# Migrations

Definimos estrutura do banco de dados, criamos tabelas, adicionar campos na tabela, adicionar chaves estrangeiras etc.

Criamos uma configuração do orm na estrutura de pasta:

\src\shared\infra\typeorm\index.ts

import { DataSource } from "typeorm";

export const AppDataSource = new DataSource({

    type: "postgres",

    host: "localhost",

    port: 5432,

    username: "diego",

    password: "vitor123123",

    database: "apiVendas",

*// synchronize: true,*

*// logging: true,*

    entities: [

        "./src/shared/infra/typeorm/entities/\*.ts"

    ],

*// subscribers: [],*

    migrations: [

        "./src/shared/infra/typeorm/migrations/\*.ts"

    ],

})

e importamos a configuração no server.ts

import { AppDataSource } from "../infra/typeorm";

AppDataSource.initialize()

    .then(async connect => {

    })

    .catch((error) => console.log("Error: ", error))

Adicionamos esse script no package json:

"typeorm": "ts-node-dev ./node\_modules/typeorm/cli.js";

Rodamos npm run typeorm para testar as migrations.

Para criarmos uma migration rodamos o comando:

npm run typeorm migration:create ./src/shared/infra/typeorm/migrations/CreateProduct

Para rodar as migrations e criar os campos da tabelas:

npm run typeorm -- -d ./src/shared/infra/typeorm/index.ts migration:run