Aula Programação Mobile – EC 10

REACT NATIVE

Instalar o Node e o NPM  
<https://nodejs.org/pt-br/blog/release/v18.12.0>

Visual studio code:

<https://code.visualstudio.com/download>

Android Studio:

<https://developer.android.com/studio?hl=pt-br>

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

Tela de celular com publicação numa rede social

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Configurar o expo dentro do Visual Studio Code

* npm install -g expo-cli (uma única vez para todos os projetos)

Para cada novo projeto criado:

* **Npx expo init** nome do seu projeto

Caso receba a mensagem: “expo init is not supported in the local CLI, please use npx create-expo-app instead”

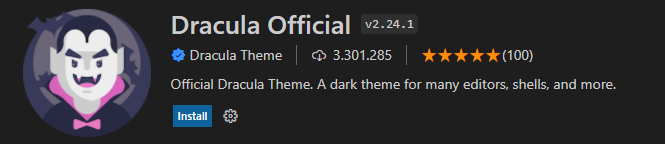
Execute assim:

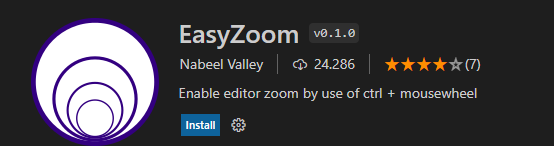
* npx create-expo-app e confirme a adicão do pacote solicitado.
* **Npx expo start** dentro da pasta do seu projeto para iniciá-lo.

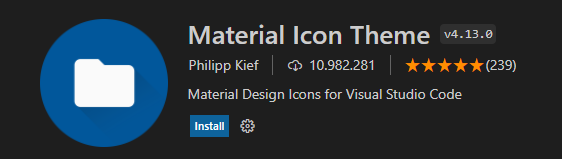
Configurações para ambiente FESA

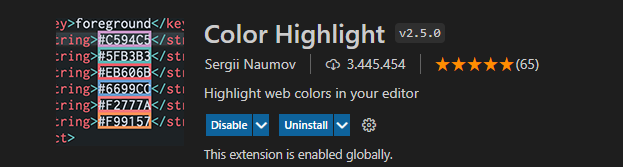
* Caso encontre problemas para executar o expo, tente especificar o path completo:
  + c:\users\pro6761\appdata\roaming\npm\expo
* Caso tenha problema com proxy:
  + npm config set strict-ssl false
* npm config set https-proxy <http://proxycefsa.cefsa.corp.local:8080>

Configuração do VS Code - Extensões









Configuração para NODE-JS

Para criar um novo projeto, dentro de uma pasta vazia:

npm, init –y

npm install express

npm install nodemon –g

npm install mongoose

npm install cors

c:\users\pro6761\appdata\roaming\npm\nodemon

Configurar package.json adicionando o parâmetro Start e configurando para sua pasta (seu RA) onde se lê SUA\_PASTA:

 "scripts": {

    "test": "echo \"Error: no test specified\" && exit 1",

    "start": "c:\\users\\SUA\_PASTA\\appdata\\roaming\\npm\\nodemon src/index.js"

  },

Criar a pasta src. Dentro da pasta src criar o arquivo index.js:



const express = require('express');

const cors= require('cors');

const server = express();

server.use(express.json());

server.use(cors());

server.get('/teste', (req,res) => {

    res.send('<h1> Servidor Operante </h1>');

});

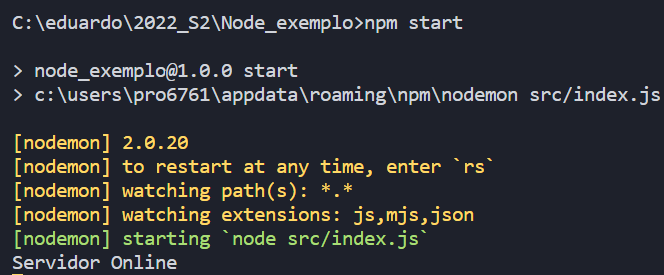
server.listen(3000, ()=>{

    console.log("Servidor Online");

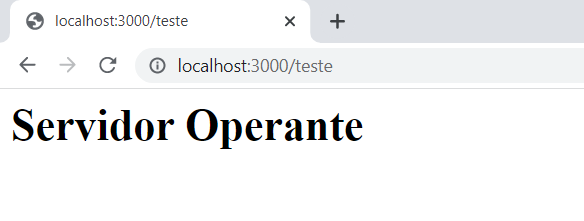
});

Para testar, digite no terminal:

npm start

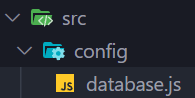


Conferir se o servidor está ok pelo navegador:



Conexão com o banco de dados (Mongo)

Crie a pasta e arquivo abaixo:



Conteúdo do Database.js:

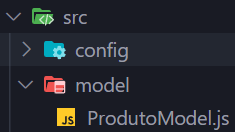
const mongoose = require('mongoose');

const url = 'mongodb://localhost:27017/produtos';

mongoose.connect(url, { useNewUrlParser: true }); *// https://mongoosejs.com/docs/5.x/docs/deprecations.html*

module.exports = mongoose;

Configuração da Model (Crie a pasta model e o arquivo ProdutoModel.js)



const mongoose = require('../config/database');

const ProdutoSchema = new mongoose.Schema(

    {

        id: { type: Number, required: true },

        nome: { type: String, required: true }

    }

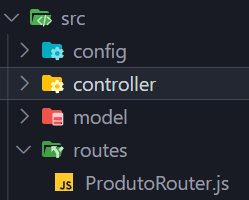
)

module.exports = mongoose.model('Produto', ProdutoSchema);

Controller:



Arquivo de Rotas:



Adicione ao final do index.js as rotas:

*//Rotas*

const ProdutoRoutes = require('./routes/ProdutoRouter');

server.use('/produto', ProdutoRoutes);

Instalação do android studio

