

# CRIAÇÃO DE UM SERVIDOR



## 1. Criar o arquivo server.js na raiz do projeto

```
require('dotenv').config() //invoca o dotenv
const app = require("./src/app") //invoca app
const PORT = process.env.PORT //traz o valor de port do arquivo .env
app.listen(PORT) //define a porta do servidor
```

## 2. Executa no terminal o comando npm init - y:

Inicia o projeto e cria o arquivo package.json(descreve o projeto).

## 3. Executa no terminal o comando npm i express

Instala o express/dependências.

## 4. Executa no terminal o comando npm i nodemon

Instala o nodemon/reinicia o servidor a cada alteração sem precisar matá-lo.

## 5. Executa no terminal o comando npm i cors

Executa funções que está entre um pedido HTTP e a resposta final que o servidor envia de volta para o Client.

## 6. Modificar a linha script-start de node para nodemon

## 7. Executa no terminal o comando npm i dotenv

Instala o dotenv/muda o ambiente para que o servidor entender.

## 8. Executa no terminal o comando npm i mongoose

Instala o mongoose/faz a comunicação do back com o banco de dados

## 9. Cria o arquivo .gitignore na raiz do projeto

Ignora os arquivos, pastas e/ou senhas do projeto. Ignorar o node\_modules

## 10. Cria o arquivo .env na raiz do projeto

Guarda variáveis para ser usado no projeto.

Insere a URI gerada no mongoDb //Como fazer:

### Passo a passo do MongoDB

#### 1. Quick Start (Começo rápido)

Cria usuário e senha.

Define e adiciona IP para acesso. Para acessar de qualquer lugar: 0.0.0.0/0

#### 2. Network Access (Acesso à rede)

Lista de IP's para acesso.

#### 3. Database Access (Acesso ao banco de dados)

Lista dos usuários com acesso ao banco de dados.

#### 4. Database (Banco de dados)

Cria o banco de dados

I. Create ou crio

II. Shared (aws, virginia, nome do cluster())

#### 5. Após criar o banco de dados

I. Clica em Connect (conectar)

II. Clica em Connect your application (Conecte ao seu aplicativo)

III. Copia o link , por exemplo:

mongodb+srv://admin:<password>@projeto-clinica.4xyp0gq.mongodb.net/?retryWrites=true&w=majority  
e insere no .env

**Exemplo do .env:**

**MONGODB\_URI="link do mongo"**

**PORT=numerodaporta**

## 11. Inserir o arquivo .env no gitignore

## 12. Cria a pasta SRC e dentro cria:

## 13. A pasta database e dentro cria:

## 14. O arquivo mongoConfig.js

```
const mongoose = require('mongoose'); //invoca o mongoose
const uri = process.env.MONGODB_URI //cria a URI com usuário e senha
const connect = async () => { //cria a função connect
  try {
    await mongoose.connect(uri, {
      useNewUrlParser: true,
      useUnifiedTopology: true})
    console.log("Banco de dados conectado!")
  } catch (error) {
    console.log(error.message)}
}
module.exports = {connect}; //exporta a função connect
```

## 15. A pasta controllers (na SRC)

## 16. A pasta routes (na SRC)

Armazenará as rotas do projeto

## 17. A pasta models (na SRC)

## 18. O arquivo app.js (na SRC)

```
require('dotenv').config() // invoca o dotenv
const express = require("express") //invoca o express
const cors = require("cors") //invoca o cors
const app = express() // executa o express
const database = require('./database/mongoConfig') //invoca o arquivo mongoConfig da database
app.use(express.json()) // transformar tudo em json *
app.use(cors()) //executar cors
database.connect() //executando a função connect do arquivo mongoConfig
module.exports = app //exporta o arquivo app.js
```