

Virtualização

Conceitos Gerais

1. Introdução

- Virtualização é a tecnologia que permite criar versões virtuais de recursos físicos, como servidores, sistemas operacionais, armazenamento e redes.
- Por meio de um software virtualizador (conhecido como hipervisor), é possível dividir um único hardware físico em várias máquinas virtuais independentes.
- Com isso, é possível rodar diversos sistemas operacionais e softwares em um mesmo computador físico, economizando recursos de hardware, isolar sistemas e facilitar testes e simulações.

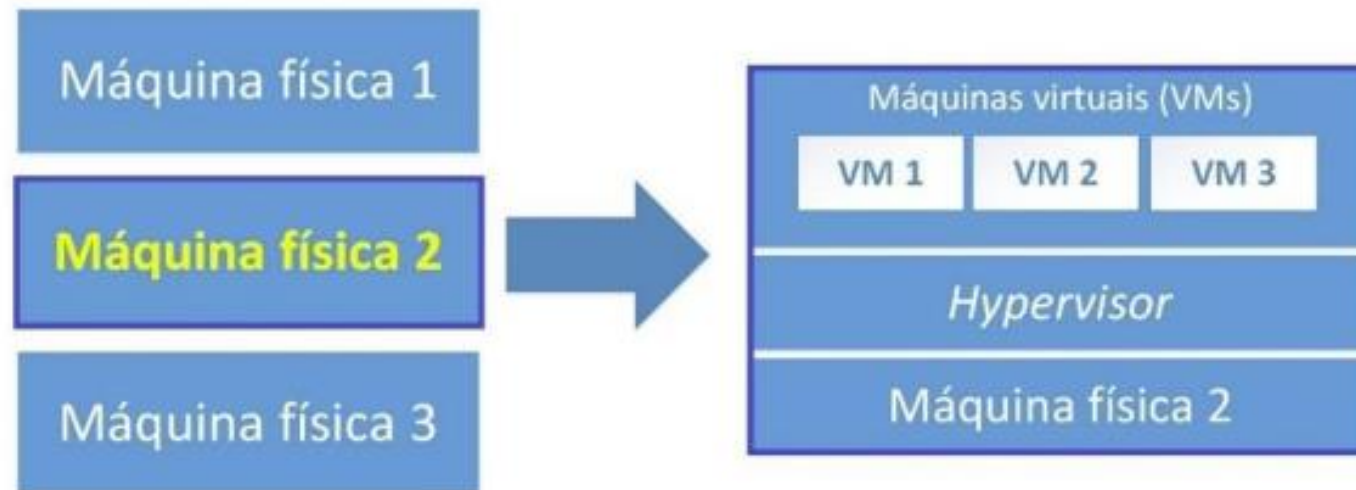
2. Virtualização

- Sabemos que os sistemas computacionais basicamente são compostos por hardware, sistema operacional e aplicações, sendo que o hardware executa as operações solicitadas pelas aplicações através do sistema operacional:



2. Virtualização

- A imagem abaixo mostra a estrutura de uma máquina física com 3 máquinas virtuais instaladas. Observe que existe um software **hypervisor** que faz a intermediação entre as máquinas físicas e virtuais:



2. Virtualização

- É possível classificar a virtualização em 3 partes:

1 - **Sistema real, nativo ou hospedeiro (host system)**: possui os recursos reais de hardware e software do sistema, ex.: processador Intel com S.O. Windows;

2 - **Camada de virtualização, hypervisor ou monitor (VMM - Virtual Machine Monitor)**: disponibiliza as interfaces virtuais e provê suporte às máquinas virtuais;

3 - **Sistema virtual ou sistema convidado (guest system)**: executa sobre o sistema virtualizado, vários sistemas virtuais podem coexistir sobre o mesmo sistema real, ex.: uma VM Linux Ubuntu e uma VM Windows XP rodando sobre um sistema operacional Windows Server 2012.

