**NodeJS – Módulo 01 – B7Web**

**Material de estudo desenvolvido por:**

**Daniel Teixeira Quadros**

**Sumário**

[Aula 01 – Instalando o NODE 2](#__RefHeading___Toc360_3708390807)

[Aula 02 – Iniciando o Projeto (1/2) 2](#__RefHeading___Toc362_3708390807)

[Aula 03 – Iniciando o Projeto (2/2) 5](#__RefHeading___Toc364_3708390807)

[Aula 04 – Instalando o Mongo DB (Windows) 8](#__RefHeading___Toc366_3708390807)

# Aula 01 – Instalando o NODE

link:

https://nodejs.org/en/

comandos para verificar se o node e o npm estão instalados:

- Acesse o Prompt de Comandos e digite:

node -v → Para verificar a versão do node que está instalado.

npm -v → Para verificar a versão do npm instalado.

Estes são os aplicativos básicos iniciais para estudarmos e utilizarmos o NodeJS.

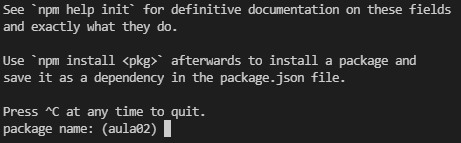
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# Aula 02 – Iniciando o Projeto (1/2)

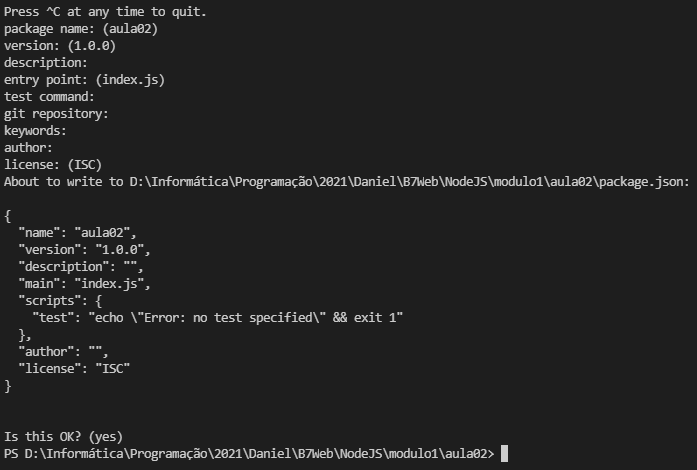
Através do terminal do VSCode acessamos a pasta onde iremos iniciar o nosso projeto. E nesta pasta então, executamos o comando para iniciar o projeto.

**1 - npm init** → Comando que inicia um projeto.

Resultado:



**2 -** Após este comando o sistema solicita que seja inserido as configurações básicas para o projeto, conforma o Print abaixo.



Obs.: Aqui na aula, apenas teclamos ‘Enter’ para todas as perguntas. De acordo com o Bonieky, nós preencheremos essas informações em outro momento.

**3** – Com o nosso projeto criado, obtemos o arquivo package.json. Veja o resultado no VSCode:

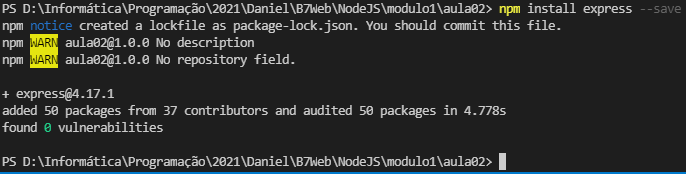


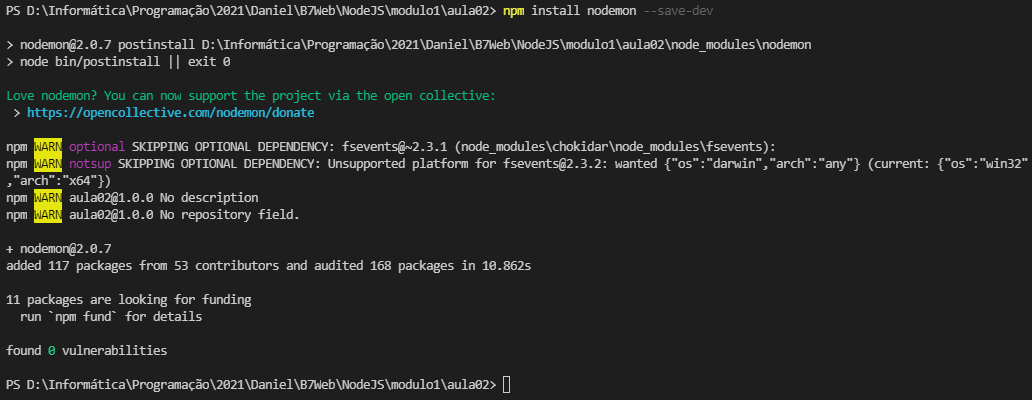
**4** – Abrindo o arquivo, temos os seguintes comandos iniciais:



