



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO  
TOCANTINS  
CAMPUS ARAGUATINS  
CURSO SUPERIOR DE PÓS-GRADUAÇÃO LATO SENSU EM  
DESENVOLVIMENTO DE SISTEMAS COMPUTACIONAIS**

**AMADEU DE SOUSA CASTRO JÚNIOR**

**Utilização da plataforma docker na criação de contêineres para o  
desenvolvimento de aplicações**

ARAGUATINS-TO  
2021



**AMADEU DE SOUSA CASTRO JÚNIOR**

**Utilização da plataforma docker na criação de contêineres para o desenvolvimento de aplicações**

Trabalho de Conclusão de Curso de pós-graduação lato sensu em desenvolvimento de sistemas computacionais na unidade *campus* Araguatins, do Instituto Federal do Tocantins, como exigência à obtenção do título de especialista em desenvolvimento de sistemas computacionais.

Orientador ME. Ancelmo Frank Coelho Castro

**Dados Internacionais de Catalogação na Publicação (CIP)  
Bibliotecas do Instituto Federal do Tocantins**

---

C355u Castro Júnior, Amadeu de Sousa  
Utilização da plataforma docker na criação de contêineres para o desenvolvimento de aplicações / Amadeu de Sousa Castro Júnior. – Araguatins, TO, 2021.  
19 f. : il. color.

Artigo (Especialização em Desenvolvimento de Sistemas Computacionais) – Instituto Federal de Educação, Ciência e Tecnologia do Tocantins, Campus Araguatins, Araguatins, TO, 2021.

Orientador: Me. Ancelmo Frank Coelho Castro

1. Containers. 2. Docker. 3. Virtualização. I. Castro, Ancelmo Frank Coelho. II. Título.

**CDD 004**

---

A reprodução total ou parcial, de qualquer forma ou por qualquer meio, deste documento é autorizada para fins de estudo e pesquisa, desde que citada a fonte.

Elaborado pelo sistema de geração automática de ficha catalográfica do IFTO com os dados fornecidos pelo(a) autor(a).

## AMADEU DE SOUSA CASTRO JÚNIOR

### Utilização da plataforma docker na criação de contêineres para o desenvolvimento de aplicações

Trabalho de Conclusão de Curso de pós-graduação lato sensu em desenvolvimento de sistemas computacionais na unidade *campus* Araguatins, do Instituto Federal do Tocantins, como exigência à obtenção do título de especialista em desenvolvimento de sistemas computacionais.

Aprovado em: 09/04/2021

### BANCA EXAMINADORA

---

ME. Ancelmo Frank Coelho Castro  
Orientador  
Instituto Federal do Tocantins – *campus* Araguatins

---

ME. Adeilson Marques da Silva Cardoso  
Instituto Federal do Tocantins – *campus* Araguatins

---

ME. Marcos Aurélio Gomes de Oliveira  
Instituto Federal do Tocantins – *campus* Araguatins

## **AGRADECIMENTOS**

Meus agradecimentos especiais para toda minha família, amigos, e todos os demais que estiveram ao meu lado nessa jornada.

## RESUMO

Guia resumido de utilização com uma abordagem em torno do conceito de virtualização, onde são discutidos os modelos de uso da máquina virtual (virtual machine), e do Docker, com suas vantagens e desvantagens, assim como o conceito geral da ferramenta. Relata uma abordagem prática e introdutória sobre o uso da plataforma Docker na criação de containers para implementação de uma infraestrutura padronizada para ambientes de desenvolvimento de software focado em testes. A metodologia abordada, consistiu em pesquisas bibliográfica em artigos, trabalhos acadêmicos e sites. Utilizando um exemplo serão descritos os processos que envolvem as configurações de modo geral, download de imagens no repositório oficial e execução dos containers de forma isolada.