2. Virtualização

- No caso de uma máquina física que hospeda diversas máquinas virtuais precisar ser desligada para manutenção, é possível fazer a chamada migração viva.
- A **migração viva (live migration)** é um processo que move as VMs de um hypervisor para outro sem ocorrer desligamento dessas máquinas, garantindo a continuidade de negócios.
- É importante frisar que não é um processo 100% seguro, pode ocorrer travamento ou comprometimento das máquinas virtuais.

3. Níveis de Virtualização

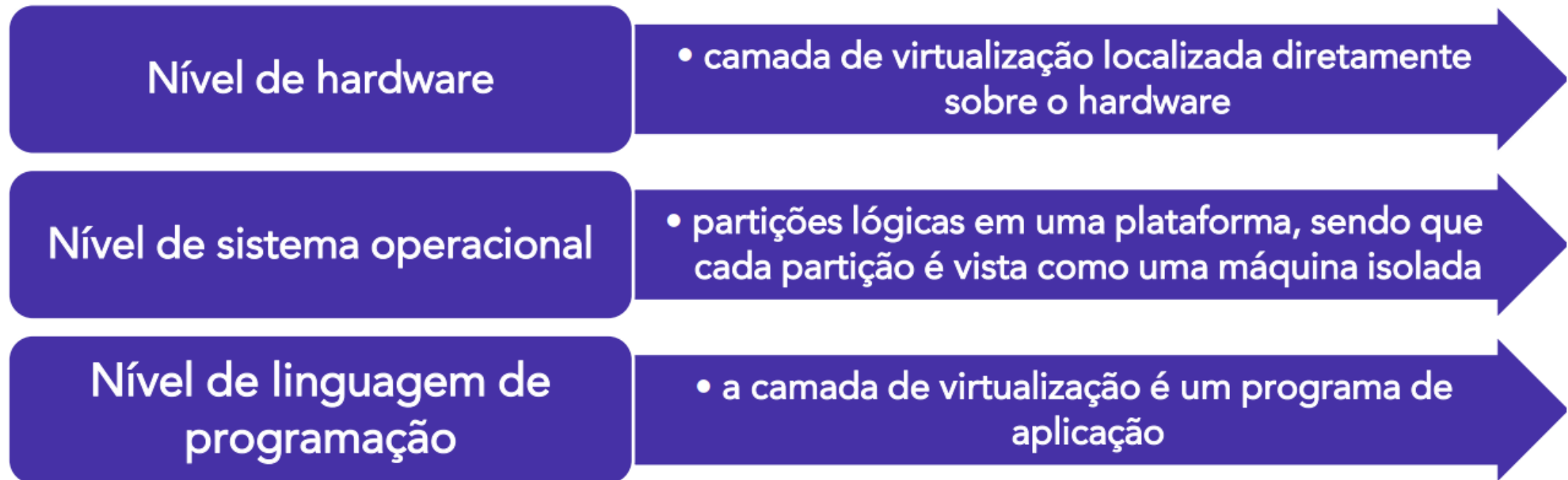
- Os níveis de virtualização definem **formas diferentes de implementar a virtualização** (ou seja, *onde* a virtualização acontece na arquitetura do sistema).
- A virtualização pode acontecer em diferentes camadas do sistema computacional. Cada nível define **como e onde a VM (ou ambiente virtual) é criada e gerenciada**. São basicamente 3 níveis:

1 - Nível de Hardware: A virtualização acontece **diretamente sobre o hardware**, usa um **hipervisor tipo 1 (bare-metal)** e não precisa de sistema operacional por baixo. Exemplos: VMware ESXi e Hyper-V (modo bare-metal). Possui alto desempenho e é muito usado em servidores.

2 - Nível de Sistema Operacional: A virtualização ocorre **dentro de um sistema operacional (com kernel compartilhado)**. Cria **ambientes isolados**, mas não VMs completas. Exemplo: Docker. Geralmente são mais leves e rápidos, com menor isolamento que a VM. Lembrando que não são VMs completas, são **containers**

3. Níveis de Virtualização

3 - **Nível de Linguagem de Programação:** A virtualização é feita por um **programa**. Cria um ambiente para executar código de uma linguagem. Por exemplo: JVM (Java Virtual Machine) ou CLR (.NET). Não virtualiza o sistema inteiro, só executa programas daquela linguagem. Possui portabilidade (roda em qualquer sistema com a VM instalada).