5 – Nesta etapa, precisamos instalar duas dependências iniciais, para isto, voltamos a acessar o terminal do projeto:

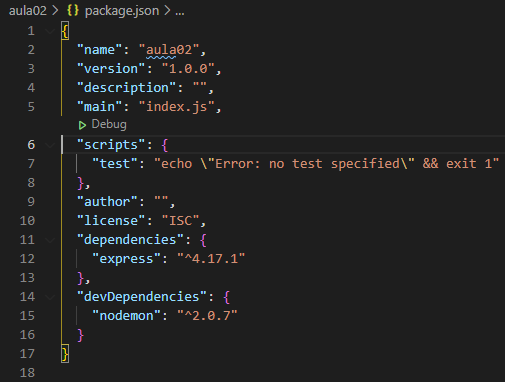
5.1 – **Express → npm install express --save (Express é uma biblioteca que auxilia no desenvolvimento).**



5.2 – **Nodemon → npm install nodemon --save-dev →** O -dev instala o nodemon nas dependências de desenvolvimento. Isto será relevante quando estivermos fazendo BUILDING na nossa aplicação para poder botar ela online, ou enviar para um servidor…

Obs.: O nodemon, reinicia o nosso servidor a cada momento em que atualizamos a nossa aplicação, facilitando o trabalho, nos momentos que testarmos o que foi alterado. Utilizando o nodemon não precisamos reiniciar o servidor manualmente.

Resultado no VSCode:



**6** – Acompanhando o Bonieky, faremos aqui algumas modificações no nosso arquivo package.json apenas para ele ficar mais conciso:

Vamos **retirar** a linha ‘**descrição**’, já que não utilizaremos.

Retiraremos a linha ‘**main**’, que não é essencial

O ‘**test**’ que está abaixo da linha script, também removemos

Retiraremos também o ‘**author**’ e a ‘**licence**’

Resultado no VSCode:



Obs: O item scripts será utilizado para receber comandos(atalhos) específicos que podemos rodar.

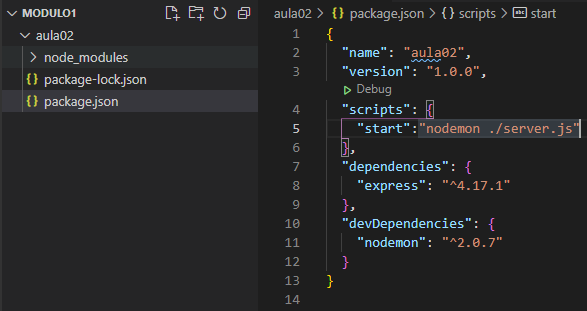
Vamos criar os seguintes atalhos:

**"start":"nodemon ./server.js" →** Atalho que iniciará o nosso servidor.

O comando **nodemon ./server.js** irá rodar, quando acessarmos o nosso terminal e digitarmos **npm start** ele irá executar o arquivo **server.js** utilizando a dependência nodemon

Aqui, neste momento, o arquivo **server.js** ainda não foi criado, mas criaremos logo em seguida.

Resultado no VSCode:



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# Aula 03 – Iniciando o Projeto (2/2)

Obs.: Continuamos utilizando os mesmos arquivos criados na aula2

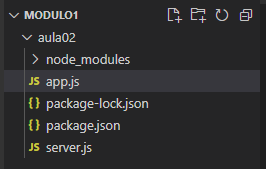
**7** – Aqui, vamos criar o nosso arquivo **server.js** em paralelo com o package.json no VSCode.

