

Explicação Docker

Vamos usar o Docker porque ele garante que todos tenham o mesmo ambiente, o que facilita a configuração do banco de dados e elimina a necessidade de baixar o MySQL separadamente.

O Docker cria "mini máquinas virtuais" na sua máquina com apenas o necessário para rodar a aplicação.

Neste caso, teremos um dockerfile para executar o Java a partir do arquivo .jar compilado e outra para executar o banco de dados MySQL.

A desvantagem é que cada pessoa terá sua própria versão do banco de dados, então não teremos os mesmos dados em cada instância.

No entanto, para propósitos de teste, isso é irrelevante.

Explicando os comandos do dockerfile:

FROM xxx: Define a imagem base para o contêiner.

WORKDIR /x/xx: Especifica o diretório de trabalho dentro do contêiner.

COPY x.jar: Copia arquivos para dentro do contêiner.

ENV X=x: Define variáveis de ambiente.

EXPOSE 8080: Expõe portas para o contêiner.

CMD ['x']: Define o comando a ser executado quando o contêiner for iniciado.

Explicação Docker Compose

Docker Compose é uma maneira de estruturar diferentes containers dockers em um só arquivo, esse arquivo irá montar e "subir" os dockers que declaramos dentro dele.

Ele é usado através de um arquivo .yml que fica na raiz do projeto, lá fica declarado as configurações de como os dockers serão estruturados.

O que precisamos para rodar o docker:

Vocês podem usar o docker via CLI com o docker mas acredito que isso não está sendo muito bem suportado.

Aconselho vocês a baixarem o docker desktop para poder usar, ele vai pedir para vocês habilitar virtualização no seu pc, se tiver dúvidas podem me chamar (Itatiano).

Comandos necessários para usar o docker:

1. docker compose up - Usado para subir os dockers, use na raiz do projeto
2. docker-compose down - Usado para parar os containers, use na raiz do projeto

Coloquei os dois no README do projeto para maior facilidade