

Passo 1: Preparação inicial

Certifique-se de ter o Node.js e o npm instalados, já que essas ferramentas são essenciais para ambas as partes (back-end e front-end).

Passo 1: Abra o Terminal Abra o terminal no seu sistema Linux. Isso pode ser feito pressionando Ctrl + Alt + T ou procurando por "Terminal" no menu de aplicativos.

Passo 2: Atualize os Pacotes (Opcional, mas recomendado) Antes de instalar o Node.js, é uma boa prática atualizar os pacotes do sistema:

→ sudo apt update

→ sudo apt upgrade

Passo 3: Instale o Node.js usando o Node Version Manager (NVM) O NVM é uma ferramenta que permite instalar e gerenciar várias versões do Node.js. É uma maneira flexível e recomendada de instalar o Node.js no Linux.

3.1. Baixe e instale o NVM executando o seguinte comando:

→ curl -o- <https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh> | bash

Certifique-se de verificar o site oficial do NVM (<https://github.com/nvm-sh/nvm>) para a versão mais recente no comando acima.

3.2. Reinicie o terminal ou execute o seguinte comando para aplicar as mudanças:

source ~/.bashrc

3.3. Verifique se o NVM foi instalado corretamente:

nvm --version

Passo 4: Instale o Node.js usando o NVM Agora que você tem o NVM instalado, pode usar o NVM para instalar o Node.js.

4.1. Instale a versão mais recente do Node.js:

nvm install node

Isso instalará a versão mais recente do Node.js.

4.2. Verifique se o Node.js foi instalado corretamente:

node -v

4.3. Configure o NVM para usar a versão recém-instalada do Node.js como padrão:

nvm use node

Passo 5: Instale o npm (Node Package Manager) O npm é instalado automaticamente com o Node.js. No entanto, você pode verificar se ele está funcionando corretamente:

```
npm -v
```

Agora você deve ter o Node.js e o npm instalados e prontos para uso em seu sistema Linux. Lembre-se de que a versão dos comandos pode variar, então sempre verifique os sites oficiais do Node.js e do NVM para obter as informações mais atualizadas.

Passo 2: Configuração do Back-End com NestJS

2.1. Instale o Nest CLI globalmente:

```
npm install -g @nestjs/cli
```

2.2. Crie um novo projeto NestJS:

```
nest new backend-project
```

2.3. Navegue para a pasta do projeto:

```
cd backend-project
```

Passo 2.4: Configuração do Banco de Dados MongoDB

Para baixar o mongoDB em sua máquina basta acessar [Download MongoDB Community Server | MongoDB](#) e baixar a versão gratuita para a sua máquina e sistema operacional

Para saber a sua versão do linux use o comando:

```
lsb-release -a
```

Após instalar use o comando:

```
sudo systemctl status mongod
```

se tudo ocorreu perfeitamente vai aparecer o serviço do mongod so que ele vai estar inativo temos que ativar, use o comando:

```
sudo systemctl start mongod
```

Agora seu banco está funcionando, porém temos que instalar a parte gráfica para ajudar na usabilidade, acesse [Download MongoDB Community Server | MongoDB](#) novamente e baixe a interface gráfica MongoDB Compass (GUI)

Após isso abra o MongoDB Compass defina sua URI de acesso ao banco

No NestJS, você pode usar o Mongoose, uma biblioteca ODM (Object-Document Mapping), para interagir com o MongoDB de maneira mais conveniente. Vamos configurar o Mongoose para se conectar ao MongoDB no arquivo `app.module.ts`.

2.4.1. Instale o pacote mongoose usando o seguinte comando:

```
npm install mongoose
```

Instale os pacotes do Mongoose para fazer o acesso ao MongoDB:

```
npm i --save mongoose @nestjs/mongoose
```

2.4.2. Abra o arquivo `app.module.ts` em seu projeto NestJS e adicione as importações necessárias:

```
import { Module } from '@nestjs/common';
import { MongooseModule } from '@nestjs/mongoose';
```

2.4.3. No decorador `@Module`, adicione a configuração do Mongoose usando `MongooseModule.forRoot()`:

```
@Module({
  imports: [
    MongooseModule.forRoot('mongodb://localhost:port/nome-do-banco-de-dados', {
      useUrlParser: true,
      useUnifiedTopology: true,
    }),
    // Outros módulos
  ],
  controllers: [],
  providers: [],
})
export class AppModule {}
```

Substitua `'mongodb://localhost:port/nome-do-banco-de-dados'` pela URL de conexão do seu banco de dados MongoDB. Por exemplo, `'mongodb://localhost:port/mydatabase'`.

2.4.4. Para usar o Mongoose em seus módulos, você pode criar módulos separados para cada entidade e importar `MongooseModule.forFeature()` para definir os esquemas e modelos de dados. Por exemplo:

```
import { Module } from '@nestjs/common';
import { MongooseModule } from '@nestjs/mongoose';
import { CatsController } from './cats.controller';
import { CatsService } from './cats.service';
import { CatSchema } from './schemas/cat.schema';

@Module({
  imports: [
    MongooseModule.forFeature([{ name: 'Cat', schema: CatSchema }]),
  ],
  controllers: [CatsController],
  providers: [CatsService],
})
export class AppModule {}
```

```
  controllers: [CatsController],  
  providers: [CatsService],  
})  
export class CatsModule {}
```

Lembre-se de substituir CatSchema pelo seu próprio esquema MongoDB.

Passo 3: Configuração do Front-End com Vue.js

3.1. Instale o Vue CLI globalmente:

```
npm install -g @vue/cli
```

3.2. Crie um novo projeto Vue.js:

```
vue create frontend-project
```

Siga as instruções para configurar o projeto Vue.js de acordo com suas preferências.

Passo 4: Integração entre Back-End e Front-End

4.1. Navegue para a pasta do projeto Vue.js:

```
cd frontend-project
```

4.2. Instale o pacote axios para fazer solicitações HTTP ao back-end:

```
npm install axios
```

4.3. Crie as chamadas de API no Vue.js para se comunicar com o servidor NestJS. Por exemplo, você pode usar o Axios para fazer solicitações HTTP para as rotas do seu servidor.

Passo 5: Execução dos Projetos

5.1. Inicie o servidor NestJS:

```
npm run start:dev
```

5.2. Inicie o servidor de desenvolvimento Vue.js:

```
npm run serve
```

Agora, você deve ter o back-end do NestJS rodando em um servidor local e o front-end Vue.js sendo servido em outro. Lembre-se de que esta é uma configuração básica. À medida que você avança no desenvolvimento, pode precisar ajustar configurações e instalar mais dependências de acordo com os requisitos do seu projeto.