

## Dockerfiles

### orders-api/Dockerfile - Configura container da API Python.

Seleciona a imagem base do python, cria um diretório /app dentro do container, copia o requirements.txt e os outros arquivos no diretório pra dentro do container, roda o pip install com o requirements.txt, expõe a porta 3002 e roda o app.py

### payments-api/Dockerfile - Configura container da API PHP.

Seleciona a imagem base do PHP, cria um diretório /app dentro do container, copia os arquivos do diretório pra dentro do container, expõe a porta 3003 e roda o index.php

### payments-api/Dockerfile - Configura container da API Node.

Seleciona a imagem base do Node, cria um diretório /app dentro do container, copia o package.json e demais arquivos pra dentro do container, roda o npm install para instalar as dependências, expõe a porta 3001 e roda o app.js

Uma observação aqui; Tendo em vista que eu usei o npm install express para instalar o express e criar o package.json, a pasta node\_modules foi gerada. Tive o cuidado de adicionar um .dockerignore para evitar que essa pasta fosse junto pra dentro do container e evitar problemas com dependências.

---

## Docker-Compose

O docker-compose inicia escolhendo a versão do compose a ser utilizada.

Na sessão de services, ele cria um container para cada serviço que será utilizado, por exemplo, no serviço products, ele constroi a imagem a partir do dockerfile contido no diretório especificado no comando "build", nomeia o container e faz com que ele escute na sua porta exposta.

No seguinte serviço, o orders, ele faz as mesmas coisas porém acrescenta uma instrução "depends\_on" e passa os serviços products, redis e db. Isso faz com que esse serviço só seja iniciado após os 3 listados, indicando assim que este depende dos outros.

O mesmo ocorre com o serviço payments, ele deve ser iniciado após o orders.

Nesta mesma sessão, é passada a instrução para buildar e iniciar o serviço redis, que será usado para cache pela api em python. Ele ouvirá na porta 6379, tradicionalmente utilizada no redis.

No serviço db, é escolhida a imagem do mysql, nomeado o container, passado algumas configurações de ambiente, selecionado a porta do banco (Tive que alterar para 3666) e, por fim, passa os volumes onde os arquivos do banco devem ser salvos

A ultima parte cria o volume do banco, os dados vão ficar ali pra não perder quando os containers forem derrubados.