4. Tipos de Hypervisor

- Os principais tipos de hypervisor são:

1 - O hypervisor tipo 1 (baremetal): roda diretamente sobre o hardware, controlando o hardware e o acesso ao sistema operacional convidado (guest). Seu papel é compartilhar os recursos de hardware entre as VMs de forma que elas pensem ter recursos exclusivos (processador, memórias, mídias de armazenamento e dispositivos de E/S. Podemos dividir o baremetal em dois subtipos:

Hypervisor monolítico: necessita de grande quantidade de código entre o hardware e as VMs, pois emula todo o hardware para as VMs. Os drivers estão no próprio hypervisor (como se fosse um S.O. com todos os drivers necessários);

Hypervisor microkernelizado: utiliza drivers em cada VM, tendo o hypervisor como única camada entre o S.O. guest e o hardware. Ou seja, cada VM que “se vire” em buscar os drivers necessários, o hypervisor fica mais enxuto, tornando-se mais seguro (superfície mínima de ataque).

4. Tipos de Hypervisor

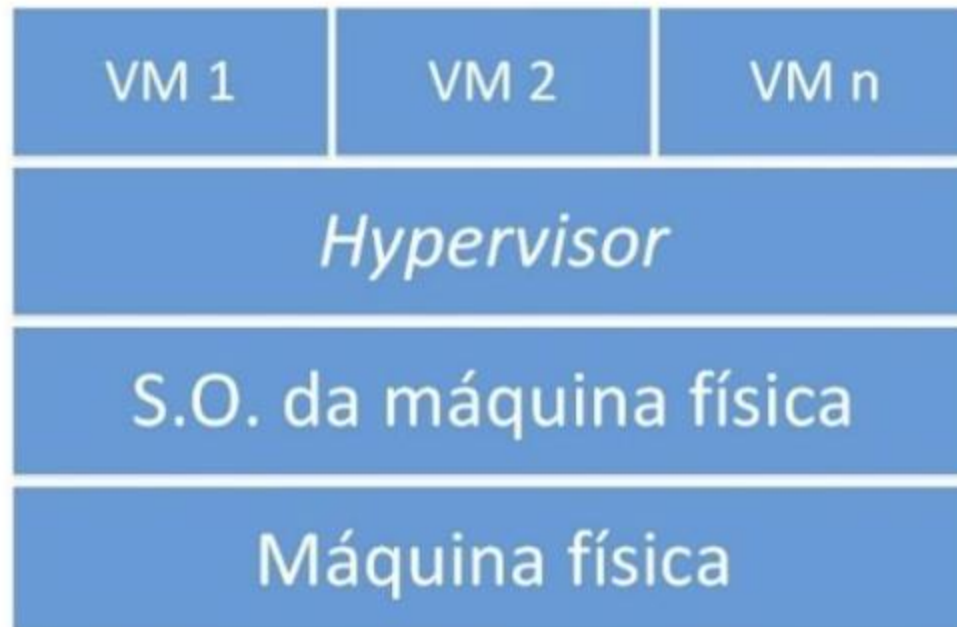
- A imagem abaixo mostra os dois subtipos de hypervisor baremetal (monolítico e microkernel):



