# **Automação de processos**

**Criando uma Aplicação Docker**

## **Alunos:** Jhonatas Junnior, Romulo Oliveira; **7º período - Ipatinga/MG;**

## **Prof.** Luiz Fernando.

## **Emails Institucionais:** aluno.jhonatas.silva@doctum.edu.br; aluno.romulo.oliveira@doctum.edu.br.

**Serviço a ser ofertado:**

Uma agenda telefônica.

**Objetivo:**

Fornece ao usuário uma tela para cadastramento de números telefônicos, sendo assim uma agenda telefônica. Depois de salvo os números cadastrados serão listados numa tabela.

**Introdução**

Vivemos em um mundo onde a comunicação é cada vez mais intensa, com isso necessitamos de guardar os contatos de forma mais rápida e dinâmica, visando isso desenvolvemos uma agenda telefônica, capaz de listar, adicionar e excluir os contatos a qualquer hora de forma rápida e prática.

Nela podemos cadastrar o nome, o telefone e o e-mail do contato, possibilitando os principais meios de comunicação que utilizamos atualmente.

Nosso projeto de uma agenda telefônica será aplicado em Docker utilizando 3 containers, um container terá a aplicação, outro terá o banco de dados MySQL e o ultimo terá o PHPMyAdmin que administra o banco de dados e faz a conexão com a aplicação.

**Justificativa**

A ideia da agenda telefônica em web é proporcionar ao usuário que tenha um tipo de agenda compartilhada, fornecendo a comodidade de obter uma agenda telefônica com outras pessoas, onde todos poderiam ter acesso aos contatos salvos uns pelos outros, a aplicação inicialmente lista todo os contatos cadastrados, os cadastros de contatos só podem ser realizados por usuários que contém um determinado acesso para criação dos mesmos. Ou seja, para criar um contato na agenda compartilhada o usuário deverá conter um cadastro na configuração da agenda. Com o cadastro realizado é só acessar a página de login com seu login e sua senha cadastrada e editar, adicionar ou excluir qualquer contato.

**Relatório Técnico**

Depois de instalado o Docker compose na sua máquina, criado o diretório (pasta) da sua aplicação crie um arquivo novo dentro desse diretório com o nome de docker-compose.yml.

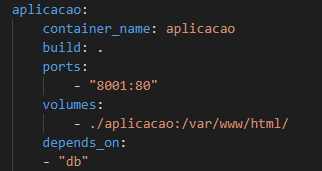
Dentro do arquivo criado especificar qual a versão do Docker compose iremos utilizar, no nosso caso foi “3.1”, em seguida escrever os services (container) e suas especificações.

No site <https://hub.docker.com/> baixamos a imagem do banco, do PHPMyAdmin

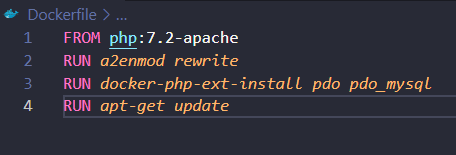
Primeiro criamos o container da aplicação (*aplicacao*) e dentro colocamos o build, volume e a porta que iremos utilizar

No nosso caso ficou:

* container\_name: dando nome ao container.
* build: aponta para a pasta onde fica o arquivo dockerfile.
* ports: redirecionamento do host com o container.
* volumes: Diretório compartilhado entre a máquina host e o nosso container.



Após configurarmos o primeiro container criamos um arquivo chamando DockerFile com os seguintes parâmetros:



Instanciamos a versão do php apache que será iniciada com o projeto e colocamos 3 RUN pra rodar updates e instalações assim que levantarmos os containers:

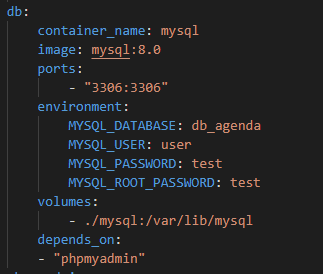
O primeiro RUN com o mod\_rewrite é uma implementação do apache para manipulação de maneira eficiente de urls feitas pelo lado do servidor;

O segundo RUN são ferramentas para facilitar a inclusão de módulos PHP no Docker;

E o terceiro RUN é um update de versões dos pacotes disponíveis do Linux.

Voltando para o arquivo Docker-compose, no nosso segundo container (*db)* foi utilizando as seguintes propriedades:

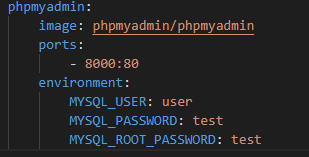
* container\_name: dando nome ao container.
* image: Imagem do banco.
* ports: Redirecionamento do host com o container.
* command: Serve para executar o serviço assim que for subido o container.
* volumes: Diretório compartilhado entre a máquina host e o nosso container.
* environment: Configura o banco de dados, usuário, a senha do usuário e a senha do root (administrador).

**Nosso container:**

No terceiro container (*phpmyadmin*) utilizamos os seguintes comandos:

* image: imagem da aplicação.
* ports: Redirecionamento do host com o container.
* environment: Configura o banco de dados, usuário, a senha do usuário.

**Nosso container:**



**Para rodar a aplicação, utilizamos os seguintes comandos:**

docker-compose build – Para fazer a criação dos serviços.

docker-compose up – Levantar os contêineres.

Quando levantados no diretório dos serviços é criado uma pasta com o nome de aplicação que foi especificada no primeiro container, o de aplicação. Nessa pasta aplicamos os arquivos da nossa aplicação (Agenda Telefônica).

Depois só acessar a aplicação por <http://localhost:8001> e o banco pelo <http://localhost:8000>

Acessando o mysql via porta 8000 criamos o banco da nossa aplicação (Script de criação do banco contido na pasta do trabalho).

**Conclusão**

Através do desenvolvimento do atual projeto, percebemos como a implementação da aplicação é fácil e reduz significativamente o tempo de implantação, com poucos comandos a aplicação web está online e funcionando perfeitamente.