

## O que é Docker?



Docker é uma plataforma de código aberto para desenvolver, que permite criar, executar e gerenciar aplicações em contêineres. Um contêiner é como uma “caixa isolada” que reúne tudo o que uma aplicação precisa para funcionar, como dependências e configurações. Isso garante que ela rode sempre do mesmo jeito, em qualquer lugar, no seu computador, em um servidor ou na nuvem.

### Principais pontos

- **Open Source:** Lançado em 2013 sob licença Apache 2.0, o Docker Engine é gratuito e mantido pela comunidade;
- **Padronização:** Evita o clássico problema “funciona na minha máquina, mas não funciona no servidor”;

- **Isolamento:** Cada aplicação roda em seu próprio contêiner, sem interferir nas outras;
- **Portabilidade:** Contêineres podem ser executados em diferentes sistemas (*Windows, Linux, macOS*) desde que o Docker esteja instalado;
- **Eficiência:** Contêineres são mais leves e rápidos do que máquinas virtuais, porque compartilham o kernel do sistema operacional.

O Docker é uma ferramenta essencial para desenvolvedores e equipes de TI, porque simplifica a criação de ambientes de desenvolvimento, testes e produção, trazendo agilidade e confiabilidade para os projetos. Tudo isso em cima de uma base open source, que garante liberdade de uso e constantes evolução pela comunidade.

#### Diferença entre Docker e Máquina Virtual

Embora o Docker seja frequentemente comparado a máquinas virtuais (VMs), eles funcionam de forma diferente.

- **Máquina Virtual (VM):** Precisa simular todo um sistema operacional, incluindo kernel, drivers e bibliotecas. Por isso, consome mais recursos e leva mais tempo para iniciar.
- **Docker (Contêineres):** Compartilha o kernel do sistema operacional do host, criando apenas ambientes isolados para as aplicações. Isso torna os contêineres mais leves, rápidos e eficientes.

Em resumo, enquanto uma VM carrega um sistema inteiro dentro dela, o Docker fornece apenas o que a aplicação precisa para rodar, mantendo isolamento sem o peso de um SO completo.

| Característica         | Docker (Contêiner)                       | Máquina Virtual (VM)                                       |
|------------------------|------------------------------------------|------------------------------------------------------------|
| Sistema Operacional    | Compartilha o kernel do host             | Inclui um sistema operacional completo                     |
| Consumo de Recursos    | Leve e rápido                            | Mais pesado                                                |
| Tempo de Inicialização | Segundos                                 | Minutos                                                    |
| Isolamento             | Isolamento a nível de processo           | Isolamento total com SO próprio                            |
| Uso Comum              | Desenvolvimento, testes e microsserviços | Execução de múltiplos sistemas distintos no mesmo hardware |

#### Componentes principais do Docker

Para entender melhor como o Docker funciona, é importante conhecer alguns de seus principais componentes:

- **Imagem:** É como uma fotografia ou cópia em camadas do ambiente necessário para rodar uma aplicação. Nela fica os binários ou executáveis (como no caso de um banco de dados ou de uma API

publicada em .NET), além das bibliotecas, dependências, arquivos de configuração e as instruções de inicialização.

- **Contêiner:** É uma instância em execução de uma imagem. Cada contêiner tem seu próprio sistema de arquivos isolado, processos e rede, garantindo que a aplicação rode sempre do mesmo jeito.
- **Dockerfile:** É um arquivo de instruções que descreve, passo a passo, como uma imagem deve ser construída.
- **Registry (ex.: Docker Hub):** É o repositório onde as imagens ficam armazenadas e de onde podem ser baixadas ou compartilhadas.
- **Docker Engine:** É o motor que executa e gerencia os contêineres, permitindo que você interaja com eles usando o comando docker no terminal ou via API.

## Conclusão

O Docker se consolidou como uma das tecnologias mais importantes para desenvolvedores e equipes de TI, justamente por unir praticidade, desempenho e padronização. Ao compreender a diferença em relação às máquinas virtuais e conhecer seus principais componentes, fica mais fácil perceber como os contêineres simplificam o desenvolvimento e a implantação de aplicações.

Com ele é possível criar ambientes consistentes, reproduzíveis e portáteis, reduzindo problemas de compatibilidade e acelerando o ciclo de entrega de software. Seja para aprender, testar novas ideias ou estruturar projetos em produção, o Docker é hoje uma ferramenta essencial no arsenal de qualquer profissional de tecnologia.