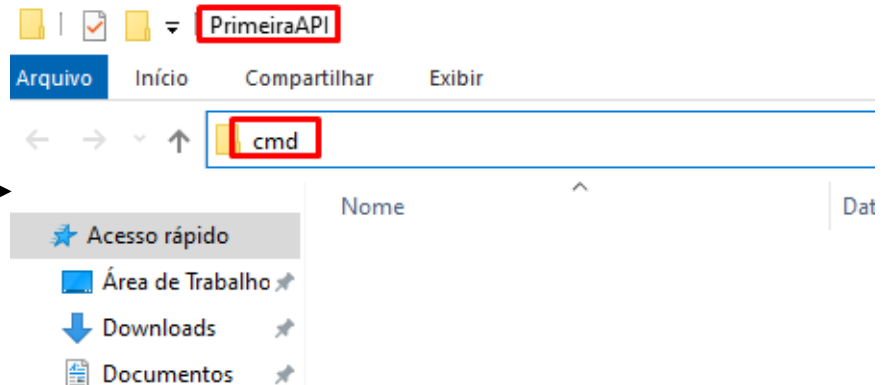


Iniciando a aula

Criar uma pasta chamada "SegundaAPI"

Dentro da pasta "SegundaAPI", digitar cmd para abrir o prompt



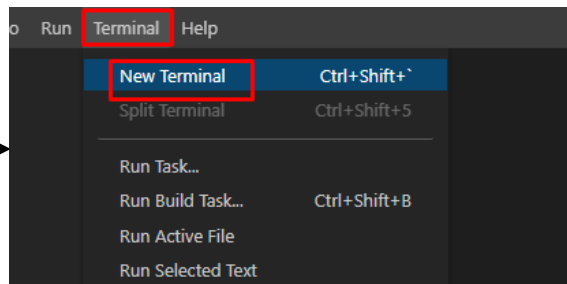
Digitar o comando para iniciar o projeto, dar enter em tudo

```
npm init  
package.json
```

Entrar dentro da pasta pelo terminal e digitar "code ." -> Para abrir o VSCode

```
code .
```

Abrir um terminal dentro do VSCode



No terminal do VSCode colocar o comando -> npm i express

```
npm i express
```

Para instalar um conjunto de bibliotecas no projeto para ajudar na construção da aplicação

Colocar um arquivo index.js no projeto

```
index.js > ...  
  
const express = require('express')  
const server = express()  
  
server.listen(3000, () => {  
  console.log('Servidor está funcionando...')  
})  
  
server.get('/', (req, res) => {  
  return res.json({mensagem: 'Nossa API está funcionando'})  
})
```

no terminal rodar o comando -> node index.js

PS D:\Vinicius\Cursos\SuperGeeks\Experts\SegundaAPI> **node index.js**
Servidor está funcionando...

entrar no navegador com o localhost:3000/ e localhost:3000/usuario

Criar uma pasta src/API e colocar o arquivo index.js dentro dela

```
> node_modules
  v src\API
    JS index.js
    {} package-lock.json
    {} package.json
```

Tentar dar o comando node index.js e verificar que está com erro

É necessário mudar para mostrar o caminho correto

node src/API/index.js

instalar o nodemon para atualizar nossa API automaticamente sem precisar parar

npm i nodemon

```
SEGUNDAAPI
> node_modules
  v src\API
    JS index.js
    {} package-lock.json
    {} package.json

{} package.json > {} scripts
1 {
2   "name": "segundaapi",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "start": "nodemon ./src/API/index.js",
8     "test": "echo \"Error: no test specified\" && exit 1"
9   },
10  "author": "",
11  "license": "ISC",
12  "dependencies": {
13    "express": "^4.17.3",
14    "nodemon": "^2.0.15"
15  }
16 }
17
```

