Relatório – Desenvolvimento da API – 08/03/2023

Nome: Bruno do Nascimento Rodrigues

1- Objetivos

Desenvolver uma API em javascript e subir para um container no docker rodando a aplicação feita e o node.

Aprender a “se virar” sozinho, ou pedir ajuda a outros desenvolvedores quando tiver dificuldades.

2- Atividade

- Data de início da atividade: 03/03/2023 as 21:30h.

Inicialmente, busquei relembrar alguns conhecimentos básicos no google e em projetos antigos, não mexia com API desde o início do curso. Foi uma etapa muito importante, relembrar da estrutura, o uso do express. Também aproveitei esse momento para ver a estrutura de um JSON usando uma “free” API no rapidapi.com.

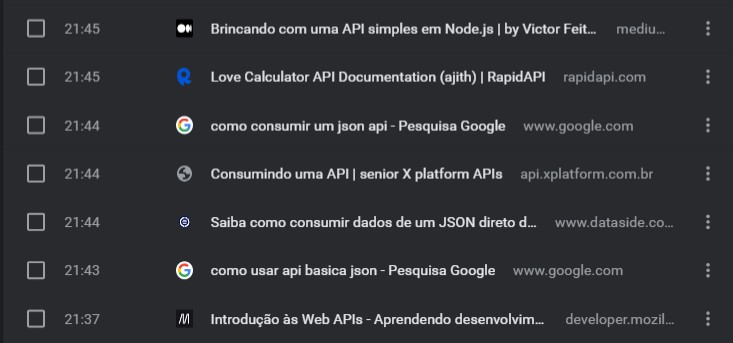


Figura 1 histórico do google do início da atividade.

Comecei a aplicação criando o arquivo app.js e sua estrutura básica, definindo as constantes express, app, port, uma app.get padrão com o caminho “/” somente, e o app.listen(port) para rodar na porta 3000 e verificar se estava funcionando.

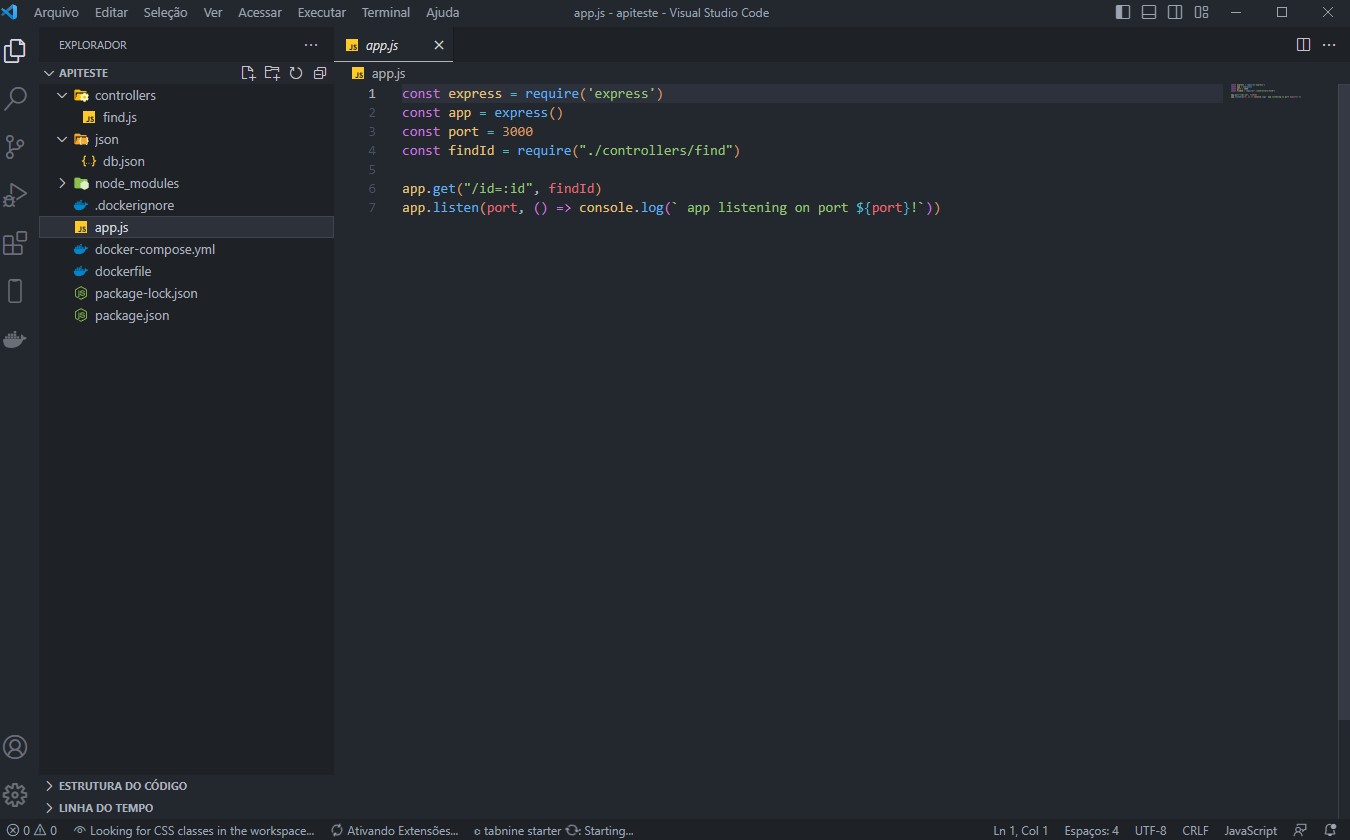


Figura 2app.js – finalizada.

