
    Nombre NVARCHAR(100) NOT NULL,
    Descripcion NVARCHAR(MAX),
    FechaInicio DATE,
    FechaFin DATE
);

-- Crear tabla de Asignacion
CREATE TABLE Asignacion (
    AsignacionID INT PRIMARY KEY,
    EmpleadoID INT,
    ProyectoID INT,
    RolID INT,
    FechaAsignacion DATE,
    FOREIGN KEY (EmpleadoID) REFERENCES Empleado(EmpleadoID),
    FOREIGN KEY (ProyectoID) REFERENCES Proyecto(ProyectoID),
    FOREIGN KEY (RolID) REFERENCES Rol(RolID)
);

```

Descripción de la solución:

### 1. Almacenamiento de Archivos Estáticos

- **Amazon S3:**
  - **Uso:** Almacenar archivos estáticos como imágenes, archivos CSS y JavaScript, así como cualquier otro contenido que no cambie frecuentemente.
  - **Configuración:**
    - Crear un bucket en Amazon S3.
    - Configurar políticas de acceso y permisos.
    - Habilitar el versionado si es necesario para mantener el control de versiones de los archivos.

### 2. Distribución de Contenido

- **AWS CloudFront:**
  - **Uso:** Distribuir el contenido estático almacenado en S3 a nivel global con baja latencia.
  - **Configuración:**
    - Crear una distribución en CloudFront.
    - Configurar el origen como el bucket de S3.
    - Establecer las configuraciones de caché, TTL y políticas de acceso.
    - Configurar el SSL/TLS para asegurar la transmisión de datos.

### 3. Exposición de API

- **Amazon API Gateway:**
  - **Uso:** Exponer y gestionar las API RESTful que permiten la interacción entre los clientes y los microservicios backend.
  - **Configuración:**
    - Crear API's en API Gateway.
    - Definir los recursos y métodos (GET, POST, PUT, DELETE).
    - Configurar el mapeo de rutas a los servicios backend que se ejecutan en Amazon ECS.

- Implementar seguridad y autenticación, como CORS y autenticación basada en tokens.

#### 4. Orquestación de Contenedores

- **Amazon ECS (Elastic Container Service):**
  - **Uso:** Ejecutar y gestionar los microservicios en contenedores Docker.
  - **Configuración:**
    - Crear un clúster en ECS.
    - Definir tareas y servicios en ECS.
    - Configurar el escalado automático y la supervisión de las tareas.
    - Establecer las redes y las configuraciones de seguridad para el acceso a los contenedores.

#### 5. Almacenamiento de Datos

- **Amazon RDS (Relational Database Service):**
  - **Uso:** Almacenar datos estructurados con alta disponibilidad y facilidad de administración.
  - **Configuración:**
    - Crear una instancia de base de datos en RDS ( MySQL).
    - Configurar las instancias para copias de seguridad automáticas, replicación y escalabilidad.
    - Ajustar los parámetros de rendimiento y las políticas de acceso.
- **Amazon DynamoDB:**
  - **Uso:** Almacenar datos no estructurados o datos que requieren una alta escalabilidad y baja latencia.
  - **Configuración:**
    - Crear tablas en DynamoDB con las claves de partición y clasificación adecuadas.
    - Configurar las políticas de escalado automático y la capacidad de lectura/escritura.
    - Configurar índices secundarios globales si se requiere una búsqueda adicional.

#### 6. Integración y Comunicación

- **Integración de Servicios:**

- **API Gateway** se comunica con **ECS** para enrutar las solicitudes de los clientes a los microservicios adecuados.
- **ECS** interactúa con **RDS** y **DynamoDB** para realizar operaciones de almacenamiento y recuperación de datos.
- **S3** se usa para servir archivos estáticos y almacenar datos de respaldo o archivos grandes.

## 7. Seguridad y Monitoreo

- **AWS IAM (Identity and Access Management):**

- **Uso:** Administrar permisos y acceso a los recursos de AWS.
- **Configuración:**
  - Crear políticas y roles para controlar el acceso a S3, CloudFront, API Gateway, ECS, RDS y DynamoDB.
  -

## Diagrama de Arquitectura

1. **Clientes:** Interactúan con la aplicación a través de **CloudFront**.
2. **CloudFront:** Distribuye contenido estático desde **S3** y enruta solicitudes de API a **API Gateway**.
3. **API Gateway:** Enruta las solicitudes a **ECS**.
4. **ECS:** Ejecuta microservicios y se comunica con **RDS** y **DynamoDB** para almacenar y recuperar datos.
5. **S3:** Almacena archivos estáticos y datos de respaldo.
6. **RDS y DynamoDB:** Almacenan datos estructurados y no estructurados.

Esta descripción y arquitectura proporcionan una visión clara de cómo cada componente se integra en tu solución basada en AWS.