**Palavras-chave:** Containers. Docker. Software. Virtualização.

## **ABSTRACT**

Short usage guide with an approach around the concept of virtualization, where the models of use of the virtual machine (virtual machine) and Docker are discussed, with their advantages and disadvantages, as well as the general concept of the tool. Reports a practical and introductory approach on the use of the Docker platform in the creation of containers for the implementation of a standardized infrastructure for software development environments focused on testing. The approached methodology consisted of bibliographic searches in articles, academic papers and websites. Using an example, the processes involving the configurations will be described, downloading images and executing the containers.

**Keywords:** Containers. Docker. Software. Virtualization.



## LISTA DE ILUSTRAÇÕES

Figura 1	– Modelo de uso de containers 1 – Visão geral.....	15
Figura 2	– Docker Linux 2 – Guia de instalação.....	18
Figura 3	– Dockerfile 3 – Configuração do Dockerfile.....	19
Figura 4	– docker-compose 4 – Configurações gerais.....	20
Figura 5	– Editor Visual Studio Code 5 - Container em execução.....	20

## LISTA DE SIGLAS

VM Virtual machine

## SUMÁRIO

1 INTRODUÇÃO.....	13
2 REVISÃO DE LITERATURA .....	14
2.1 O que é docker e as vantagens em sua utilização.....	14
2.2 Quais as diferenças entre os sistemas de virtualização da virtual machine(vm) e do Docker.....	15
2.3 Repositório docker hub.....	16
3 MATERIAIS E MÉTODOS.....	17
3.1 Instalação da ferramenta.....	17
4 RESULTADOS E DISCUSSÕES.....	21
5 CONSIDERAÇÕES FINAIS .....	21
REFERÊNCIAS .....	24

## 1 INTRODUÇÃO

A padronização no uso de diferentes tecnologias é um desafio complexo e crescente, se tratando de um ambiente de TI, isso fica evidente dentre as diversas demandas oriundas de diferentes situações e com versões de software distintas são cada vez mais recorrentes. Essas demandas surgem por diversos motivos, seja pela escassez de profissionais para compor uma equipe para manter a solução nas condições especificadas em um padrão, ou simplesmente pela mudança nas regras dos softwares que compõem a aplicação que, mesmo poupando trabalho, em determinado momento pode prejudicar a padronização do ambiente.

Com o passar do tempo, e as inúmeras dificuldades das equipes em manter padronizados inúmeros sistemas, foram desenvolvidos tecnologias e conceitos para melhorar o desempenho tanto de hardwares, como softwares, no intuito de automatizar tarefas que na maioria das vezes são trabalhosas e complexas em sua execução.

Em ambiente de desenvolvimento, que faz referência a fase de testes, desenvolvedores de sistemas fazem uso de uma plataforma de virtualização, esse recurso é muito poderoso e conta com diversas vantagens.

Segundo (JUNIOR, 2020), virtualização é a abstração representada de um recurso computacional qualquer, caracterizado principalmente através de máquinas virtuais. Com o uso da máquina virtual é facilmente construído um ambiente completo, similar a uma máquina física, tendo recursos similares como serviço de rede(internet), totalmente isolada da máquina hospedeira.

No contexto de isolar recursos em uma máquina, surge o Docker, uma ferramenta que representa o conceito de container, que se trata de um ambiente aberto de desenvolvimento, que tem como objetivo facilitar o trabalho dos desenvolvedores de software, isso, se faz, com a criação e execução de aplicações em hosts com rapidez e facilidade de acesso, podendo ser replicado de uma forma mais ágil em diversos sistemas operacionais (ANDRADE, 2016).

Esta pesquisa tem como o objeto de estudo mostrar através de exemplos práticos o funcionamento de containers com a plataforma Docker através da virtualização, onde será possível perceber a praticidade e velocidade no uso desta tecnologia, além de demonstrar os conceitos iniciais necessários, desde a obtenção

de imagens e gerenciamento dos containers para os recursos utilizados no phpmyadmin, mysql e Framework AdonisJs, através do ambiente de desenvolvimento NodeJs.

