



Tema: Parcial 3 – REST RRHH

Docente: Ing. Carlos Boris Martínez Calzadia

Alumnos:

Ernesto José Barrientos Cedillos BC100520

Diego Alfredo Molina Medina MM101020

Módulo: DESARROLLO DE APLICACIONES WEB - GRUPO 01

Diagrama Base de datos – Entidad Relación

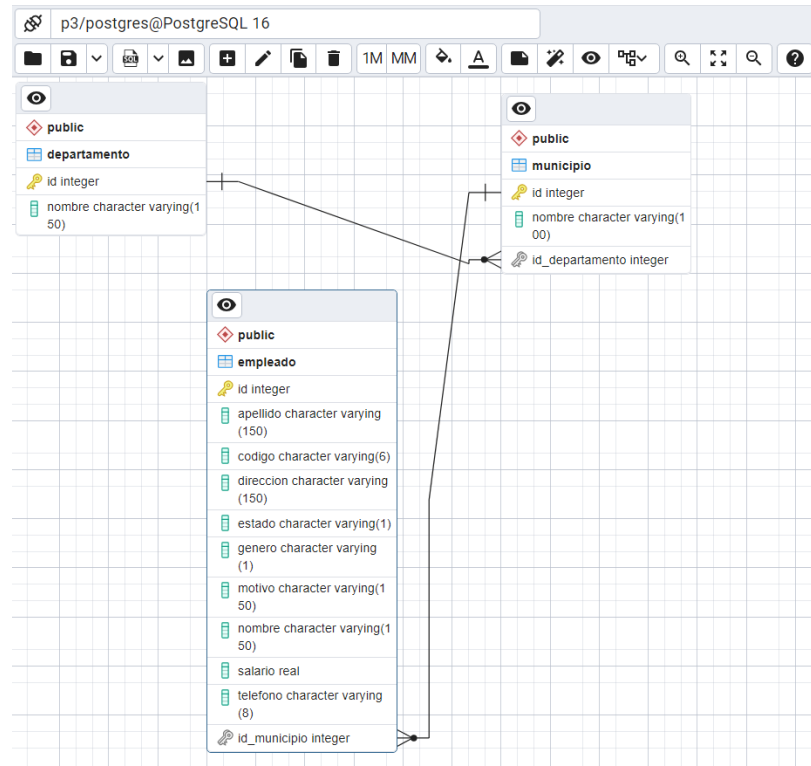


Tabla Departamento:

La tabla "Departamento" representa la entidad Departamento. La columna "id" se mapea a una columna en PostgreSQL con una clave primaria y se genera automáticamente utilizando "IDENTITY".

Tabla Empleado:

La tabla "Empleado" representa la entidad Empleado. La columna "id" se mapea a una columna en PostgreSQL con una clave primaria y se genera automáticamente utilizando "IDENTITY". La columna "id_municipio" se mapea como una clave foránea (@OneToOne) que hace referencia a la tabla "Municipio" a través de la columna "id".

Las demás columnas se mapean según sus respectivas anotaciones, incluyendo las restricciones de longitud y nulabilidad.

Tabla Municipio:

La tabla "Municipio" representa la entidad Municipio. La columna "id" se mapea a una columna en PostgreSQL con una clave primaria y se genera automáticamente utilizando "IDENTITY". La columna "id_departamento" se mapea como una clave foránea (@OneToOne) que hace referencia a la tabla "Departamento" a través de la columna "id".

En resumen, las relaciones se dan a través de claves foráneas en las tablas que se relacionan. En este caso, la tabla "Empleado" tiene una relación uno a uno con la tabla "Municipio", y ambas hacen referencia a la tabla "Departamento" a través de las claves foráneas "id_municipio" e "id_departamento" respectivamente. Por lo que la estructura de la base de datos sería: los empleados están asociados a municipios, y los municipios a su vez están asociados a departamentos.

Diagrama de estructura – Java Class

En IntelliJ IDEA, creamos un diagrama de clases para visualizar las relaciones y la estructura de nuestras clases Java. En general, se está utilizando una arquitectura basada en interfaces (ICRUD, IGenericRepo) para definir operaciones comunes y permitir la implementación en clases concretas (CRUDImpl, DepartamentoServiceImpl, EmpleadoServiceImpl, MunicipioServiceImpl, etc.).