**8** – Neste arquivo, iremos puxar a nossa aplicação utilizando o Express (dependência que importamos anteriormente).

**Obs.: Logo em seguida, faremos ela rodar em uma porta específica e iniciar o servidor.**

**9** – Criamos outro arquivo, também paralelo com o package.json. O **app.js. →** o nome app.js é apenas para dar a ideia de aplicação.

**10** – No app.js, a primeira coisa que precisamos fazer é **importar o Express.**



**11** – Para importar o Express para o app.js utilizamos os seguintes comando:

const express = require('express')

**12** - Precisamos estabelecer as rotas:

const router = express.Router()

Estamos utilizando o Express para criar nossa primeira rota

**13** – Utilizando a variável ‘**router**’



Esta é a primeira rota, a **'/'** → raiz (home ou página inicial) do site, do sistema.

Criamos, após o caminho uma função anônima. Esta função tem dois parâmetros:

1º - **req** → requisição

2º - **res** → resposta

Obs.: Por enquanto, utilizaremos apenas a resposta.

Quando o usuário acessar a página inicial, o comando **res.send** enviará uma resposta

**14** – Configurações:

**const app = express() →** Inicia o aplicativo.

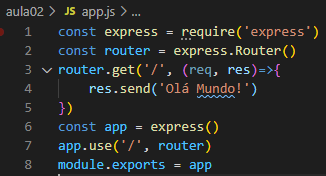
**app.use('/', router)** → Esta linha de comando enviará todas as rotas existentes. Até o momento, criamos apenas uma. O Bonieky não explicou claramente o **.use**, porém sobre esta linha, por enquanto isto foi o que ele explicou sobre o que ela faz.

**module.exports = app** → Esta linha é responsável por exportar o app.js.

**Obs.: Precisamos exportar o app.js, porque importaremos ele lá no server.js.**

**INICIALMENTE, PARA A NOSSA APLICAÇÃO FUNCIONAR, JÁ TEMOS O MÍNIMO NECESSÁRIO NO NOSSO app.js.**

App.js fica assim:



**AGORA PASSAREMOS PARA AS CONFIGURAÇÕES REALIZADAS NO ARQUIVO server.js.**

**const app = require('./app') →** Esta linha que está importando o app.js para cá.

**app.set('port', 7777)** → Nesta linha, estamos configurando uma porta para o nosso servidor, que aqui, estamos utilizando, juntamente com o Bonieky a porta '7777'.

**const server = app.listen(app.get('port'), ()=>{**

**console.log("Servidor rodando na porta"+server.address().port)**

**}) →** Esta linha é que realmente inicia o nosso servidor. Com o comando **listen**, ele passa a observar (ouvir) uma determinada porta (que no nosso caso é a '7777') e utilizamos o **app.get('port')** para puxar da própria porta que configuramos.

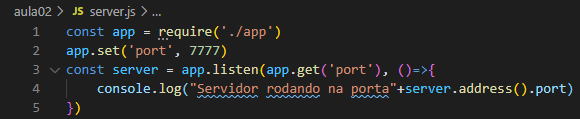
**Obs.: Mais para frente, faremos com que o comando 'app.set('port', 7777)' fique dinâmico, por isso utilizamoso 'app.get('port')' aqui.**

**Existe um segundo parâmetro** que podemos configurar no app.get que nos auxilia a mostrar no terminal que o servidor está rodando e em qual porta. Isto é utilizado para que o usuário possa ver estas informações e facilitar o seu trabalho.

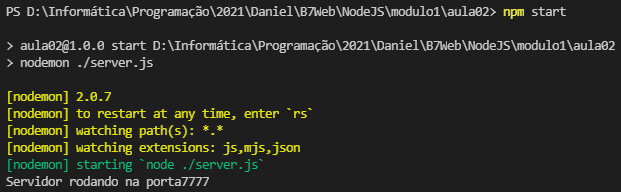
Para isto, criamos uma função anônima no segundo parâmetro e para o usuário poder visualizar estas informações, adicionamos no console.log o comando '**server.address().port**' → Esta função pega a porta em que o servidor está rodando.