Feita a estrutura base da API, criei pastas para separar os arquivos utilizados de forma organizada, json e controllers, contendo respectivamente, db.json e find.js.

Fiz o arquivo JSON olhando uma response que obtive através do site rapidapi.com

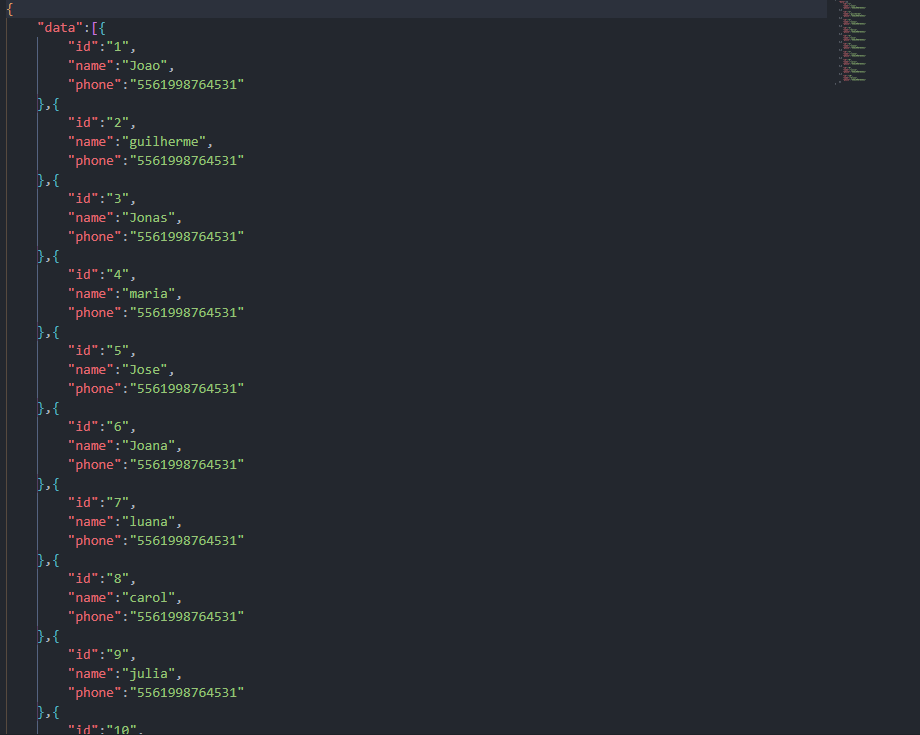


Figura 3 arquivo JSON.

Importei o arquivo db.json da pasta json para ser usada no arquivo find.js. Durante a função – só tinha escrito o if – dei uma olhada sobre path params, não conhecia ainda, apenas tinha usado header e url-form-encoded. Para minha surpresa era bem simples e atualizei o path no app.get do arquivo app.js e coloquei o id obtido na variável pathId, para compará-lo com os ids do json.

