

## Entendendo a aplicação

### Transcrição

Para começarmos a entender como funciona o **Docker Compose**, primeiramente vamos entender como funciona a aplicação que utilizaremos como base. É uma aplicação bem semelhante à utilizada na aula anterior, com o mesmo servidor, rotas e banco de dados. De novidade, é que agora precisamos criar o **NGINX**, que é mais um *container* que devemos subir.

Então, ou utilizamos a imagem **nginx**, ou criamos a nossa própria. Como vamos configurar o **NGINX** para algumas coisas específicas, como lidar com os arquivos estáticos, vamos criar a nossa própria imagem, por isso que na aplicação há o **nginx.dockerfile**:

```
FROM nginx:latest
MAINTAINER Douglas Quintanilha
COPY /public /var/www/public
COPY /docker/config/nginx.conf /etc/nginx/nginx.conf
EXPOSE 80 443
ENTRYPOINT ["nginx"]
# Parametros extras para o entrypoint
CMD ["-g", "daemon off;"]
```

Nesse arquivo, nós utilizamos a última versão disponível da imagem do **nginx** como base, e copiamos o conteúdo da pasta **public**, que contém os arquivos estáticos, e um arquivo de configuração do NGINX para dentro do *container*. Além disso, abrimos as portas **80** e **443** e executa o NGINX através do comando **nginx**, passando os parâmetros extras **-g** e **daemon off**.

Por fim, vamos ver um pouco sobre o arquivo de configuração do NGINX, para entendermos um pouco como o *load balancer* está funcionando.

No arquivo **nginx.conf**, dentro **server**, está a parte que trata de servir os arquivos estáticos. Na porta 80, no *localhost*, em **/var/www/public**, ele será responsável por servir as pastas **css**, **img** e **js**. E todo resto, que não for esses três locais, ele irá jogar para o **node\_upstream**.

No **node\_upstream**, é onde fica as configurações para o NGINX redirecionar as conexões que ele receber para um dos três *containers* da nossa aplicação. O redirecionamento acontecerá de forma circular, ou seja, a primeira conexão irá para o primeiro *container*, a segunda irá para o segundo *container*, a terceira irá para o terceiro *container*, na quarta, começa tudo de novo, e ela vai para o primeiro *container* e assim por diante.

Isso já está tudo pronto, basta baixarmos o código da imagem e da aplicação [aqui \(https://s3.amazonaws.com/caelum-online-public/646-docker/06/projetos/alura-docker-cap06.zip\)](https://s3.amazonaws.com/caelum-online-public/646-docker/06/projetos/alura-docker-cap06.zip).

Agora, no próximo vídeo, escreveremos o responsável por orquestrar a subida de cada uma dessas partes da nossa aplicação, o **docker-compose.yml**.