También las clases están relacionadas con la gestión de entidades, como Departamento, Empleado, y Municipio, y se relacionan con servicios y controladores que proporcionan la funcionalidad relacionada con estas entidades.

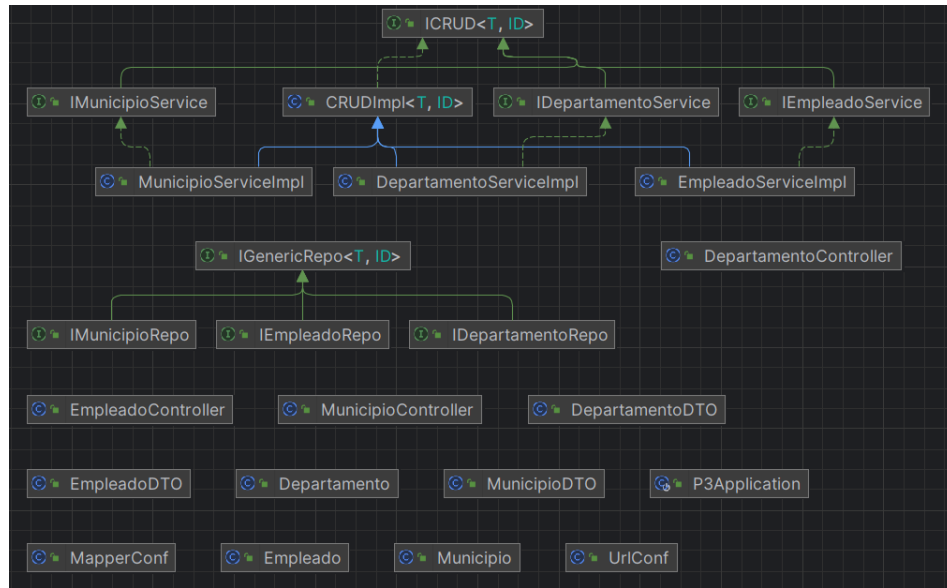
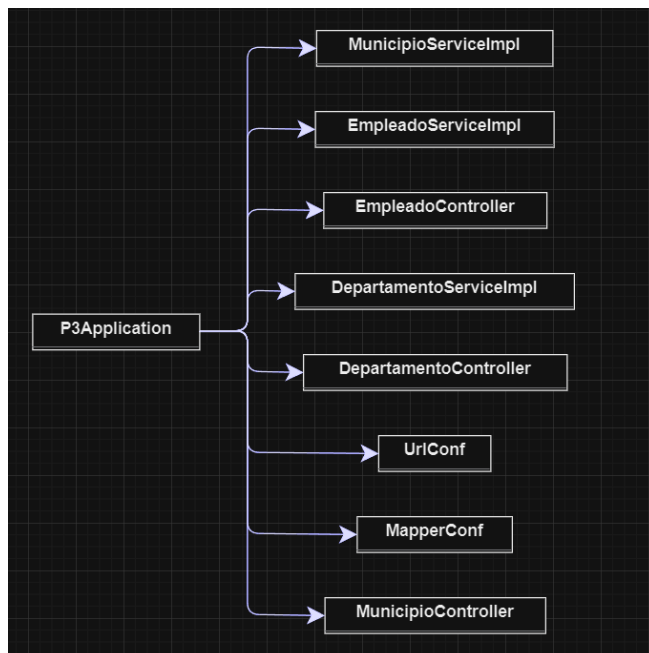


Diagrama de estructura – Spring Model Dependencies



El diagrama representa controladores (DepartamentoController, EmpleadoController, MunicipioController), servicios (DepartamentoServiceImpl, EmpleadoServiceImpl, MunicipioServiceImpl), una clase llamada MapperConf, una clase P3Application, y una clase UrlConf (por el tema de sensibilidad entre mayus y minúsculas).

Las flechas que se extienden desde la clase P3Application hacia los otros componentes indican que P3Application depende de estos componentes. En otras palabras, P3Application utiliza los servicios proporcionados por los otros componentes que se ven en la imagen.