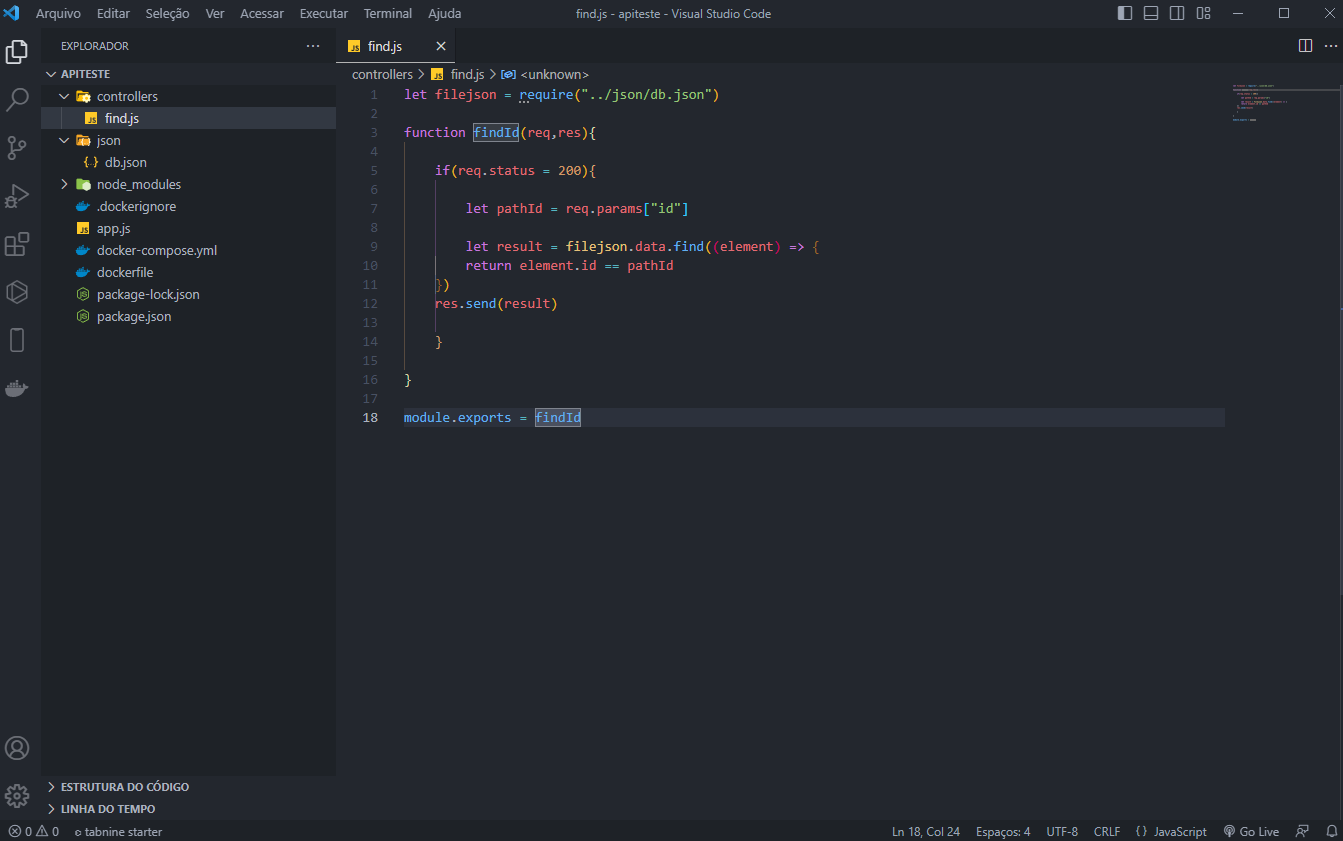


Figura função de busca do id.

As 23:30 terminei a aplicação e fiz alguns testes rápidos no navegador para confirmar que estava tudo ok. A response estava retornando o esperado, no entanto, resolvi mudar para a function find() à filter(), pois essa última retornava dentro de um array desnecessário, visto que a busca era feita por id, o response sempre retornará apenas um item.

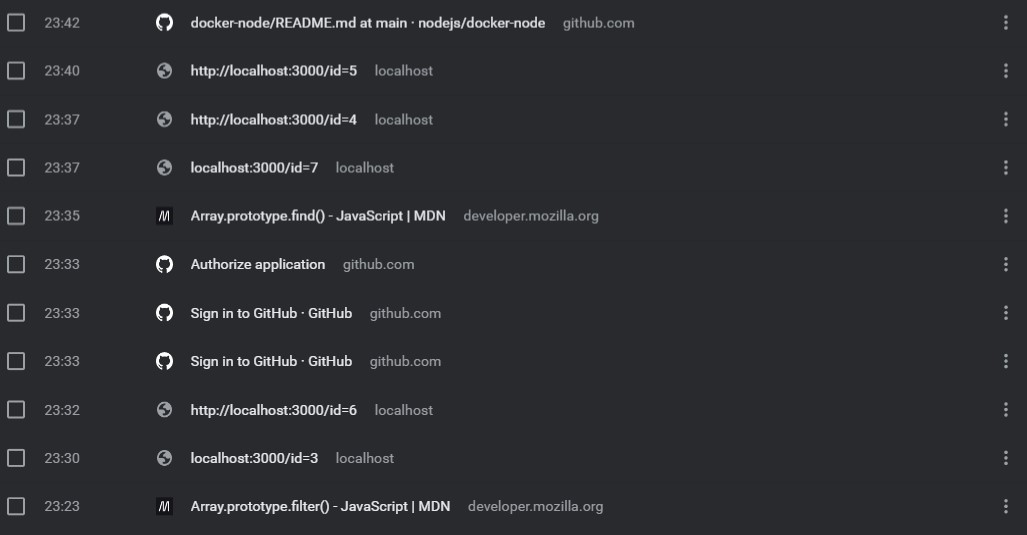


Figura 5 histórico do google - testando a api.

Para subir a aplicação para o docker foi onde encontrei maiores dificuldades, estava tentando fazer da maneira que aprendi nas atividades passadas, porém não funcionaram como esperava, o container iniciava e parava no mesmo instante.

Busquei por algumas referências em sites de como devia ser feito o container, obtive relativo sucesso, o container rodava – ficava verde -, entretanto, não iniciava a aplicação.

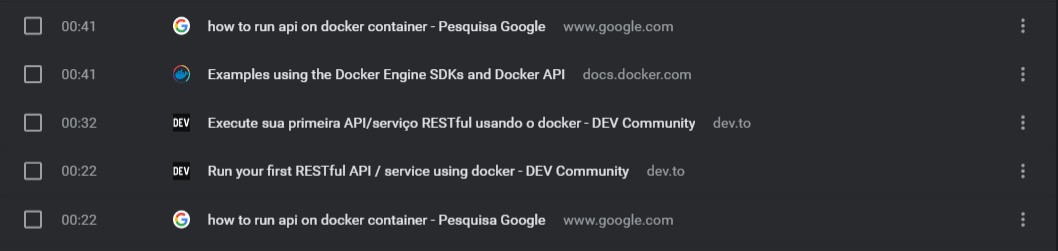


Figura 6 histórico do google – referencias.



Figura 7 dockerfile da tentativa.

Após algumas tentativas e falhas, busquei por referencias no youtube para o problema e encontrei o vídeo que resolveu meus problemas e me ensinou um pouco mais sobre dockerfile, e outros dois arquivos que não conhecia, .dockerignore e docker-compose.