4. Tipos de Hypervisor

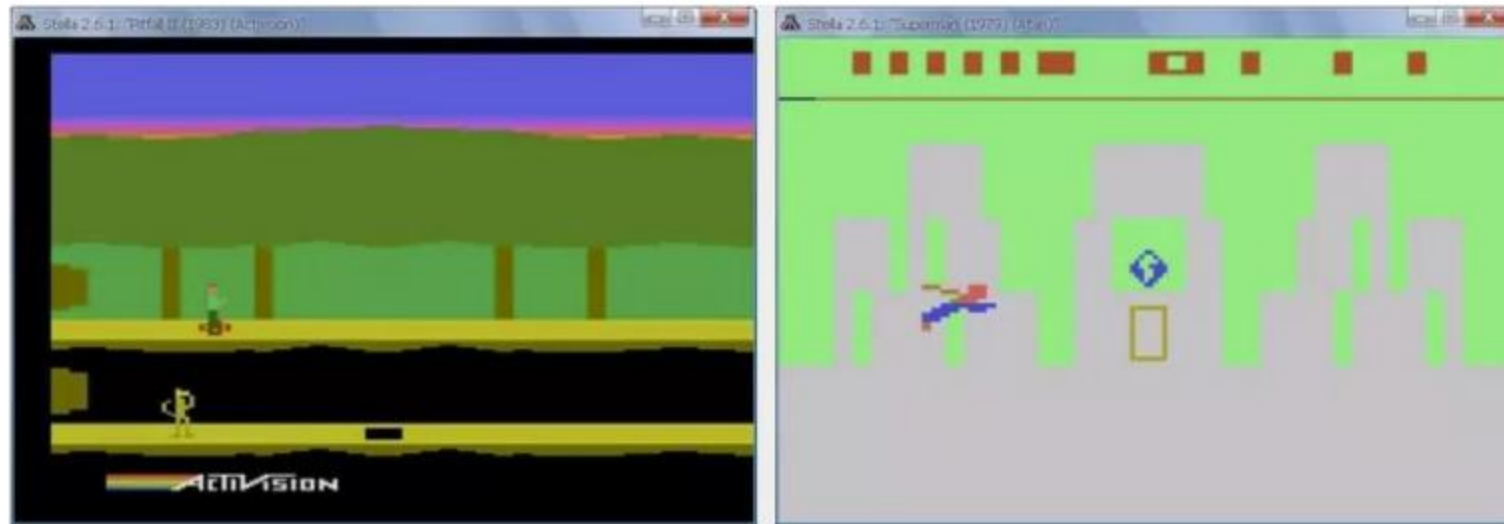
2 – Hypervisor tipos 2 (hosted): Não é tão baixo nível como o anterior, é um hypervisor que funciona como uma aplicação que fornece um ambiente de execução para outras aplicações. Necessita de um SO nativo.

Também é criado um hardware virtual sobre os recursos de hardware oferecidos pelo sistema operacional nativo. Como exemplo, temos o Virtualbox, VMware Player, VMware Server.



5. Emulador

- Emulador é um software que permite a que um software feito para uma arquitetura seja executado em outra. Um exemplo clássico são os emuladores de videogame, como o emulador Stella (emulador do videogame Atari).



6. VMWare

- O **VMware vSphere** é uma plataforma de virtualização completa, com um conjunto abrangente de serviços de infraestrutura e aplicativos.
- Essa plataforma de virtualização permite que os usuários virtualizem aplicativos de dimensionamento vertical e horizontal com confiança, redefine o significado de disponibilidade e simplifica o data center virtual.
- O resultado é uma infraestrutura altamente disponível, resiliente e sob demanda que é a base de qualquer ambiente de nuvem.

6. VMWare

- A arquitetura do VMware vSphere Hypervisor fornece uma camada de virtualização de alto desempenho, permitindo que várias máquinas virtuais (VMs) compartilhem recursos de hardware com desempenho que pode corresponder ao throughput original e, em alguns casos, excedê-lo.
- O VMware vSphere Virtual Symmetric Multiprocessing permite a utilização de VMs ultrapotentes, que tenham mais de 128 CPUs virtuais.
- O VMware vSphere Virtual Machine File System (VMFS) permite que as VMs acessem dispositivos de armazenamento compartilhados (Fibre Channel, iSCSI, entre outros) e até mesmo storages.
- O VMware vSphere vMotion possibilita a migração em tempo real de máquinas virtuais entre servidores, sem interrupção para usuários ou a perda de serviço.

6. VMWare

- O VMware vSphere High Availability (HA) permite a reinicialização automática, em poucos minutos e com um excelente custo-benefício para todos os aplicativos.
- O VMware vSphere Data Protection é uma solução de backup e replicação. Fornece backups de armazenamento eficiente por meio de eliminação de duplicação.
- O VMware vSphere Distributed Resource Scheduler (DRS) fornece balanceamento de carga dinâmico e independente de hardware.
- O VMware vSphere Distributed Power Management automatiza a eficiência de energia nos clusters.
- O VMware vSphere Big Data Extensions executa o Hadoop no vSphere para obter utilização, confiabilidade e agilidade mais avançadas. Permite gerenciar cargas de trabalho do Hadoop em uma plataforma comum sem interrupções.