## **2 REVISÃO DE LITERATURA**

### **2.1 O que é docker e as vantagens em sua utilização**

Esse conceito pode estar relacionado a diferentes partes, desde a tecnologia Docker, a empresa Docker Inc e a comunidade Docker.

A tecnologia Docker surgiu a partir de um projeto de software open source (código aberto). Essa é uma tecnologia voltada para a containerização, sendo utilizada para criar containers Linux que atuam como uma máquina virtual modular. Por ser leve e flexível, é comumente adotado por pessoas gestoras de TI que conseguem implantar, copiar e migrar ambientes de trabalho facilmente.

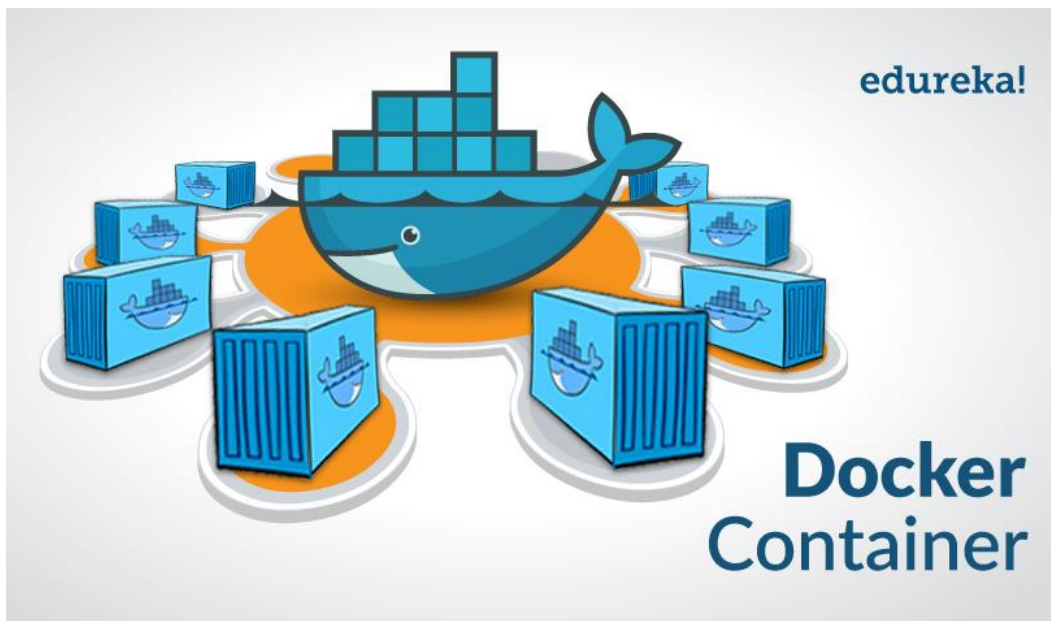
A Docker Inc é uma empresa voltada para a manutenção da plataforma Docker. A mesma ajuda a comunidade em volta do sistema de forma a otimiza a sua manutenção, sendo ainda a principal organização da iniciativa privada voltada para o suporte e destruição a tecnologia Docker em ambientes privados.

A comunidade Docker é composta por profissionais que trabalham voluntariamente em grupos de discussão, sendo o propósito buscar melhorias e a criar novos recursos, para que tecnologia pode evoluir e atender a novos desafios (NOLETO).

Um container Docker é um ambiente isolado que agrupa partes de softwares de um sistema de arquivo completo e que abrange todos os recursos necessários para a sua execução. Por isso, é uma plataforma de containers.

Isso significa que um sistema pode ser dividido, instalado no servidor e armazenado nos containers. Dessa forma, os mesmos softwares e as suas versões podem ter uma execução facilitada independentemente do ambiente de desenvolvimento.