> npm start

Explicar o que é versionamento de código

Criar uma conta no GitHub

Criar um repositório no GitHub

Overview

Repositories 19

Projects

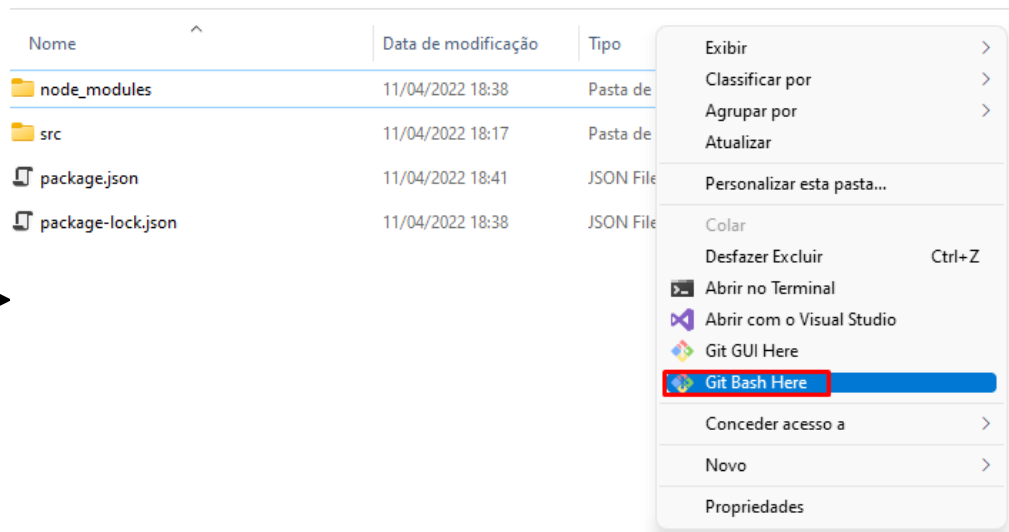
Packages

Stars 1

New

Preencher o título e descrição e NÃO selecionar para adicionar o arquivo README, para aparecer o passo a passo

Entrar no diretório da pasta e clicar com o botão direito



Inicializar um repositório

```
vinic@DESKTOP-VK9QC9V MINGW64 /d/Vinicius/Cursos/SuperGeeks/Experts/SegundaAPI
$ git init
Initialized
```

Adicionar um arquivo README

```
vinic@DESKTOP-VK9QC9V MINGW64 /d/Vinicius/Cursos/SuperGeeks/Experts/SegundaAPI
$ echo "# Aula02-API" >> README.md
```

Indicar os arquivos para enviar

```
vinic@DESKTOP-VK9QC9V MINGW64 /d/Vinicius/Cursos/SuperGeeks/Experts/SegundaAPI
$ git add .
warning: LF
```

Realizando a mudança

```
vinic@DESKTOP-VK9QC9V MINGW64 /d/Vinicius/Cursos/SuperGeeks/Experts/SegundaAPI
$ git commit -m "Estrutura da pronta"
```

Adicionar um repositório remoto

```
vinic@DESKTOP-VK9QC9V MINGW64 /d/Vinicius/Cursos/SuperGeeks/Experts/SegundaAPI
$ git remote add origin https://github.com/ViniciusMazinetti/Aula02-API.git
```

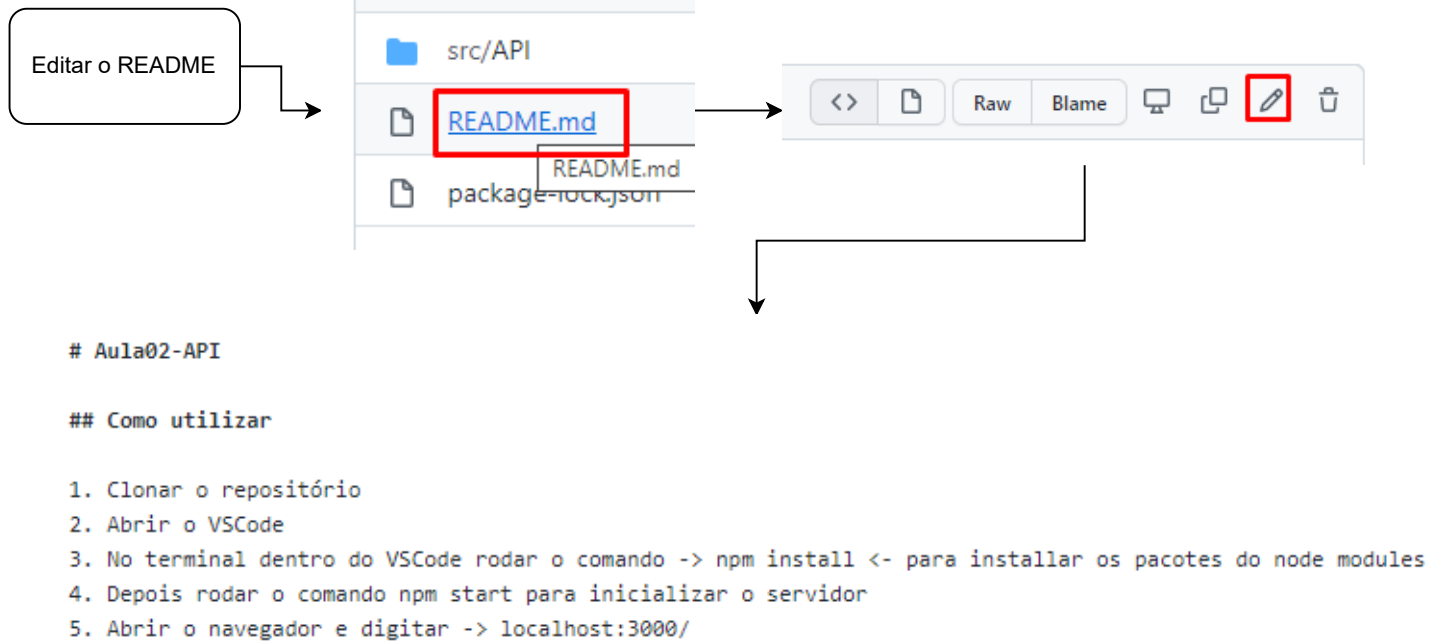
Colocar no repositório remoto

```
vinic@DESKTOP-VK9QC9V MINGW64 /d/Vinicius/Cursos/SuperGeeks/Experts/SegundaAPI (master)
$ git push -u origin master
```