7. XEN

- É um **hypervisor tipo 1 (bare-metal)**. Instalado direto no hardware e usado para criar e gerenciar máquinas virtuais. É de uso livre e gratuito. **Xen NÃO é o produto final, ele é o “motor” de virtualização.** Várias plataformas usam esse “motor” para criar seus próprios virtualizadores.
- No geral, as plataformas usam o motor XEN, e criam interface gráficas, recursos e suporte para monetizar ou não sua ferramenta.
- O **XCP-ng** é um dos mais usados com base XEN, sendo totalmente gratuito. Já o **Citrix Hypervisor** (antigo XenServer) é uma ferramenta paga com base XEN.
- O XEN atualmente possui recursos de live migration (migração de VM em execução), possui alto desempenho, isolamento entre VM's.

7. XEN

- Existe uma VM especial chamada **Dom0**, que controla o Disco, Rede e Criação de VMs, de forma que possa oferecer recursos para as demais máquinas virtuais.
- O Xen suporta **Paravirtualização (PV) e Virtualização total (HVM)**. Na paravirtualização o SO da VM sabe que está virtualizado. Desta forma, ele não tenta acessar o hardware direto, acessa via hypervisor.
- A paravirtualização é mais rápida e com menos overhead. Mas precisa **modificar o sistema operacional**, por isso funciona bem com Linux e não funciona bem em Windows antigo.
- Já o HVM funciona em qualquer SO, pois o processador tem suporte nativo (Intel VT-x/AMD-V). Desta forma, a CPU ajuda a controlar as VMs. No geral, tem um desempenho um pouco menor.

8. Hyper-V

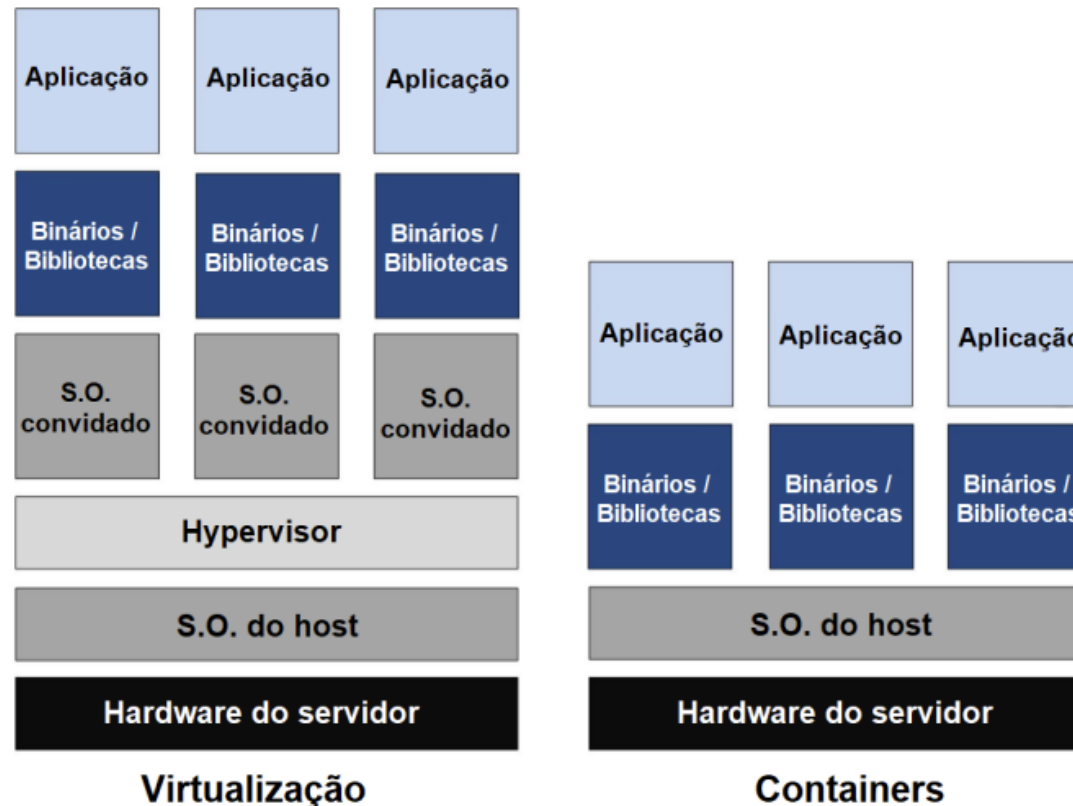
- O Hyper-V foi lançado junto com o Windows Server 2008 e está disponível sem custo adicional para todos os Windows Server e Windows 8 e posteriores. Anteriormente ao Windows 8, era utilizado o Windows Virtual PC como componente de virtualização para clientes do Windows NT.
- Hyper-V é uma tecnologia que permite a virtualização do hardware em um computador físico. Ou seja, é possível criar e gerenciar computadores virtuais e seus recursos, onde cada máquina virtual (VM) é considerada um sistema isolado (o conceito clássico de VM, como já vimos).
- O Hyper-V é baseado no hypervisor, uma camada adicional entre os recursos físicos e virtuais, responsável pelo gerenciamento, para que os recursos de hardware da máquina sejam distribuídos eficientemente entre as VMs.