## Modelo de uso de containers 1 – Visão geral



Fonte: site: <https://i1.wp.com/n8d.at/wp-content/uploads/2017/03/Dockerfile-to-create-spx-docker-image-2.png?w=865&ssl=1> (2021)

Na imagem Modelo de uso de containers 1, vemos uma abstração de como o docker mantém em seu ecossistema as imagens, onde cada container representa uma parte da aplicação ou recurso disponibilizado no sistema.

### 2.2 Quais as diferenças entre os sistemas de virtualização da virtual machine(vm) e do Docker

O conceito de virtualização tem origem do paradigma da computação em nuvem, sendo amplamente utilizado no universo tecnológico. Por ter uma forte adesão por parte das empresas e instituições a nível mundial, a virtualização oferece maior flexibilidade, melhor utilização de recursos, maior escalabilidade, adaptabilidade e redução de custos (FERREIRA, 2021).

A diferença com maior destaque é que o container exclui a virtualização pelo hypervisor (software que cria e executa máquinas virtuais) e muda o processo para o docker.

O uso através de uma máquina virtual (VM, do inglês virtual machine), trabalha com um sistema operacional (SO) completo, porém, separado do equipamento. Nesse modelo a execução do software é feita em cima de um servidor físico, tendo como finalidade emular determinado sistema de hardware.

Segundo (MEU POSITIVO, 2020), esse processo é possibilitado através do hypervisor, software que cria e efetiva a VM. Basicamente, ele se localiza entre as camadas do hardware e do SO, sendo fundamental para a virtualização do servidor. Sendo exemplos de hypervisor a VMWare, Hyper-V e o VirtualBox.

Um mesmo Sistema Operacional pode conter inúmeras VMs. Cada uma opera um SO exclusivo possuindo seu próprio kernel, binários, aplicativos e bibliotecas, o que ocasiona em uso de grande parte da memória no servidor.

Através do container Docker, os desenvolvedores conseguem criar e administrar diferentes ambientes de forma isolada, isso faz com que o Docker seja um sistema de virtualização diferente do tradicional.

De forma prática, o S.O. convidado e o hypervisor são eliminados e o host entra em contato direto com as bibliotecas. mediante essa ligação, é criada uma portabilidade entre os itens que podem ser compartilhados entre qualquer outro host que também possua o sistema de virtualização instalado. E como consequência gera uma redução no tempo de deploy de uma aplicação ou infraestrutura.

Devido o sistema ser sempre igual e configurado uma vez, torna-se desnecessário ajustar o ambiente para que o serviço funcione corretamente, bastando apenas replicá-lo. Existem ainda a possibilidades de criar containers customizados, principalmente para deploy quando utilizado em ambiente de produção.

Vale destacar que os containers se localizam em cima de um servidor físico e do Sistema operacional hospedeiro. Onde compartilham o kernel do SO host também podendo compartilhar as bibliotecas e binários (MEU POSITIVO, 2020).

O Docker possui uma documentação aberta, completa e gratuita no seu site, a qual pode ser consultada a qualquer momento para resolução de dúvida, lá estão descritos todos os comandos Docker e seus parâmetros. Dessa forma percebemos que os containers não são de fato VM's, este é um erro que muitos cometem ao dizer que containers são máquinas virtuais "enxutas", logo, as diferenças e arquitetura e funcionamento limitam tal comparação (PINTO, 2019).

### **2.3 Repositório docker hub**

Quando se inicia com a criação de containers, o primeiro passo após a configuração da ferramenta, é baixar as Docker Images conforme a necessidade. Essas imagens são a base do container e criam de forma isolada a parte do sistema

que se quer virtualizar.

A imagem ainda traz uma série de instruções que definem como o container será construído, sendo consideradas o código fonte do container, pois detalham como ele deverá ser criado e configurado. Podemos ressaltar a facilidade em serem compartilhadas, armazenadas e atualizadas (TURNBULL, 2014).

