**Comandos:**

Baixar o nestjs:

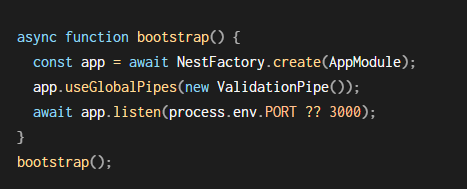
npm install -g @nestjs/cli

Criar um arquivo com o nest:

npm nest new task-management-api-2

Dependências para instalar:

npm install @nestjs/typeorm typeorm mysql2 (typeOrm)

npm i --save class-validator class-transformer (class validator) e dps colocar no main: app.useGlobalPipes(new ValidationPipe());

npm install uuid (biblioteca para gerar os uuid)

Criar Módulo de User:

nest generate module user

nest generate controller user

nest generate service user

Testar a aplicação:

npm run start:dev

(Ao executar este comando, o nestjs, por padrão, cria um servidor com a porta 3000, ou seja, a rota é http://localhost:3000.

**Explicações:**

O typescript é como um Javascript, mas com maior segurança. Utiliza tipos para indicar as variáveis e se algo for dar errado, ele avisa antes de iniciar o sistema.

Main.ts -> arquivo de entrada

O Module (modulo raiz) é onde são registrados os imports, os controllers e os providers.

No module é onde é feita a conexão com o banco.

Os imports são outros módulos/bibliotecas que a aplicação precisa;

Os Controllers é que lida com as requisições

E os Providers é que lida com as regras de negócio(service)

O controller recebe o que tem que fazer, indica qual é a função do service correspondente pra requisição e chama o service. O service faz as validações e faz as manipulações no banco de dados. Depois o Controller retorna a resposta.

O controller é o garçom (pega o pedido, leva até o chefe e depois traz o pedido ao cliente)

O service é a cozinha/chefe, faz o pedido que chega até ele, podendo ser pelo navegador ou pelo postman, etc.

O nest usa o TypeOrm para conversar com o banco de dados

O arquivo Entity é uma representação da tabela no banco de dados, é a classe user que tem as propriedades do usuário(atributos)

O @Body usado dentro do Controller, serve para quando vamos enviar dados dentro do corpo da requisição. O body(corpo da requisição) é como uma caixa onde fica guardado as informações em um Post ou Put, os quais são enviados ao Service

Passo a passo:

Fazer a importação do typeOrm

Criar o module de user

Criar o arquivo user.entity.ts (Entity de user) com os atributos do usuário

Conectar a Entity ao module

Métodos do CRUD – No service:

Save

Find

findOneBy

update

Delete