8. Hyper-V

- O Hyper-V possui recursos de snapshot (salvar estado e voltar ao estado se necessário), ajuste de memória automático, switch virtual (possibilita comunicação das VM's entre elas e externamente), possui criptografia contra acesso indevido.
- Possui recursos de réplica de VM, o que auxilia em backup e recuperação de desastres. Pode ser gerenciado via gerenciador do Hyper-V ou com Power Shell.
- Funciona muito bem com o Windows Server, e permite criar máquinas virtuais dentro de máquinas virtuais.

9. Docker/Container

- O Docker não é um sistema de virtualização tradicional. Enquanto na virtualização tradicional nós temos um sistema operacional completo e isolado, dentro do Docker nós temos recursos isolados que utilizam bibliotecas de kernel em comum (entre host e container).



9. Docker/Container

- O Docker é uma ferramenta que permite rodar aplicações em **containers** — ambientes leves e isolados. Em vez de criar uma máquina virtual inteira (com outro sistema operacional), o Docker cria “**mini ambientes**” só com o que o programa precisa.
- Um container é como uma **caixinha pronta** para uso com Aplicação, Bibliotecas e Dependências. Tudo já configurado e funcionando igual em qualquer lugar.
- Como exemplo, vamos pensar em um sistema Python que utiliza uma série de bibliotecas. É possível criar um container com o código, bibliotecas e interpretador Python na versão utilizada pelo programa.
- Este container poderá ser utilizado em qualquer máquina que tiver o Docker instalado.

9. Docker/Container

- O Docker possui algumas desvantagens, pois compartilha o sistema operacional, se o host tiver problema, afetará os containers. No geral, a segurança é menor que na VM.
- Orquestrar muito containers em uma aplicação em produção também não é uma atividade muito fácil. Por isso, os DevOps são muito bem remunerados atualmente.
- Não é recomendado usar o Docker em ambientes que precisam de Isolamento forte (melhor usar VM), rodar outro sistema operacional completo, aplicações muito acopladas ao hardware e Ambiente simples demais (pode ser “overkill”).

10. Big Data

- **Big Data** é o uso de tecnologias para **armazenar, processar e analisar grandes volumes de dados** que não cabem em sistemas tradicionais.
- Hoje geramos dados o tempo todo em Redes sociais, Compras online, Sensores, Apps entre outros. Isso gera **dados em quantidade enorme**. Big Data serve para **transformar esses dados em informação útil**.
- **Para ser considerado um Big Data, deve obedecer os chamados 5V's:**