A principal fonte de downloads das imagens é o servidor do Docker Hub, nele é possível baixar diversos tipos, podendo escolher entre a versão oficial, criada pelo desenvolvedor da parte do ambiente que se quer virtualizar, ou mesmo de terceiros, que criam imagens personalizadas e as compartilham.

#### Orquestradores de contêineres

Com o crescimento das aplicações, e maior quantidade de imagens utilizadas, o processo de executar e manter uma por uma passa-se a se tornar custoso. Para contornar esse problema foram criados os orquestradores de containers. Sendo os mais conhecidos: Kubernetes, Docker Swarm, OpenShift e AWS ecs.

Bastante difundido, principalmente devido sua facilidade de uso, aparece o Docker-Compose, um dos orquestradores da Docker. Sua configuração na maioria das vezes requer apenas um arquivo com a extensão YML (acrônimo recursivo para YAML Ain't Markup Language), onde será descrito uma série de regras para gerenciamento dos containers em serie.

### **3 MATERIAIS E MÉTODOS**

#### **3.1 Instalação da ferramenta**

A instalação é bem simples e pode ser realizar nos Sistemas Operacionais Linux, MAC e Windows, detalhe, para instalação neste último S.O. é necessário o sistema de 64bits, além da versão profissional do Windows.

Em nosso exemplo será utilizado a imagem do Nodejs para o container da aplicação, e ainda utilizamos as imagens do PhpmyAdmin, para gerenciar a interface do banco de dados e o Mysql, para receber os dados. E neste mesmo processo foi instalado no container do Node o CLI do Adonis, para termos acesso a sua linha de comandos.

No exemplo vamos instalar no Linux Ubuntu 18.04



## Docker Linux 2 – Guia de instalação

No terminal:

Atualize o sistema

```
sudo apt update  
sudo apt upgrade
```

Instale Pacotes de Pré-requisitos

```
sudo apt-get install curl apt-transport-https ca-certificates software-properties-common
```

Adicione os Repositórios do Docker

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
sudo add-apt-repository "deb [arch=amd64]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