**INICIALMENTE, PARA A NOSSA APLICAÇÃO FUNCIONAR, JÁ TEMOS O MÍNIMO NECESSÁRIO NO NOSSO server.js.**

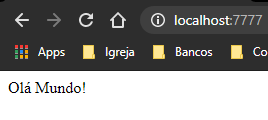
O server.js ficou assim:



A partir daqui, já temos o nosso servidor criado aqui no 'server.js' e a nossa aplicação criada no 'app.js'. Agora **para dar início** (subir) o projeto, vamos no terminal e executamos o comando **npm start.**



E, ao digitarmos no navegardor o comando **localhost:7777** (localhost → máquina local, 7777 → porta que configuramos) conseguimos ver a nossa aplicação rodando:



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# Aula 04 – Instalando o Mongo DB (Windows)

link:

https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/

Procurar o link MongoDB Community Edition e Mongo DB Download Center.

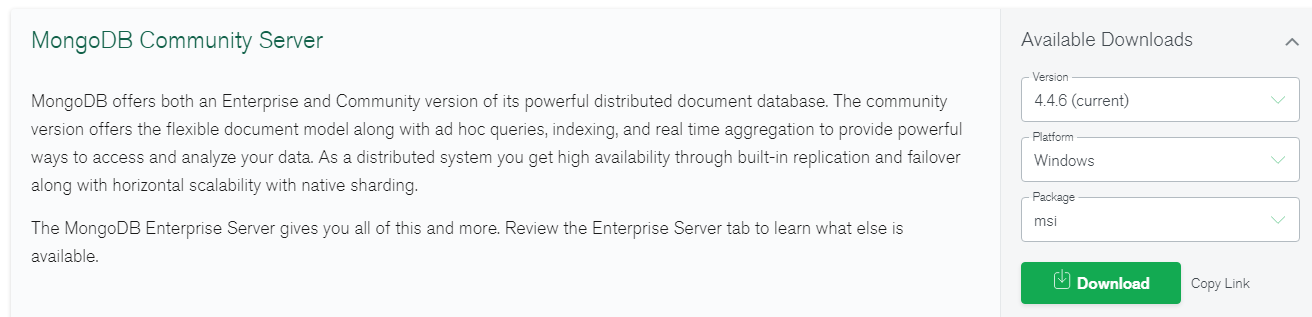
MongoDB

No link que abrir, precisamos escolher o MogoDB Community Server

- Version.

- Plataform

- Package

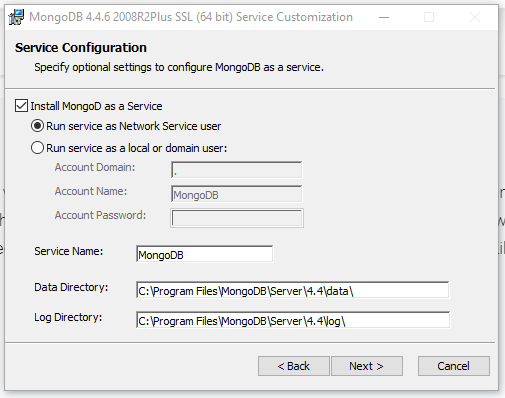


Durante a instalação do MongoDB, escolhemos a versão completa.

Nesta tela, existe a opção de não deixarmos o MongoDB rodando sempre, desde que o computador iniciar, ou a opção de iniciarmos o MongoDB manualmente:

1 – Se quisermos iniciar o MongoDB manualmente, precisamos desmarcar a opção ‘Install MongoD as a Service’.

Obs.: Aqui, a recomendação do Bonieky é que deixemos marcado esta opção, para não termos que ficar preocupados se o servidor está rodando ou não.



O Restante das configurações deixamos como está.

Obs.: É muito importante que anotemos em algum lugar de fácil acesso e visualização os dois diretórios que aparecem aqui:

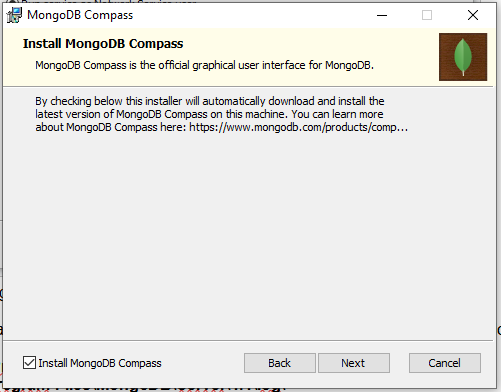
**Data Directory: C:\Program Files\MongoDB\Server\4.4\data\**

**Log Directory: C:\Program Files\MongoDB\Server\4.4\log\**

O Log Directory é onde serão armazenados todos os logs do sistema, é onde poderemos encontrar algum erro que venha acontecer, lista de acesso, bem como também, todos os outros eventos.