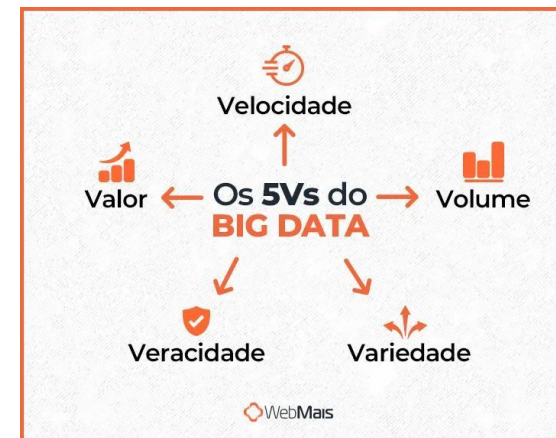
1 - Volume → muitos dados

2 - Velocidade → dados chegando rápido

3 - Variedade → vários formatos (texto, vídeo, etc.)

4 - Veracidade → qualidade dos dados

5 - Valor → gerar informação útil



10. Big Data

- O Big Data segue os seguintes processos de funcionamento:

1 - Coleta de dados: Dados vêm de várias fontes: Sites, Sensores, Sistemas, etc.

2 – Armazenamento: Os dados são guardados em sistemas distribuídos. Não fica tudo em um só computador. Fica distribuído em vários servidores

3 – Processamento: Os dados são processados em paralelo, com vários computadores trabalhando juntos. Muito mais rápido.

4 - Análise: Extração de informações por meio de Relatórios, Previsões e Padrões.

5 - Visualização: Mostrar os resultados em Gráficos e Dashboards.

10. Big Data

As principais ferramentas do Big Data são:

1. **Hadoop:** Um sistema para armazenar grandes volumes de dados em vários computadores. Ele divide os dados em pedaços e espalha pelos servidores. Muito usado para armazenamento distribuído.
2. **Apache Spark:** Ferramenta para processar dados muito rápido. Processa dados em memória, então é bem mais rápido que o Hadoop tradicional. Usado para análise e processamento
3. **MapReduce:** Modelo de processamento do Hadoop. Divide o trabalho em partes (Map) e junta resultados (Reduce). Base do processamento distribuído clássico. Atualmente é o Spark é mais usado que o MapReduce.
4. **Apache Hive:** Ferramenta para consultar dados em Big Data usando SQL.
5. **Apache Pig:** Ferramenta para processar dados com uma linguagem simples (Pig Latin). Mais fácil que programar MapReduce direto. Usado para manipulação de dados. Geralmente usamos mais o Hive.
6. **Power BI ou Python:** Para gerar Dashboards e Gráficos

10. Big Data

- Com o advento da inteligência artificial, o Big data ganhou muita importância. O Big Data coleta e armazena muitos dados, que são organizados e processados. Em seguida, a IA usa esses dados para treinar modelos e começar a prever e automatizar decisões.
- Exemplos reais de sucesso no uso de Big Data e IA: Netflix, Google Maps, Uber, Amazon, Youtube, entre outros.
- O Big Data também pode apresentar desvantagens, são elas:
 - 1 - Falta de segurança → vazamentos
 - 2 - Falta de privacidade → uso indevido
 - 3 - Falta de ética → decisões questionáveis
 - 4 - Interpretação errada → conclusões perigosas

10. Big Data

- Um dos grandes problemas é o de segurança, pois como o Big Data possui muitos dados centralizados, pode ser um alvo em potencial. Já houveram casos reais de invasão:

1 – Equifax: Grande empresa global de dados e análise. Teve dados de 140 milhões de usuários vazados. Prejuízo bilionário.

2 – Facebook: Dados de milhões de usuários expostos (telefone, localização, etc.). Além do caso da Cambridge Analytics.

3 – eBay: Milhões de contas invadidas.

4 – Yahoo: Invadiram bilhões de contas, inclusive email.

11. Big Data e Máquinas Virtuais

- Ferramentas como Hadoop e Apache Spark precisam de vários nós (máquinas) para o processamento distribuído. Neste ponto, pode ser útil a utilização de VMs.
- É possível criar diversos servidores, cada um em uma máquina virtual, que forma uma estrutura de Big Data.
- Também podemos ter containers que rodam serviços específicos, como Spark, Hive.