Atualize o repositório

```
sudo apt update
```

Para finalizar instale o docker

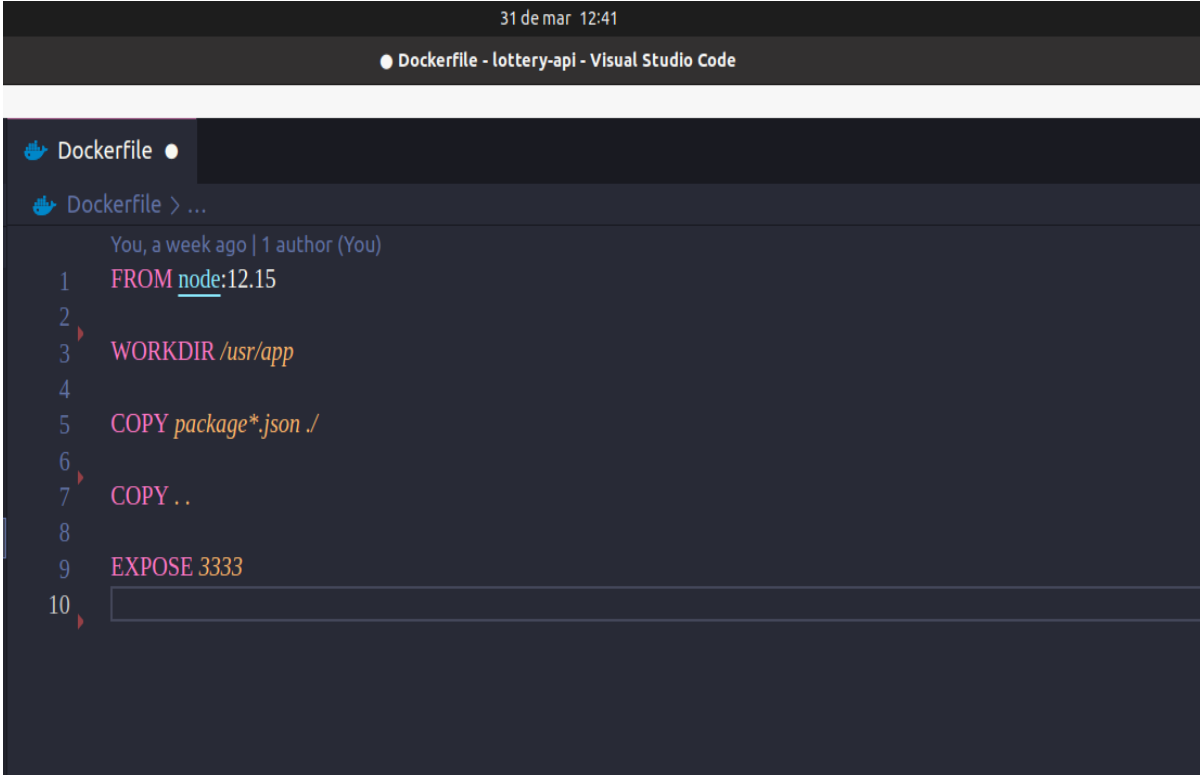
```
sudo apt install docker-ce
```

Fonte: o autor

### Configuração das imagens

Após o download da imagem no repositório, é necessário a criação de um arquivo para indicar sua forma de uso. Esse arquivo obrigatoriamente deve ser nomeado como Dockerfile, além de ser único na raiz do projeto.

## Dockerfile 3 – Configuração do Dockerfile



```
31 de mar 12:41
Dockerfile - lottery-api - Visual Studio Code

Dockerfile ●
Dockerfile > ...
You, a week ago | 1 author (You)
1 FROM node:12.15
2
3 WORKDIR /usr/app
4
5 COPY package*.json ./
6
7 COPY . .
8
9 EXPOSE 3333
10
```

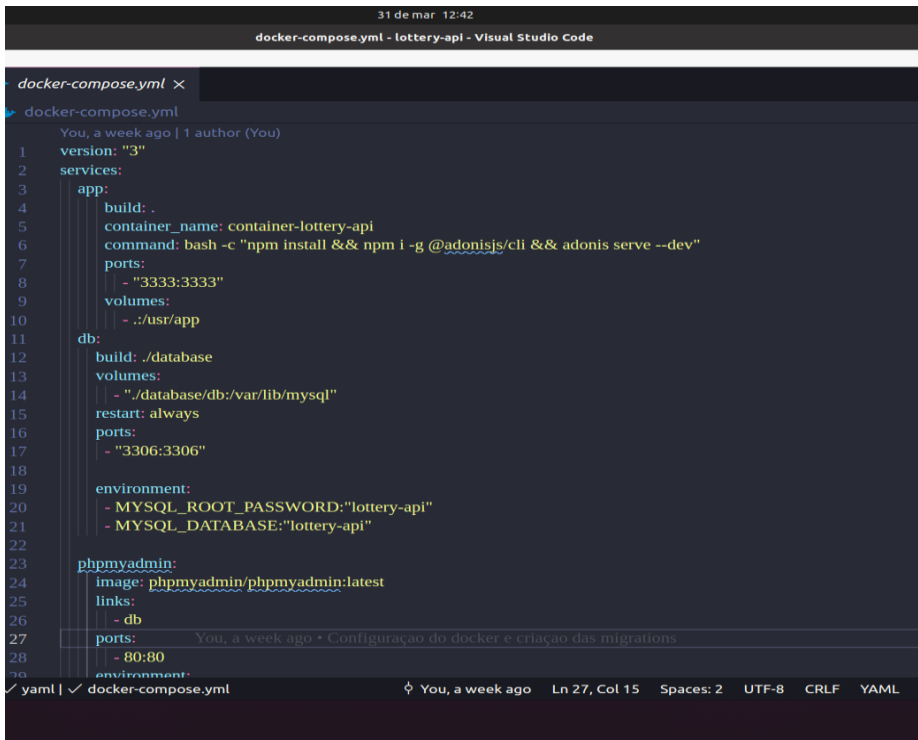
Fonte: site: <https://i1.wp.com/n8d.at/wp-content/uploads/2017/03/Dockerfile-to-create-sfx-docker-image-2.png?w=865&ssl=1> (2021)