Deletar o node modules e colocar no git

```
vinic@DESKTOP-VK9QC9V MINGW64 /d/Vinicius/Cursos/SuperGeeks/Experts/SegundaAPI
$ git add .
vinic@DESKTOP-VK9QC9V MINGW64 /d/Vinicius/Cursos/SuperGeeks/Experts/SegundaAPI
$ git commit -m "Deletando o node modules"
master ad7c6417 Deletando o node modules
vinic@DESKTOP-VK9QC9V MINGW64 /d/Vinicius/Cursos/SuperGeeks/Experts/SegundaAPI
$ git push origin master
Enumerating objects: 3, done.

```



Commit changes

Atualizando o README

Add an optional extended description...

☒ Commit directly to the master branch.

☐ Create a new branch for this commit and start a pull request

Commit changes Cancel

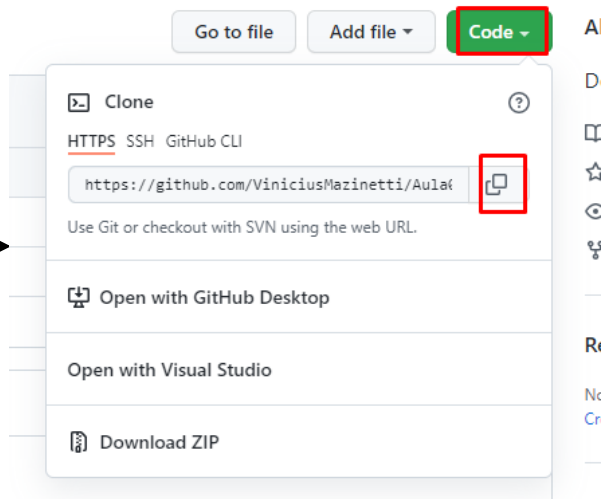
Entrar no README dentro do computador e mostrar que está desatualizado

Atualizar o repositório no git bash

```
git pull origin master
remote: Enumerating objects
```

Fechar o VSCode , terminal e gitBash

Criar outra pasta para exemplificar o clone



```
git clone https://github.com/ViniciusMazinetti/Aula02-API.git
```

```
cd Aula02-API/
```

Para funcionar é necessário instalar o node modules com o comando

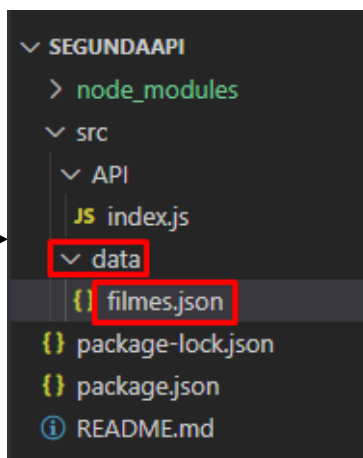
```
npm install
```

Iniciar o servidor novamente

```
npm start
```

Exercicio -> Mapear uma rota GET '/endereco'

Adicionar dados de filmes



em filmes.json
escrever

```
[
  {
    "id": "1",
    "nome": "Vingadores Guerra Infinita"
  },
  {
    "id": "2",
    "nome": "Spider-man"
  }
]
```

Fazer a rota no
index.js

```
const express = require('express');
const server = express();
const filmes = require('./src/data/filmes.json');

server.get('/filmes', (req, res) => {
  return res.json(filmes)
});

server.listen(3000, () => {
  console.log('Servidor está funcionando...')
});
```

Testar

Para o servidor com
CTRL+C

Deletar o node
modules e colocar no
git

```
vinic@DESKTOP-VK9QC9V MINGW64 /d/Vinicius/Cursos/Sup
master)
$ git add .

vinic@DESKTOP-VK9QC9V MINGW64 /d/Vinicius/Cursos/Sup
master)
$ git commit -m "Adicionando arquivo data"
[master dc1b627] Adicionando arquivo data
2 files changed, 12 insertions(+), 1 deletion(-)
create mode 100644 src/data/filmes.json

vinic@DESKTOP-VK9QC9V MINGW64 /d/Vinicius/Cursos/Sup
master)
$ git push origin master
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
```