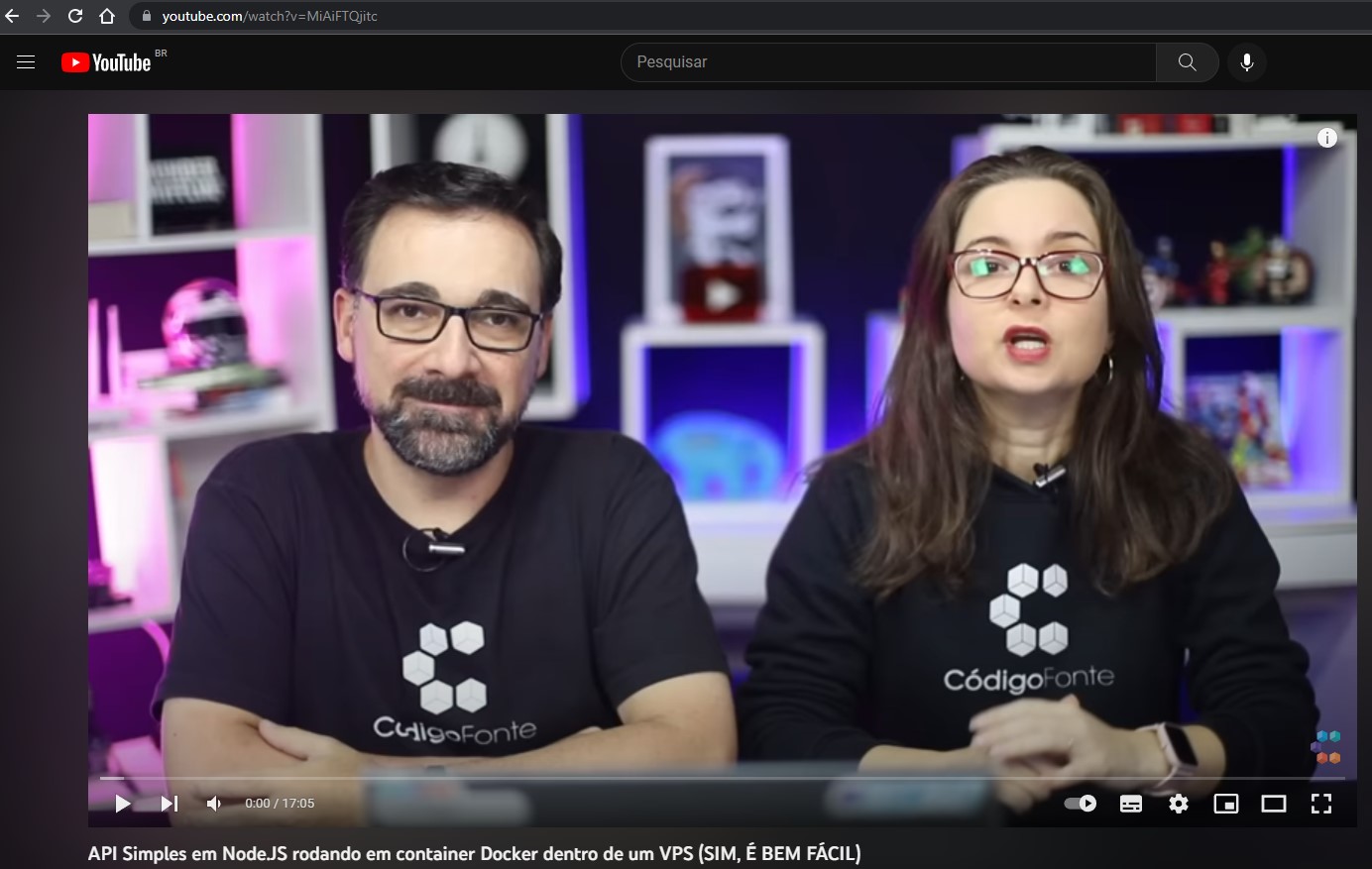


Figura 8 vídeo sobre node no docker.

Fiz as mudanças necessárias no dockerfile, mesmo que não tenha entendido o que seria chown, e acrescentei os dois novos arquivos na pasta. No dockerignore, coloquei os arquivos que não são necessários copiar para o docker. No docker-compose, os scripts necessários para fazer o build do dockerfile e em seguida rodar a aplicação automaticamente na porta 3000.