Na imagem Dockerfile 3, temos os comandos para executar a imagem do Node, fazer a cópia do sistema local para o container virtualizado e expor as portas entre o host e o container.

### Configurando o docker compose

A configuração é bem simples, e dependendo da complexidade da aplicação. Da mesma forma que o Dockerfile, o Docker Compose precisa ser criado na raiz projeto, porém, apenas é obrigatório informar para o nome do arquivo a extensão YML.

## docker-compose 4 – Configurações gerais



```

31 de mar 12:42
docker-compose.yml - lottery-api - Visual Studio Code

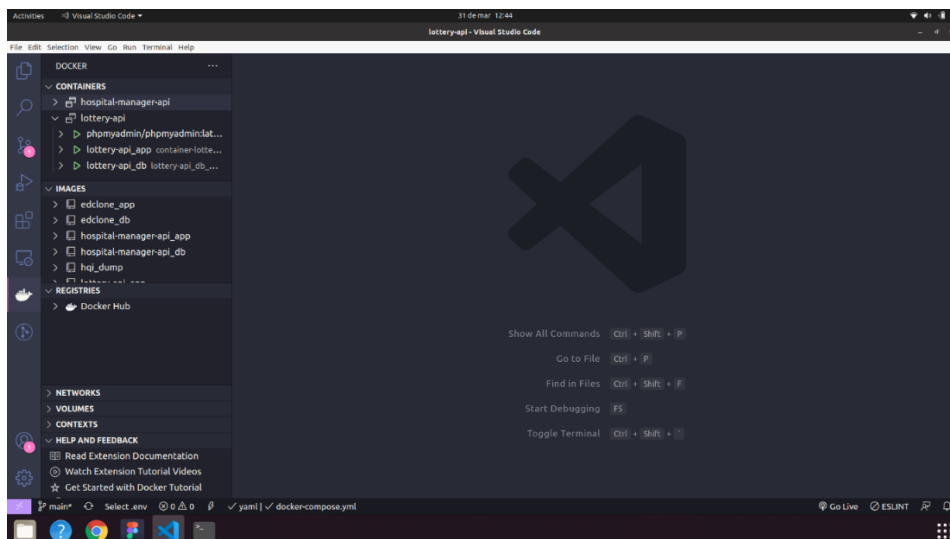
docker-compose.yml
You, a week ago | 1 author (You)
1 version: "3"
2 services:
3   app:
4     build: .
5     container_name: container-lottery-api
6     command: bash -c "npm install && npm i -g @adonisjs/cli && adonis serve --dev"
7     ports:
8       - "3333:3333"
9     volumes:
10      - ./usr/app
11   db:
12     build: ./database
13     volumes:
14       - ./database/db:/var/lib/mysql"
15     restart: always
16     ports:
17       - "3306:3306"
18
19     environment:
20       - MYSQL_ROOT_PASSWORD:"lottery-api"
21       - MYSQL_DATABASE:"lottery-api"
22
23   phpmyadmin:
24     image: phpmyadmin/phpmyadmin:latest
25     links:
26       - db
27     ports:
28       - 80:80
29     environment:
30
yml | ✓ docker-compose.yml
You, a week ago Ln 27, Col 15 Spaces: 2 UTF-8 CRLF YAML

```

Fonte: o autor

Na imagem docker-compose 3, temos os comandos para orquestrar os containers através do docker compose. No exemplo citado, temos as imagens do Node, Phpmyadmin e Mysql, além da criação de um volume para armazenar os dados. Na imagem Editor Visual Studio Code 4, vemos os containers em execução através do editor de códigos do Visual Studio Code.

