

# **BASE DE DADOS**

**LUKEBA ANTÓNIO MANUEL JOSÉ**

**DOCKER**

## **Docker**

O Docker é uma plataforma open source que facilita a criação e administração de ambientes isolados. Ele possibilita o empacotamento de uma aplicação ou ambiente dentro de um container, se tornando portátil para qualquer outro host que contenha o Docker instalado. Então, você consegue criar, implantar, copiar e migrar de um ambiente para outro com maior flexibilidade. A ideia do Docker é subir apenas uma máquina, ao invés de várias. E, nessa única máquina, você pode rodar várias aplicações sem que haja conflitos entre elas.

Vale lembrar que a tecnologia e a empresa compartilham o mesmo nome. A empresa Docker Inc. desenvolve a tecnologia com base no trabalho realizado pela comunidade do Docker. Essa comunidade trabalha gratuitamente para melhorar essas tecnologias em benefícios de todos.

### **Então, podemos dizer que o Docker é uma máquina virtual?**

O Docker é algo parecido com uma máquina virtual extremamente leve, mas não se trata de fato de uma máquina virtual. O Docker utiliza containers que possuem uma arquitetura diferente, permitindo maior portabilidade e eficiência. O container exclui a virtualização e muda o processo para o Docker. Então, não podemos dizer que o Docker é uma máquina virtual. Veja na imagem abaixo as diferenças entre o Docker e uma virtualização.

Podemos ver que, na virtualização, temos um maior consumo de recursos, uma vez que para cada aplicação precisamos carregar um sistema operacional. Já no Docker, podemos ver que não existe essa necessidade de múltiplos sistemas operacionais convidados.

### **O que são esses containers?**

Um container é um ambiente isolado. Um container contém um conjunto de processos que são executados a partir de uma imagem, imagem esta que fornece todos os arquivos necessários. Os containers compartilham o mesmo kernel e isolam os processos da aplicação do restante do sistema.

Por exemplo: se você está desenvolvendo uma aplicação para um cliente, você pode fazer suas configurações nessa aplicação... Mas, em um ambiente convencional, você precisará replicar estas configurações para os outros ambientes de execução. Com o Docker, você estará fazendo isso em um ambiente isolado e, por causa da facilidade para replicação de containers, você acaba criando ambientes padronizados, tanto em desenvolvimento como em produção, por exemplo. Assim, você pode disponibilizar essa arquitetura toda para seu cliente, onde ele estiver: basta replicar os containers, que serão executados da mesma maneira em qualquer lugar.

Como o container possui uma imagem que contém todas as dependências de um aplicativo, ele é portátil e consistente em todas as etapas de desenvolvimento. Essa imagem é um modelo de somente leitura que é utilizada para subir um container. O Docker nos permite construir nossas próprias imagens e utilizá-las como base para os containers.

Vale lembrar que, apesar do Docker ter sido desenvolvido inicialmente com base na tecnologia LXC (Linux Containers – sendo, portanto, mais associado aos containers Linux), hoje essa tecnologia tornou-se independente de sistema operacional: podemos utilizar o Docker em ambientes Linux, Windows e até mesmo MacOS.

### **Por que utilizar o Docker?**

Depois de conhecer um pouco mais sobre o Docker, você já deve ter percebido algumas vantagens de sua utilização, como economia de recursos, melhor disponibilidade do sistema (pelo compartilhamento do SO e de outros componentes), possibilidades de compartilhamento, simplicidade de criação e alteração da infraestrutura, manutenção simplificada (reduzindo o esforço e o risco de problemas com as dependências do aplicativo), entre muitas outras. Sendo assim, você terá muitos motivos e oportunidades para fazer uso do Docker.