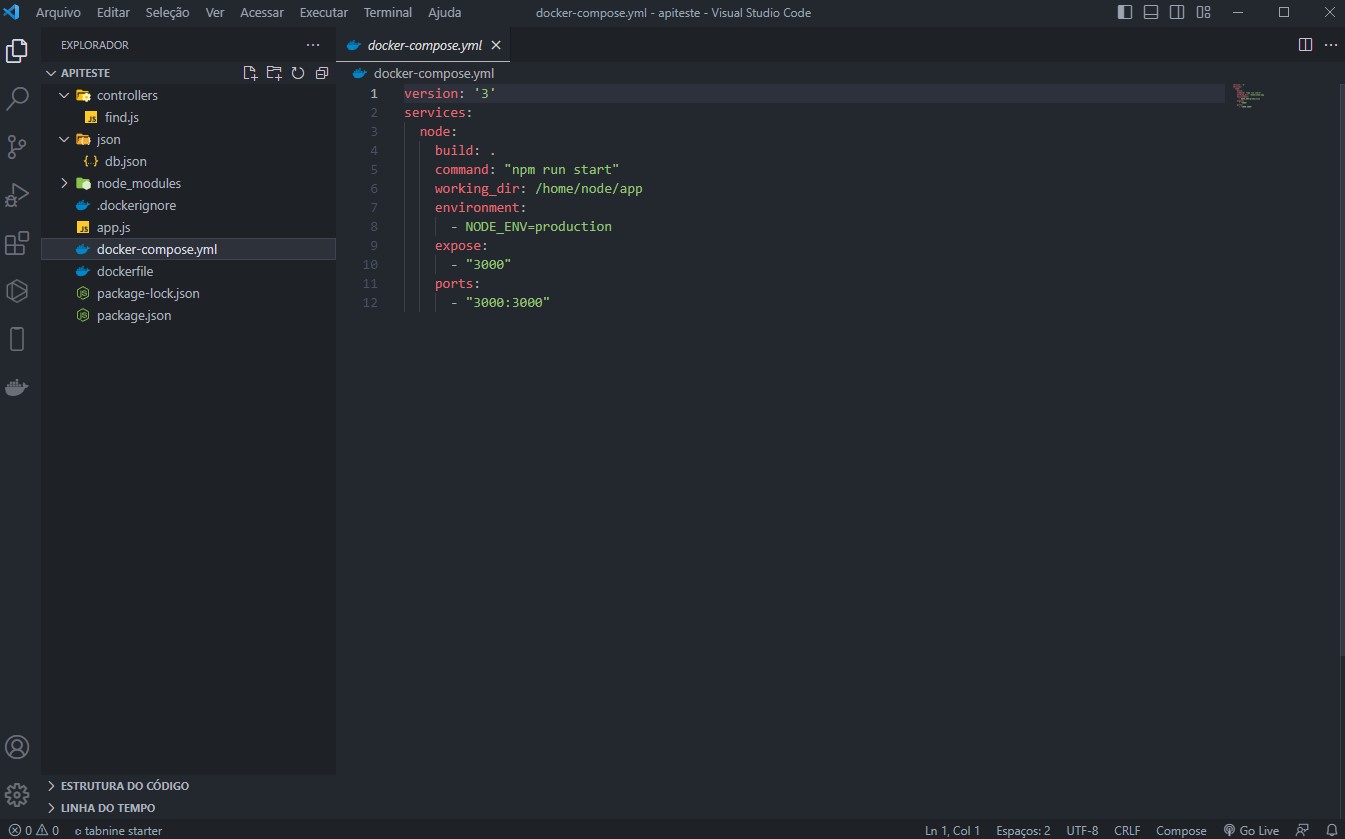


Figura 9 script do docker-compose.

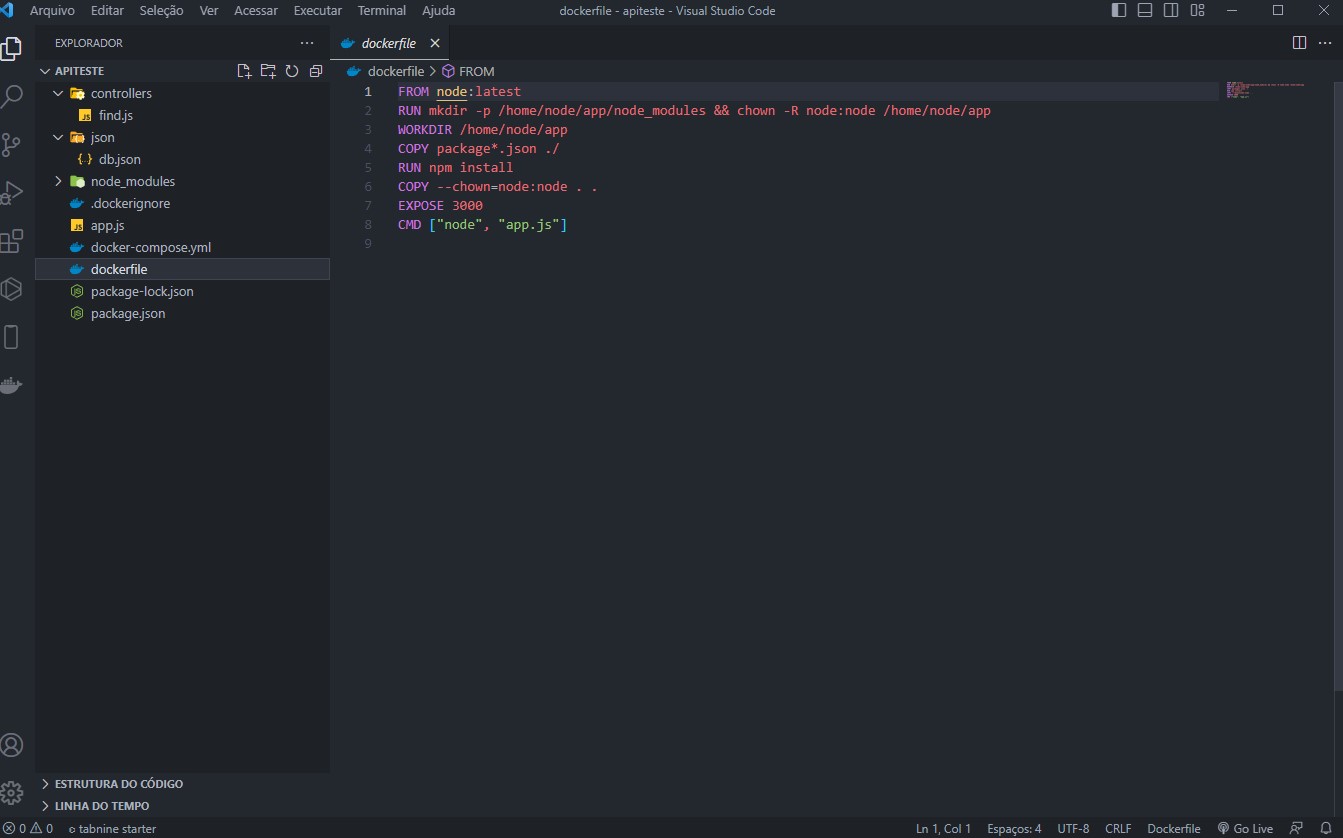


Figura 10 Script do dockerfile.