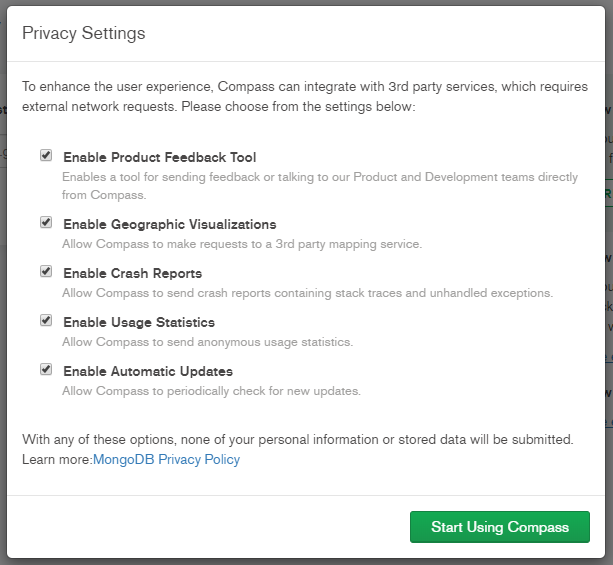
Data Directory é onde efetivamente ficarão os bancos de dados dos sistemas que iremos criar.

Na próxima tela deixamos marcado o Compass, que é a applicação visual do banco de dados



Após a instalação, o Compass irá iniciar pela primeira vez. Então aparecerá a tela abaixo:

Aqui, podemos deixar todas as opções marcadas.

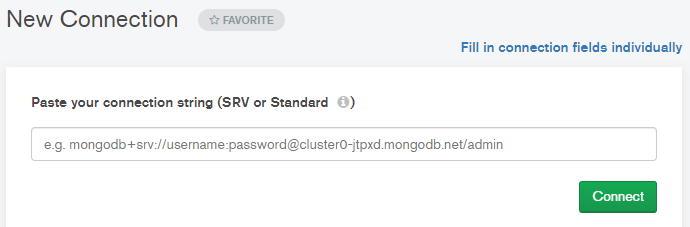


Uma vez instalado, ele já está rodando o Node no Computador. Podemos realizar o primeiro teste:

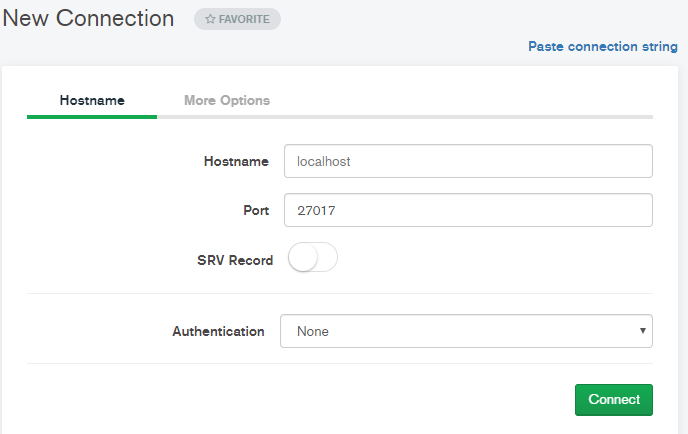
Obs.: É importante saber, que apesar de termos instalado esta interface gráfica do Mongo, também, se quisermos, podemos utilizá-lo no terminal.

Ao iniciarmos o MongoDB Compass, há uma pequena diferença entre a versão que ele ensinou no curso.

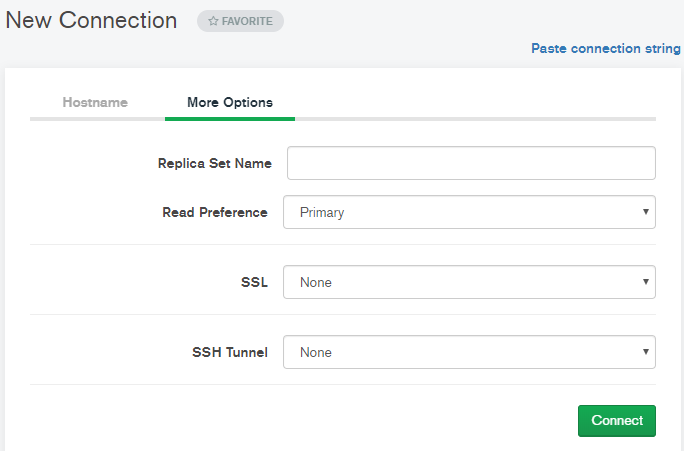
Ao invés de iniciar com a tela de configurações iniciais, a primeira tela que aparece tem apenas o campo para inserirmos a nossa **connection String**. Para irmos para as configurações, precisamos clicar em **Fill in connection fields individually**