Editor Visual Studio Code 5 - Container em execução.



Fonte: autor

## **4 RESULTADOS E DISCUSSÕES**

A principal vantagem da virtualização com docker foi o fato de poder utilizar as mesmas ferramentas de desenvolvimento em qualquer Sistema Operacional. Ainda que, aconselhado o uso em um ambiente Linux, devido seu kernel, já que nos demais S.O é precisar baixar uma máquina virtual, o que gera perda de desempenho.

Podemos observar que o uso do docker compose facilitou bastante o trabalho de gerenciamento dos módulos e que com poucos comandos os containers são içados. Evitando executar as imagens uma a uma daria muito trabalho.

Ficou evidente a melhora na escalabilidade e no compartilhamento da aplicação a partir do momento em que se passa a utilizar a containerização, isso porque não se precisa instalar na máquina local os mesmos recursos que estão nos containers. lembrando ainda que a máquina hospedeira(host) precisa dispor de recursos de hardware suficientes para suportar, além da instalação, a execução.

## **5 CONSIDERAÇÕES FINAIS**

Como descrito no exemplo prático, foi escolhido o Docker compose, porém, é valido em testes futuros, testamos outros orquestradores como Kubernetes, Docker Swarm, OpenShift e AWS ecs. Tais ferramentas recebem constantes atualizações, o que normalmente sempre traz novos recursos e melhorias na performance.

Neste trabalho foi demonstrado os passos para obtenção das imagens e criação dos containers. E apesar da pratica ser simples, nota-se o quão útil é a plataforma, através da sua alta escalabilidade e fácil gerenciamento dos módulos. Onde o foco foi demonstrar uma visão introdutória sobre a ferramenta e suas particularidades no cenário de desenvolvimento de software.

## REFERÊNCIAS

JUNIOR, C.S.F, LIMA, S.M.N, B. (2020) **TECNOLOGIA DOCKER: otimizando tempo e recursos no ambiente de desenvolvimento abordagem introdutória.**

FERREIRA, D.M.P, (2021) **Análise e comparação de Desempenho de Containers em Docker e em Apache Mesos** – Universidade Beira Interior, Covilhã - Portugal.

PINTO, W.D, Pereira, F.C. (2019) **DOCKER – CONTAINERS NÃO SÃO VM's.** 2º Seminário de Tecnologia Gestão e Educação – Faculdade Alcides, Porto Alegre – Rio Grande do Sul.

ANDRADE, M. (2016). **Usando o Docker.** Editora Novatec, São Paulo – SP – Brasil

NOLETO, Cairo. **Docker: O que é, como funciona e suas vantagens!** (2020). Disponível em: <<https://blog.betrybe.com/tecnologia/docker/>>. Acesso em: 30 de Mar. de 2021.

TURNBULL, James. (2014) **The Docker Book: Containerization is the new virtualization.**

**Quais são os orquestradores de containers mais conhecidos e utilizados?**

Disponível em:

<<https://cl9.com.br/quais-sao-os-orquestradores-de-containers-mais-conhecidos-e-utilizados/>> Acesso em: 30 de Mar. de 2021.

**Container Docker: O que é e quais são as vantagens em usar?** (2017). Meu Positivo. Disponível em:

<<https://www.meupositivo.com.br/panoramapositivo/container-docker/>> Acesso em: 30 de Mar. de 2021.

TRUCCO, Cristian. **Docker Compose: O que é? Para que serve? O que come?** Disponível em: <<https://imasters.com.br/banco-de-dados/docker-compose-o-que-e-para-que-serve-o-que-come>> Acesso em: 30 de Mar. de 2021.