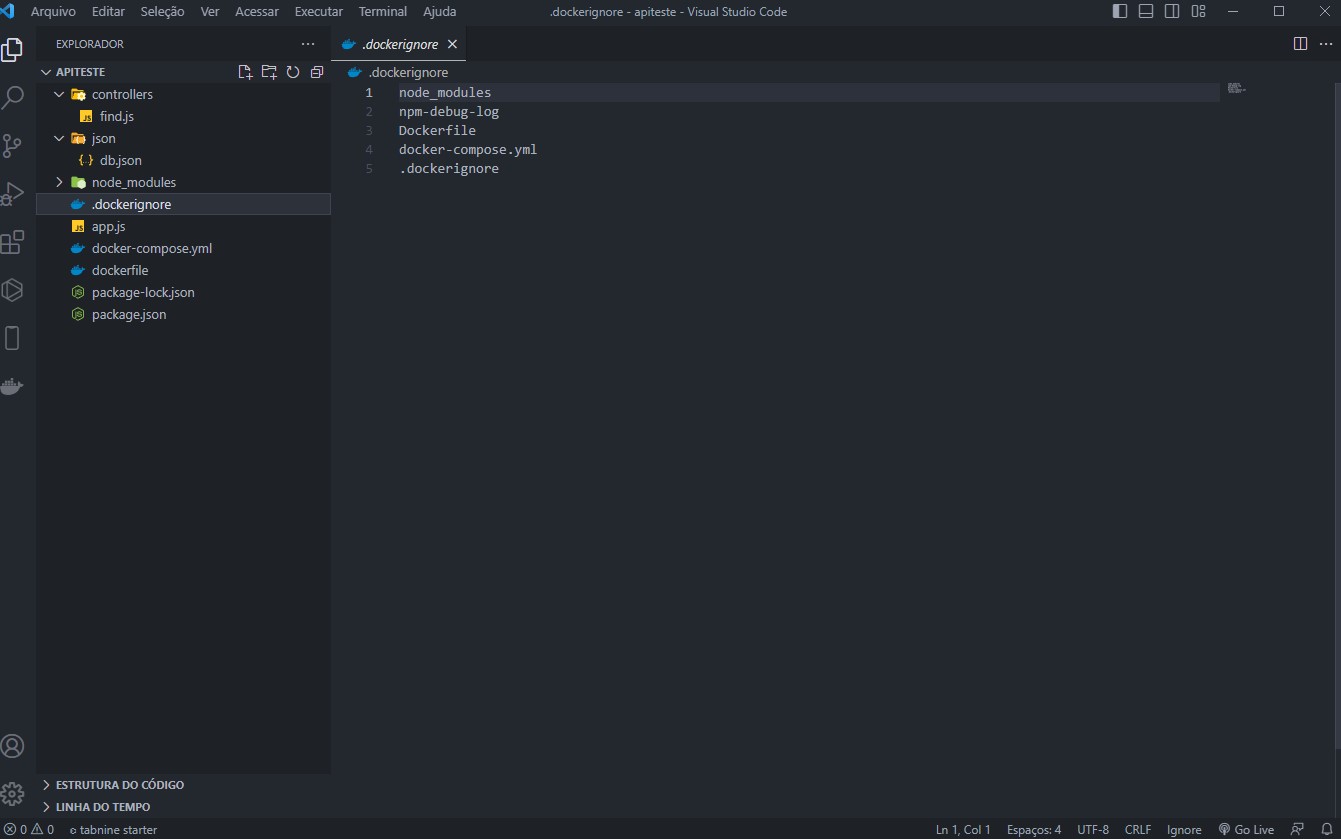


Figura 11 arquivos ignorados pelo. Dockerignore.

No docker, o container se encontra da seguinte forma – em verde – e sua imagem apinode-node respectivamente:

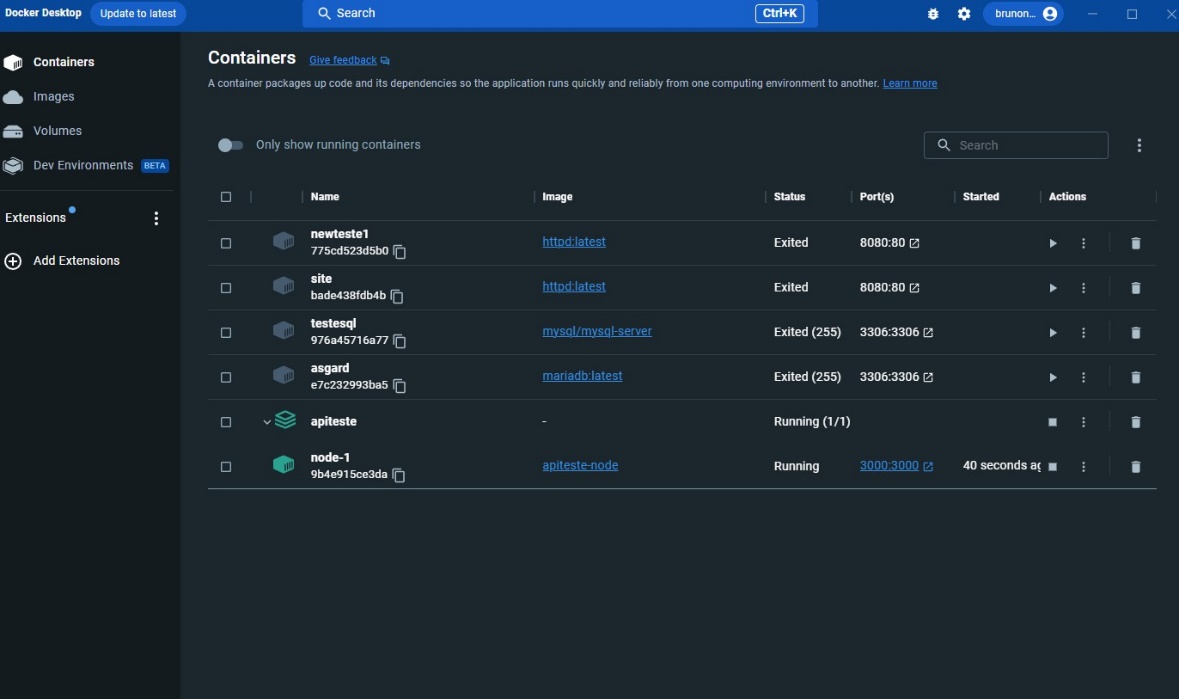


Figura 12 container rodando na porta 3000.

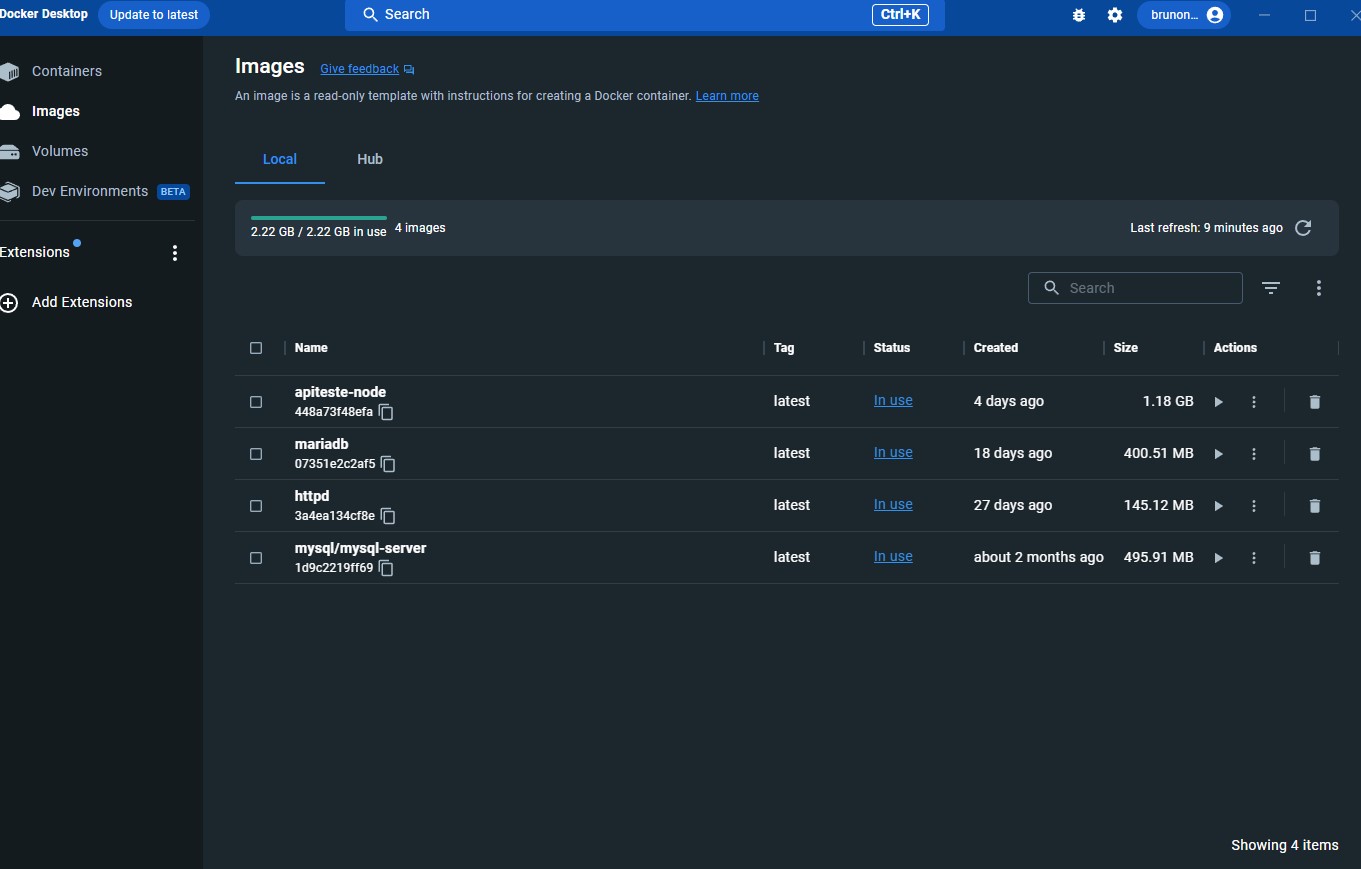


Figura 13 imagem criada com a api e o node.

O log do container para confirmar que estava tudo rodando perfeitamente.

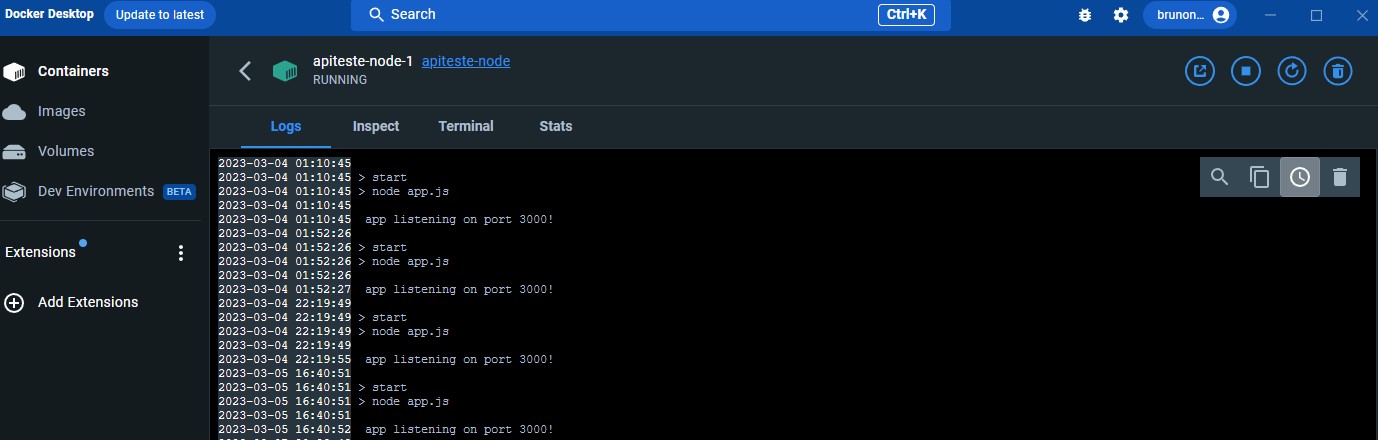


Figura 14 log da aplicação rodando.

Antes de confirmar que havia finalizado a tarefa para dupla de QA, fiz alguns testes com através do Insomnia – já havia instalado na máquina há algum tempo -, a fim de evitar possíveis erros ao instalar na máquina de QA.

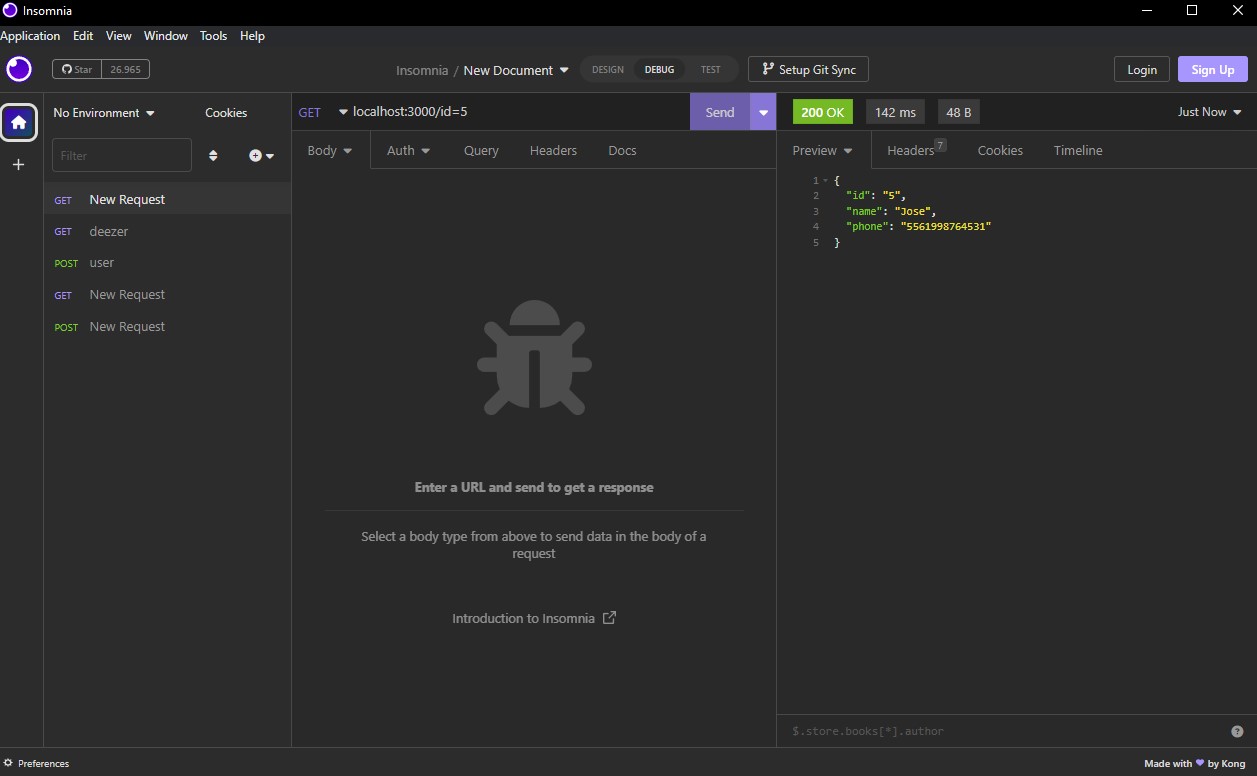


Figura response da API no insomnia.

3 – Conclusão

Foi uma atividade interessante para aprender um pouco mais sobre scripts do dockerfile e docker-compose, e como de certa forma eles facilitam a criação de um container, além de automatizar para melhor funcionamento.

Quanto a api, aprendi sobre path params e relembrei vários conhecimentos que acabei esquecendo sem praticar no dia a dia.