Ao clicarmos em ‘Fill in connection fields individually’, Temos duas guias com todas as configurações:



e



- Agora vamos conectar em um **host específico**

Hostname: **localhost** → nosso computador

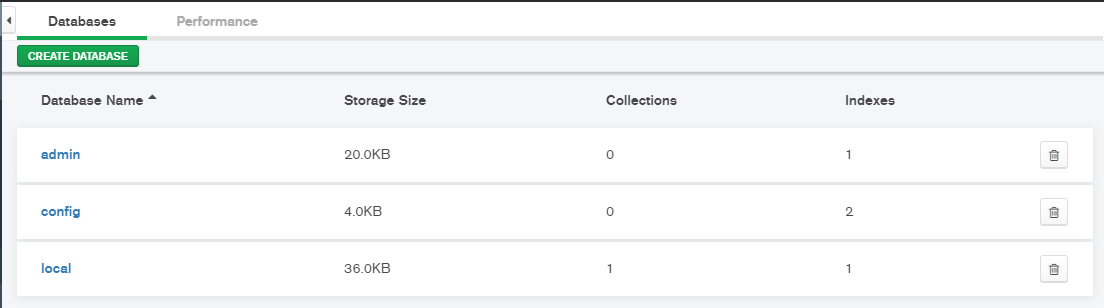
Port: **27017**

Authentication: Não precisa de autenticação, pois não estamos rodando com nenhum serviço aqui.

Apenas isto. Se guisermos salvar, podemos clicar no link ‘Favorite’ e darmos um nome a ele.

Por último, clicamos em ‘Connect’.

Automaticamente ele já cria os três servidores abaixo:



Obs.: Se precisarmos desconectar o servidor para conectarmos outro, basta clicarmos em ‘Connect’ no canto esquerdo superior da tela e Desconectarmos o servidor ativo.

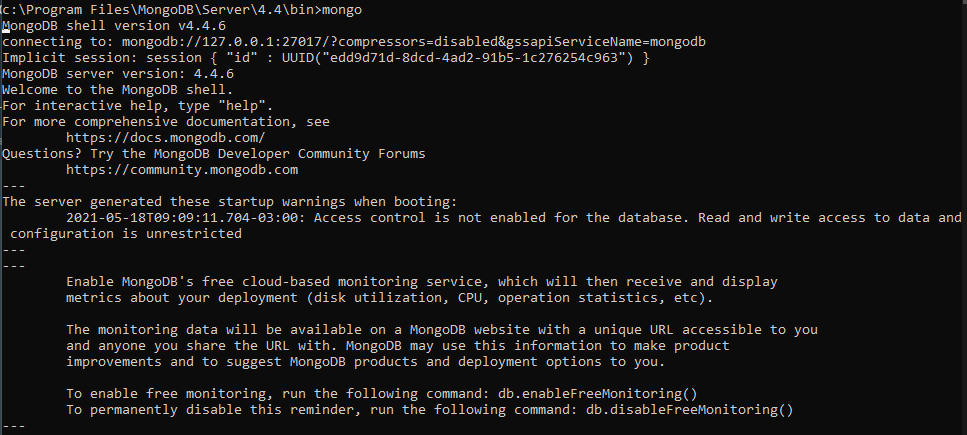
**Veremos agora, como realizamos a conecção do servidor, via Prompt de comando, se necessário:**

No Prompt de comando, executamos o comando ‘**mongo**’ dentro do seguinte diretório:



Obs. Aqui, a pasta **4.4** pode mudar de nome, de acordo com a versão instalada.

Ao executarmos, temos o seguinte resultado:



Para fechar o mongo aqui, precionamos as teclas de atalho ‘CTRL+C’

