

5 - Rodar o comando `docker-compose up -d --force-recreate` no terminal, para atualizar as informações do container.

Para testar, você pode verificar se a aplicação está rodando, através do comando `docker logs rentx -f`

IMPORTANTE

Testar também as migrations, com o comando `yarn typeorm migration:run`

Se tudo der certo, vencemos o DOCKER 🚀

Explicação

Mas você não disse que utilizando a propriedade `links` conseguiríamos utilizar o nome do service ao invés do IP? Por que então utilizamos `localhost` no host do `ormconfig.json`?

O que acontece é que no momento em que o arquivo `ormconfig` é carregado no sistema para iniciar a nossa aplicação, o `typeorm` não consegue utilizar a referência que criamos na propriedade `links`, por isso utilizamos o `localhost` como valor para a propriedade `host`.

Por isso no nosso arquivo de configuração de criação de conexão (`database/index.ts`) precisamos sobrescrever o host do nosso banco de dados, nesse momento sim apontando para o valor que colocamos no `links`.

Certo! Entendi. Mas por que criamos a interface `IOptions` ?

Boa pergunta! A propriedade `host` por padrão é uma propriedade somente de leitura. Se tentarmos sobrescrever o valor dela, o typescript recusa. Por esse motivo criamos uma interface, somente com a propriedade `host` e forçamos que o tipo nas nossas `options` seja da interface. Quando fazemos `const newOptions = options as IOptions` o nosso atributo `newOptions` e `options` se comportam como se fossem o mesmo objeto e aí com isso conseguimos sobrescrever
