# Universidad de San Carlos de Guatemala

Facultad de Ingenieria

Escuela de Ciencias y Sistemas

Julio Roberto Vasquez Santiago

200915080

# Proceso de Configuración e Implementación

**Paso 1: Preparar el Entorno**

* **Instalar Docker**: Asegúrate de tener Docker y Docker Compose instalados en el sistema.
* **Configurar MySQL**: Crea una base de datos y tablas en MySQL las cuales utilizaran los microservicios.
* CREATE TABLE IF NOT EXISTS empleados(  
   id INT AUTO\_INCREMENT PRIMARY KEY,  
   nombre VARCHAR(100) NOT NULL,  
   puesto VARCHAR(100),  
   salario DECIMAL(10, 2),  
   fecha\_contratacion DATE  
  );

**Paso 2: Crear los Microservicios**

* **Microservicio de Gestión de Empleados**:
  + **Responsabilidad**: Manejar la información de los empleados.
  + **Dependencias**: express, mysql2.
  + **Estructura**:
    - src/index.js - Configuración del servidor.
    - routes/empleadosRoutes.js - Rutas del microservicio.
    - controllers/empleadosController.js - Lógica del controlador.
    - models/empleadosModel.js - Acceso a la base de datos.
    - Config/db.js –Configuracion de conexión BD.

Texto

Descripción generada automáticamenteDocker File:

**Texto

Descripción generada automáticamenteConfiguracion de docker compose**:

Estructura de Carpetas:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Construir e iniciar contenedores:

* 1. docker-compose -f docker-compose-db.yml up --build
  2. docker-compose -f docker-compose-db.yml -f docker-compose-services.yml up --build
  3. **Contratos de Microservicios**

**1. Creación de Rol**

* **Método**: POST
* **URL**: http://localhost:3002/api/creacion-rol
* **Descripción**: Crea un nuevo rol en el sistema.

**Cuerpo de la Solicitud (Entrada):**

json

Copiar código

{

"nombre": "Nuevo Rol"

}

**Respuesta (Salida):**

* **201 Created**: El rol ha sido creado exitosamente.

json

Copiar código

{

"id": 1,

"nombre": "Nuevo Rol",

"message": "Rol creado exitosamente"

}

* **400 Bad Request**: Error en la solicitud (por ejemplo, falta el campo nombre).

json

Copiar código

{

"error": "El campo 'nombre' es obligatorio"

}

* **500 Internal Server Error**: Error en el servidor al procesar la solicitud.

json

Copiar código

{

"error": "Error al crear el rol"

}

**2. Consulta de Rendimiento**

* **Método**: GET
* **URL**: http://localhost:3001/api/consulta-rendimiento
* **Descripción**: Consulta las evaluaciones de rendimiento de los empleados.

**Parámetros de la Solicitud (Entrada):**

* No se requieren parámetros en la solicitud.

**Respuesta (Salida):**

* **200 OK**: Retorna la lista de evaluaciones de rendimiento.

json

Copiar código

{

"evaluaciones": [

{

"empleado\_id": 1,

"nombre": "Juan Pérez",

"rendimiento": "Excelente",

"fecha": "2024-08-25"

},

{

"empleado\_id": 2,

"nombre": "Ana Gómez",

"rendimiento": "Bueno",

"fecha": "2024-08-20"

}

]

}

* **500 Internal Server Error**: Error en el servidor al procesar la solicitud.

json

Copiar código

{

"error": "Error al consultar el rendimiento"

}

**3. Consulta de Empleado**

* **Método**: GET
* **URL**: http://localhost:3004/api/consulta-empleado
* **Descripción**: Consulta la información detallada de un empleado.

**Parámetros de la Solicitud (Entrada):**

* **Query Parameters**:
  + id (opcional): Identificador del empleado a consultar.

**Respuesta (Salida):**

* **200 OK**: Retorna la información detallada del empleado.

json

Copiar código

{

"empleado\_id": 1,

"nombre": "Juan Pérez",

"puesto": "Desarrollador",

"fecha\_ingreso": "2022-01-15"

}

* **404 Not Found**: Empleado no encontrado.

json

Copiar código

{

"error": "Empleado no encontrado"

}

* **500 Internal Server Error**: Error en el servidor al procesar la solicitud.

json

Copiar código

{

"error": "Error al consultar la información del empleado"

}

**4. Actualización de Fecha de Pago**

* **Método**: PUT
* **URL**: http://localhost:3003/api/actuliza-fpago/:id
* **Descripción**: Actualiza la fecha de pago de un empleado.

**Parámetros de la Solicitud (Entrada):**

* **URL Parameters**:
  + id (obligatorio): Identificador del empleado cuya fecha de pago será actualizada.

**Cuerpo de la Solicitud (Entrada):**

json

Copiar código

{

"fecha\_pago": "2024-08-25"

}

**Respuesta (Salida):**

* **200 OK**: La fecha de pago ha sido actualizada exitosamente.

json

Copiar código

{

"message": "Fecha de pago actualizada"

}

* **400 Bad Request**: Error en la solicitud (por ejemplo, falta el campo fecha\_pago).

json

Copiar código

{

"error": "El campo 'fecha\_pago' es obligatorio"

}

* **404 Not Found**: Empleado no encontrado.

json

Copiar código

{

"error": "Empleado no encontrado"

}

* **500 Internal Server Error**: Error en el servidor al procesar la solicitud.

json

Copiar código

{

"error": "Error al actualizar la fecha de pago"

}

**5. Eliminación de Reporte**

* **Método**: DELETE
* **URL**: http://localhost:3005/api/delete-reporte/:id
* **Descripción**: Elimina un reporte específico del sistema.

**Parámetros de la Solicitud (Entrada):**

* **URL Parameters**:
  + id (obligatorio): Identificador del reporte a eliminar.

**Respuesta (Salida):**

* **200 OK**: El reporte ha sido eliminado exitosamente.

json

Copiar código

{

"message": "Reporte eliminado exitosamente"

}

* **404 Not Found**: Reporte no encontrado.

json

Copiar código

{

"error": "Reporte no encontrado"

}

* **500 Internal Server Error**: Error en el servidor al procesar la solicitud.

json

Copiar código

{

"error": "Error al eliminar el reporte"

}

DockerFile de Microservicio:

Texto

Descripción generada automáticamente

Docker compose:

Mysql:

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamenteDocker compose Servicios:

Arquitectura:

Diagrama

Descripción generada automáticamente

ER:

