



Big Data Fundamentos

Fonte: [Data Science Academy](#)

1. INTRODUÇÃO ✓

2. O QUE É BIG DATA?

3 SISTEMA DE ARMAZENAMENTO DE DADOS: VOLUME

Como armazenamos Big Data?

Bancos de Dados Relacionais x Bancos de Dados Não Relacionais (NoSQL)

DATA WAREHOUSES

DATA LAKES

DATA STORES

SISTEMAS HÍBRIDOS DE ARMAZENAMENTO

4. ARMAZENAMENTO E ESPAÇO PARALELO

Cluster de computadores

Armazenamento Paralelo

Software de armazenamento paralelo - Apache Hadoop

Processamento paralelo de Big Data

Arquitetura de Armazenamento e Processamento Paralelo

Soluções de Armazenamento e Processamento Paralelo

5. CLOUD COMPUTING

PRINCIPAIS PROVEDORES EM NUVEM:

CONHECENDO O AWS (AMAZON WEB SERVICES) - demonstração:

6. MLOps (OPERAÇÕES DE MACHINE LEARNING) e DataOps OPERAÇÕES COM DADOS)

MACHINE LEARNING

DevOps, MLOps, AIOps, DataOps

BIG DATA x SMALL DATA

7. DADOS COMO SERVIÇOS (DaaS)

ARQUITETURA DaaS

PRINCIPAIS BENEFÍCIOS

ARQUITETURAS MODERNAS DE BIG DATAS

DATA MESH COMO PARADIGMA DE ARQUITETURA DE DADOS

SOLUÇÕES COMERCIAIS

8. ETL - Extração, Transformação e Carga de Dados

ETL x ELT (a evolução do ETL)

PRINCIPAIS SOLUÇÕES DE ETL E ELT NO MERCADO

DEMO ELT E BIG DATA

9. COMO INICIAR UM PROJETO BIG DATA?

CASOS DE USO DE BIG DATA ANALYTICS

COMO INICIAR UM PROJETO DE BIG DATA?

1. INTRODUÇÃO

2. O QUE É BIG DATA?

Definição: coleção de conjunto de dados, grandes e complexos, que não podem ser processados por bancos de dados ou aplicações de processamentos tradicionais. Por isso, necessitam de ferramentas especialmente preparadas para lidar com grandes volumes, velocidade e variedade, de forma que toda e qualquer informação disponível nos dados possa ser encontrada, analisada e aproveitada em tempo hábil.

- As técnicas existem há décadas, mas só agora existe um volume tão grande para que sejam colocadas em prática;
- Estima-se a geração de 2,5 quintillionbytes de dados por dia;
- V's do Big Data: Volume, Velocidade (de geração de dados), Variedade (formato de dados), Veracidade (confiabilidade)(Definição IBM);
- Relação entre Big Data e Ciência de Dados: Big Data é a matéria prima (dados) para ao conjunto de técnicas de análise (ciência);
- Processos de aplicação: extrair, armazenar, processar e analisar;

3 SISTEMA DE ARMAZENAMENTO DE DADOS: VOLUME

- 3 perguntas essenciais:
 - Como vamos armazenar grandes conjuntos de dados?

- Como vamos acessar grandes conjuntos de dados armazenados (o valor está na análise e não no dado bruto armazenado)?
- Precisamos realmente armazenar tudo? (tudo começa com a definição do produto)

Como armazenamos Big Data?

- Regra geral:
 - **Data Warehouse** para dados estruturados ou que podem ser estruturados antes do armazenamento;
 - **Data lake ou Data Store** para dados não estruturados ou que não podem ser estruturados antes do armazenamento.
 - Sistema híbrido: divide por camadas e faz as duas coisas.
 - Primeiro precisa saber o que a empresa precisa para depois decidir a tecnologia aplicada

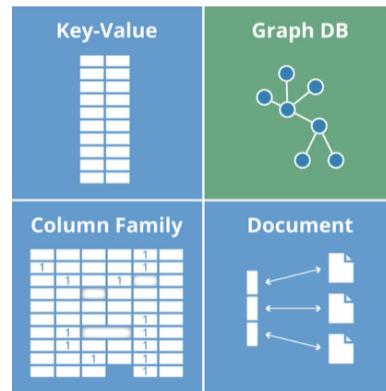
Bancos de Dados Relacionais x Bancos de Dados Não Relacionais (NoSQL)

BANCOS DE DADOS RELACIONAIS

- São bancos de dados estruturados e com **schema** (organização de dados) bem definido;
- O **schema** é definido e criado antes do armazenamento dos dados (define, cria no software gerenciador e depois carrega os dados);
- Um **Data Warehouse**, por exemplo, é criado com alguma tecnologia de banco de dados relacional com SGDB (Sistema Gerenciador de Banco de Dados) Oracle, IBM DB2, Microsoft SQL Server, MySQL, PostgreSQL e muitos outros;
- São organizados em tabelas que se relacionam

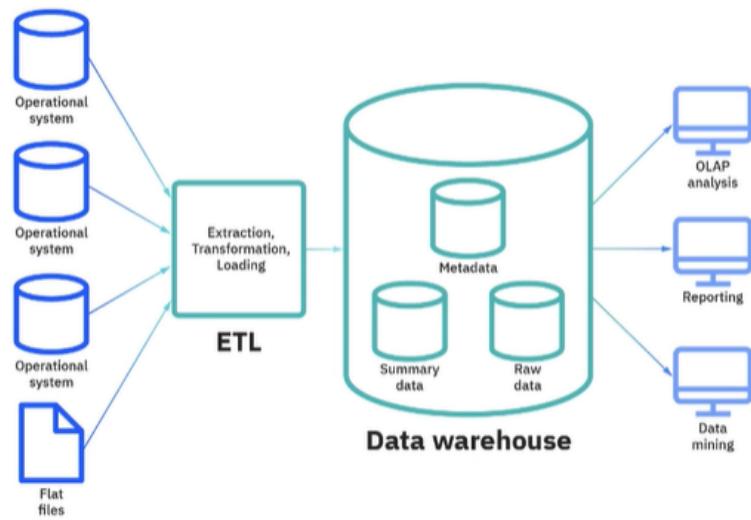
BANCOS DE DADOS NÃO RELACIONAL (NoSQL)

- Partem do princípio que os dados podem ser semi ou não estruturados e que outros tipos de relacionamentos podem existir entre os dados;
- Podem ser usados para construir **Data Lakes** e **Data Stores**;
- Normalmente não precisamos definir o **schema** antes do armazenamento ou o **schema** é definido no momento do armazenamento;
- Existem diversos tipos de bancos de dados NoSQL (ex: mongodb)



DATA WAREHOUSES

- Mais utilizado no mercado;
- Sistema de armazenamento que conecta e harmoniza grandes quantidades de dados de muitas fontes diferentes;
- Pode ser implementado usando um banco de dados relacional especificamente;
- O objetivo do DW é alimentar a inteligência de negócios: relatórios e análises, e oferecer suporte para os requisitos de negócio para transformar dados em insights e tomar decisões inteligentes baseadas em dados;
- Armazena dados atuais e históricos em um único lugar e atuam como a única fonte de informações confiáveis para uma organização;



Processo ETL: extrai dados das fontes e aplica algum processo de transformação, provavelmente já preparando os dados de acordo com o schema que foi definido previamente, e então carregamos os dados no DW. Então conecta no DW com a ferramenta de análise (PowerBI por exemplo) e extrai o relatório necessário para o tomador de decisão.

- Empresas grandes costumam ter um banco de dados central e pequenos datamartes que são porções dele específicas para as diversas áreas.

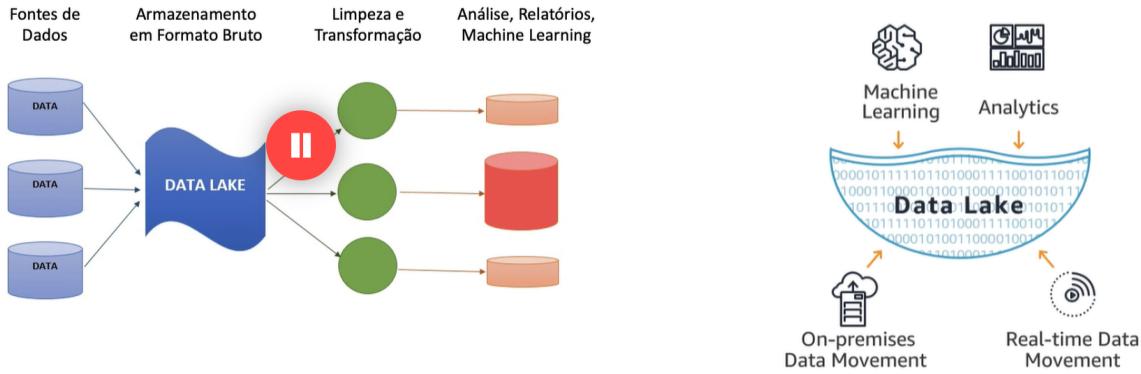


- Periodicidade de carregamento de dados depende de cada empresa, diária, semanais...
- Os dados fluem para um DW a partir de sistemas transacionais (como ERP e CRM), bancos de dados e fontes externas, como sistemas de parceiros,

dispositivos de Internet das Coisas (IoT), aplicativos de mídia social - geralmente em cadência regular;

- Surgimento da computação em núvem causou uma transformação no cenário. Nos últimos anos os locais de armazenamento mudaram para vários locais, incluindo nuvens privadas e públicas;
- O **schema** deve ser definido antes do processo de armazenamento dos dados, sendo local ou na nuvem (ou seja, criar algum processo de estrutura);
- Os DW modernos são projetados para lidar com dados estruturados e não estruturados (vídeos, imagens e dados de sensor (embora **Data Lakes** ainda sejam opções melhores para dados não estruturados));
- Alguns aproveitam a análise integrada e a tecnologia do banco de dados in-memory (na memória do computador ao invés do armazenamento em disco) para fornecer acesso em tempo real a dados confiáveis e impulsionar a tomada de decisões (dessa forma o acesso é feito mais rapidamente);
- Sem DW é muito difícil combinar dados de fontes heterogêneas, garantir que estejam no formato certo para análise e obter uma visão atual e de longo alcance dos dados ao longo do tempo.
- **Para o DW** normalmente usamos DTL (Extração, Transformação e Carga);
- **Benefícios:**
 - **Melhor análise de negócio:** dados de várias fontes, informações completas;
 - **Consultas mais rápidas:** são construídos pra isso, recuperação e análise rápida com pouco ou nenhum suporte de TI (DW não podem ter dados em excesso para não ficarem lentos, é bom fazer otimização dos dados);
 - **Melhoria da qualidade dos dados:** antes de serem carregados no DW passam por um processo de limpeza, garantindo que sejam transformados em um formato consistente para apoiar análises - e decisões - com base em dados precisos e de alta qualidade, isso é feito na hora de scriar o **schema**;
 - **Visão histórica:** ao armazenar dados históricos ricos, um DW permite que tomadores de decisão aprendam com tendências e desafios passados e façam previsões.

DATA LAKES



- Principal característica: armazenar **dados brutos** sem limpeza ou transformação prévia, porém para analisar, isso deve ser feito (mas também armazena dados estruturados não prontos para análise - tem alternativas para q seja possível a análise);
 - Na limpeza corre-se o risco de perder dados, por isso às vezes esse modelo é melhor;
 - Uma empresa pode precisar de um **Data Lake** (normalmente porta de entrada de dados) e um **Data Warehouse**, porque atendem a casos de uso diferentes;
 - Armazena tudo no formato bruto sem necessidade de já saber o que será feito com eles. Dados devem ser capturado rápidos para não serem perdidos;
 - SQL pode ser usado neles (ex apache hive);
 - Permite que as empresas gerem diferentes tipos de percepções sobre os dados, desde relatórios sobre dados históricos até modelos preditivos criados com Machine Learning;
 - **Principal desafio:** dados brutos são armazenados sem supervisão de conteúdo, por isso, precisa ter mecanismos definitos para catalogar e proteger os dados. Se não for feito, não é possível encontrar dados, e dados confiáveis em um “**Data Swamp**” (“Pântano de Dados”) - lixo armazenado! Por isso, **Data Lakes** precisam ter governança, gestão de metadados, consistência semântica e controles de acesso.

- Pode ser construído com diferentes tecnologias, como **Apache Hadoop** ou **Banco de Dados NoSQL**;
- Dados podem ser importados no **DW** para o **Data Lake** e vice-versa;
- **Para o Data Lake** normalmente usamos ELT (Extração, Carga e Transformação);
- **Data Lakes e DWs podem fazer parte de uma grande estrutura de armazenamento chamada Data Hub.**
- **Benefícios do Data Lake:**
 - **Armazenamento em formato bruto** (mas se quiser pode limpar e transformar antes, caso tenham erros gritantes);
 - **Importação de qualquer quantidade de dados em tempo real**: economiza tempo;
 - **Repositório central para todos os dados da empresa**: os Data Lakes permitem que várias funções como Cientista de Dados, Engenheiros de Machine Learning, Analista de Dados e Analistas de Negócios acessem os dados com sua ferramenta analítica específica.
 - **Sem necessidade de Movimentação dos Dados**: análises podem ser executados sem ter que move-los para um sistema de análise separado.

DATA STORES

Os tipos mais comuns de Data Stores:

- Armazenamento de chave-valor (Redis, Memcached)
- Motor de pesquisa de texto completo (Elastic Search)
- Fila de mensagens (Apache Kafka)
- Sistema de arquivos distribuídos (Hadoop HDFS, AWS S3)



- É um repositório para armazenar e gerenciar de forma persistente coleções de dados que incluem não apenas dados estruturados, mas também tipos de

armazenamento variado, como documentos, dados no formato chave-valor, filas de mensagens e outros formatos de arquivo;

- **Sistema de armazenamento que armazena dados específicos já preparados para uma aplicação final;**
- **Benefícios:**
 - **Armazenamento variado de vários tipos de dados:** dados que não se encaixam em outros repositórios de armazenamento (necessidades específicas);
 - **Flexibilidade:** armazenamento aderente às necessidades da aplicação final;
 - **Suporte a dados semi-estruturados:** dados que possuem alguma organização prévia, mas que devem ser usados em seu formato original;
 - **Custo total menor:** por se tratar de um tipo simplificado;

SISTEMAS HÍBRIDOS DE ARMAZENAMENTO

- DWs, Data Lakes e Data Stores serão usados em conjunto, criando, assim, uma grande estrutura de armazenamento de dados, um **Data Hub**.

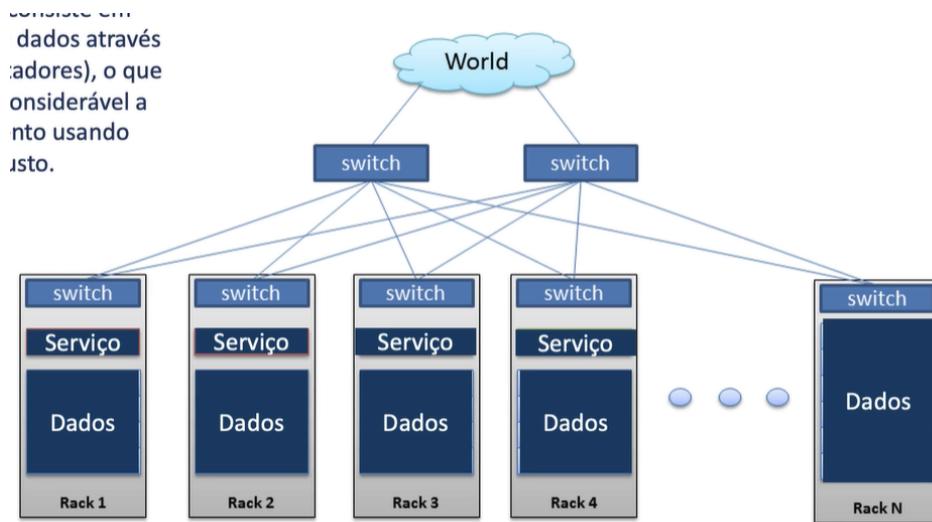
4. ARMAZENAMENTO E ESPAÇO PARALELO

Cluster de computadores

- **Servidor:** computador de alta capacidade que fornece serviços de armazenamento, aplicações ou banco de dados. Mesmo que você tenha um “super computador”, você terá um série de limitações físicas no hardware (espaço em disco, processadores e memória RAM), ou seja, não dá para pensar em um único servidor;
- Um **cluster de computadores** é um conjunto de servidores com um mesmo propósito visando fornecer um tipo de serviço como armazenamento ou processamento de dados;

- Além da escalabilidade vertical de cada máquina do cluster, possui escalabilidade horizontal (podem ser incluídas mais máquinas);

Armazenamento Paralelo

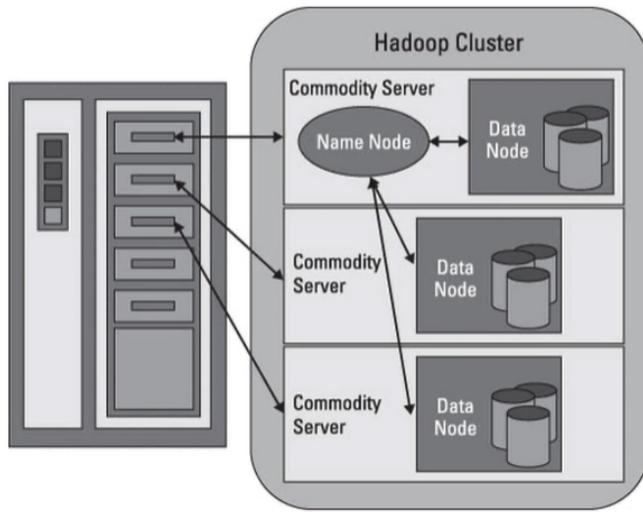


- Consiste em distribuir o armazenamento de dados através de vários servidores (computadores), aumentando a capacidade de armazenamento usando hardware de baixo custo;

Software de armazenamento paralelo - Apache Hadoop

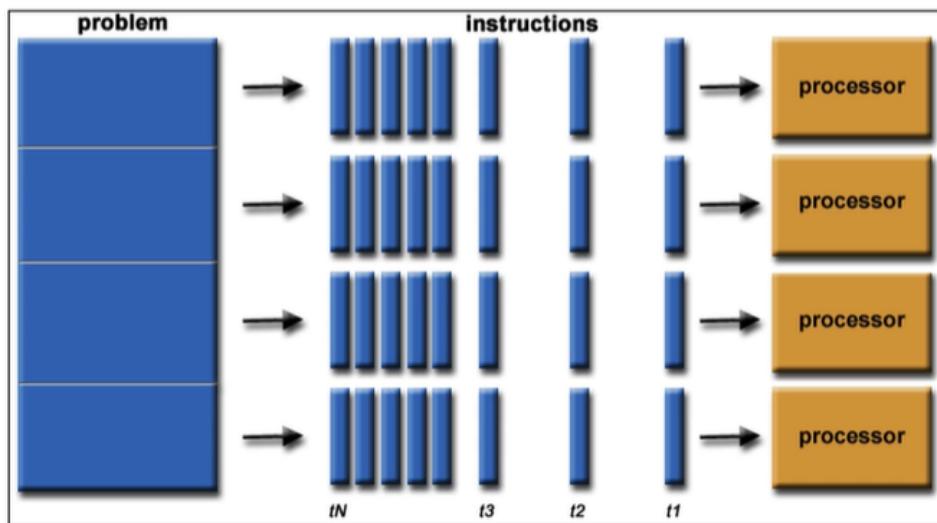
- É a forma de gerenciar o armazenamento paralelo através de diversos computadores;
- Precisa-se de um sistema de arquivos distribuído (que faz interação entre você e o seu hardware). Os computadores pessoais tem, mas não foram desenvolvidos pra isso;
- Uma das opções: **Apache Hadoop HDFS (Hadoop Distributed File System)**, responsável pela gestão do cluster de computadores definindo como os arquivos serão distribuídos através dele;
- Vantagens **HDFS**: desenvolvido na época da big data, open source, pensado para hardware commodity (de baixo custo), pensado com tolerância a falhas;

- Sobre o **HDFS** podemos construir um **Data Lake** que roda sobre um cluster de computadores e permite o armazenamento de grandes volumes com hardware commodity. Isso permitiu que o Big Data pudesse ser usado em larga escala.



Processamento paralelo de Big Data

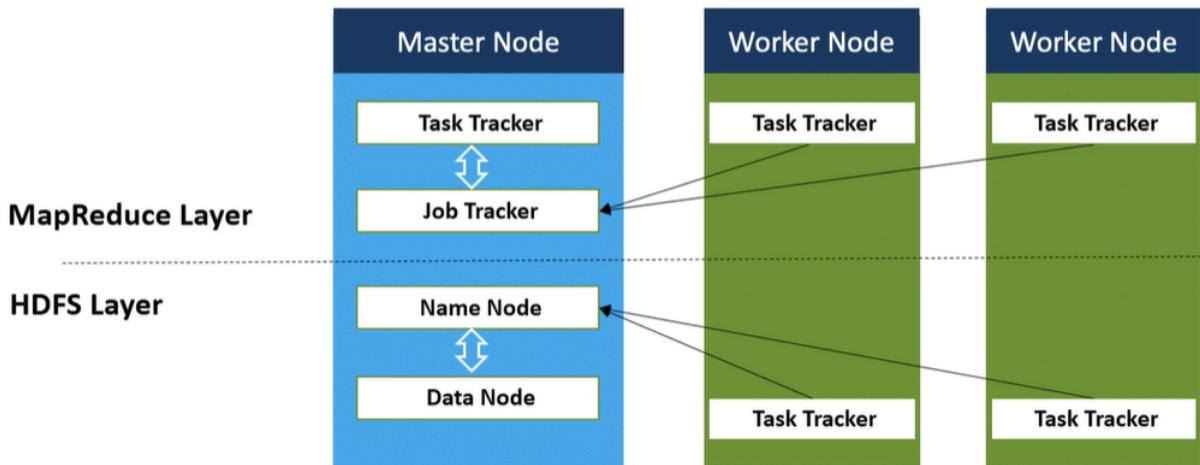
- O objetivo é dividir uma tarefa em várias sub-tarefas e executá-las em paralelo. O **Apache Hadoop MapReduce** e o **Apache Spark** são dois frameworks para esse propósito. Podem ser usados em ambiente em nuvem.



- Ao usar o framework de processamento paralelo, as sub-tarefas são levadas para o processador da máquina do cluster onde os dados estão armazenados, aumentando a velocidade de um processamento de grandes volumes de dados;

Arquitetura de Armazenamento e Processamento Paralelo

- Considerando o Apache Hadoop, teríamos o seguinte esquema:



- O **HDFS Layer** é um serviço rodando em todas as máquinas do cluster, sendo um **NameNode** para **gerenciar** o cluster e o **DataNodes** que fazem o trabalho de **armazenamento** propriamente dito;
- O **MapReduce** também é um serviço rodando em todas as máquinas do cluster, sendo um **Job Tracker** para **gerenciar o processamento** de **Task Trackers** que fazem o trabalho de **processamento**;
- O **Job Tracker** consulta o **NameNode** (que tem o índice completo) a fim de saber a localização dos blocos de dados na máquina do cluster, e aciona os **Tasks Trackers**. Cada um deles executa seu trabalho e retorna para o **Job Tracker**;
- Os **Task Trakers** se comunicam com os **DataNodes** para obter os dados do disco, executar o processamento e retornar o resultado ao **Job Tracker**.
- Essa arquitetura permite armazenar e processar grande quantidade de dados e assim extrair valor do BigData através da análise de dados;

Soluções de Armazenamento e Processamento Paralelo

- **Apache Hadoop:** montar do zero;
- **Cloudera:** suíte de produtos baseados no Hadoop, e a Cloudera implementa o cluster e configura o ambiente;
- **Opções na nuvem:**
- **Microsoft Azura DHInsight:** serviço para utiliza o apache hadoop, dentre outros da mesma família e só paga pelas horas de uso;
 - **Amazon EMR (Elastic Elastic MapReduce)** (o maior provedor em nuvem do mercado);
- **DataBricks:** permite a criação de uma plataforma de **Lakehouse** (mesma equipe que desenvolveu o Apache Spark);
- **Vantagens:** a empresa não tem que se preocupar em comprar máquinas, instalar computadores, instalação elétrica, refrigeração, segurança, troca de equipamentos, etc.

5. CLOUD COMPUTING

- **Cloud Computing (Computação em Nuvem)** é a entrega de serviços de computação - incluindo servidores, armazenamento, banco de dados, red, software, análise e inteligência - pela internet ("a nuvem") para oferecer recursos flexíveis, inovação e economia de escala;
- **Cloud Computing e Big Data:** ao invés de gerir localmente, gere na nuvem;

PRINCIPAIS PROVEDORES EM NUVEM:

- **AWS (Amazon Web Services)**
 - Nasceu há mais de 10 anos com o conceito de ociosidade em servidores;
 - Hoje em dia oferecem mais de 1000 serviços em nuvem;
- **Microsoft Azure**

- 2º maior do mercado;
- Forte integração com uma grande parte de serviços da Microsoft;
- **Google Cloud Platform (GCP)**
 - Surgiu no mercado um pouco mais tarde;
- **IBM Cloud**
- **Oracle Cloud**

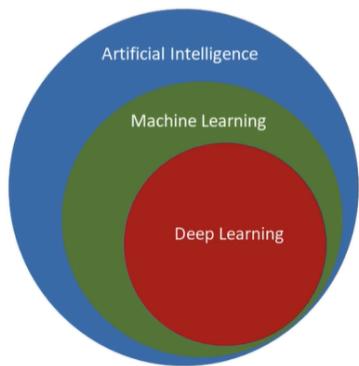
CONHECENDO O AWS (AMAZON WEB SERVICES) - demonstração:

- Serviços > Console de gerenciamento de serviços > Todos os serviços > Computação > EC2
- Dashboard do EC2
 - Cada uma das máquinas tem o hadoop instalado, que é um framework para armazenamento e processamento de BigData;
 - Cria as máquinas, configura o sistema operacional, depois atualiza o S.O., e instala o apache hadoop, faz as configurações necessárias e sobe um cluster (inicializa o cluster hadoop);
 - Ao criar um cluster implementa o conceito de DataLake (repositório que permite armazenar dados em qualquer formato), os dados estarão distribuídos através da máquina do cluster;
 - Se der problema em uma das máquinas, basta ir em “Estado da instância” e encerrar a instância;
 - Criar um grande ambiente de computação para armazenamento e processamento distribuído através de várias máquinas de baixo custo: o **Hadoop** é open source e faz a gestão dessas máquinas;
 - MultiNode Cluster: mais de uma máquina no cluster;
 - **Demonstração...**

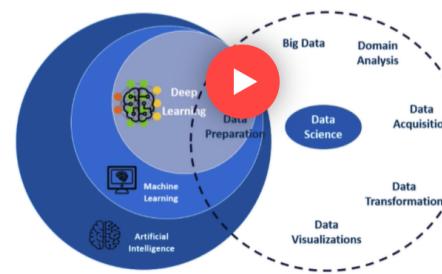
6. MLOps (OPERAÇÕES DE MACHINE LEARNING) e DataOps (OPERAÇÕES COM DADOS)

MACHINE LEARNING

O Que é Machine Learning?



O Que é Machine Learning?

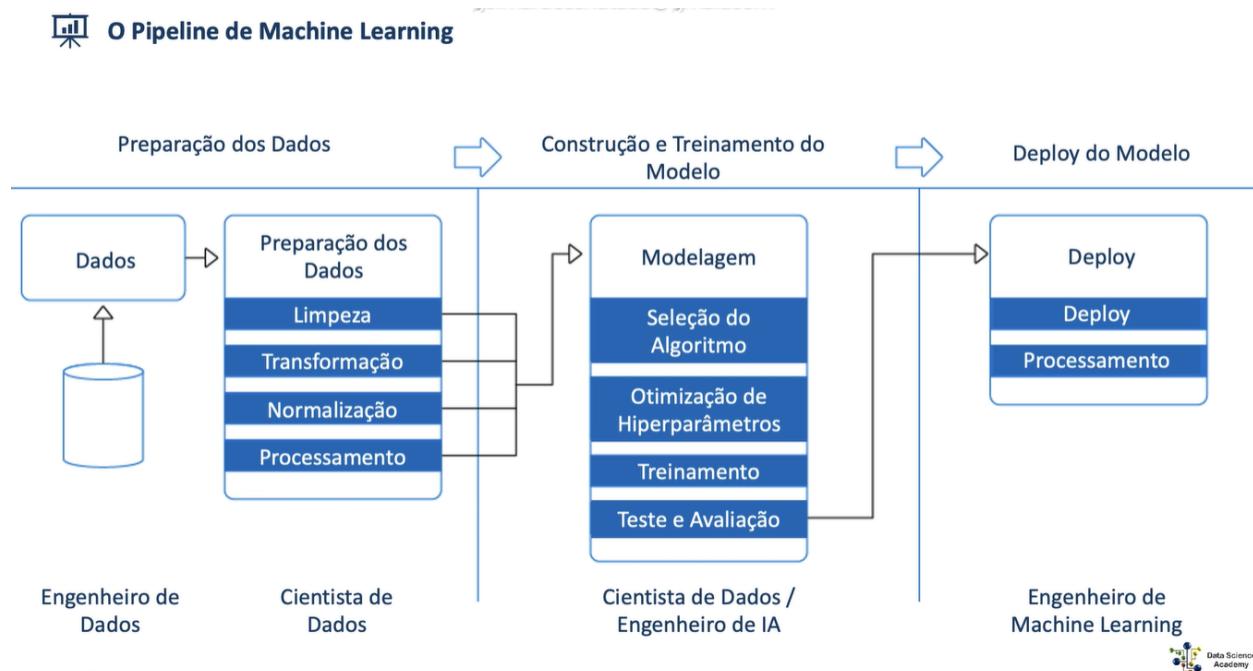


- É uma subárea da Inteligência Artificial (IA) e da Ciência da Computação que se concentra no uso de **dados** e **algoritmos** para imitar a forma como os humanos aprendem, melhorando gradativamente a sua precisão;
- Aplica-se um algoritmo de aprendizado de máquina, utilizando linguagem de programação, prepara-se este algoritmo para rodar através dela (ex. python), apresenta dados a ele (preparar, limpar, transformar e etc.), e depois do treinamento dentro do algoritmo é criado um modelo. Esse modelo aprendeu o relacionamento matemático dos dados.

O Que é Machine Learning?



O PIPELINE DE DADOS DE MACHINE LEARNING (FLUXO DE TRABALHO)

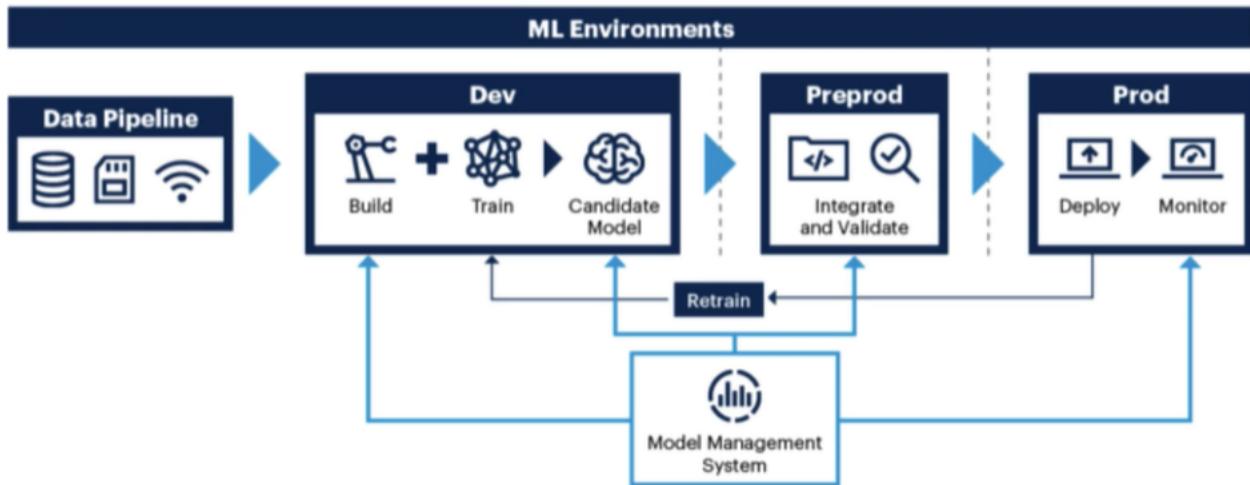


- 1º Definição do problema: não aparece no diagrama, mas é o mais importante;
- Não existe um modelo de machine learning que resolva todos os problemas, cada modelo é específico para um problema;

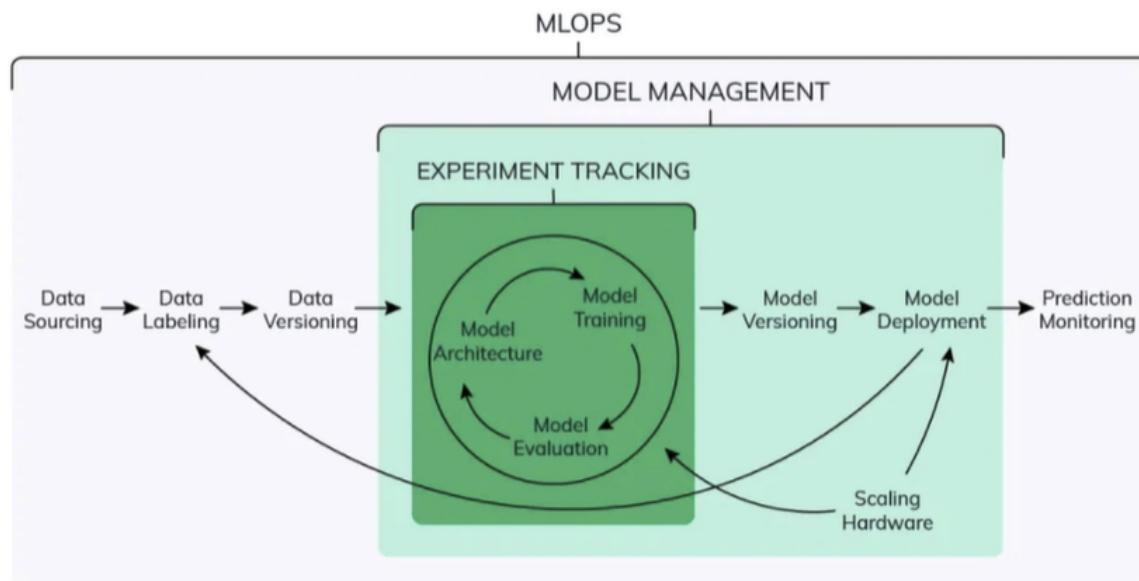
MACHINE LEARNING OPS (MLOps)

- Formalmente, é um conjunto de práticas para colaboração e comunicação entre Cientistas de Dados e profissionais de operações;
- MLOps é, normalmente, tarefa do Engenheiro de Machine Learning;
- A aplicação dessas práticas aumenta a qualidade, simplifica o processo de gerenciamento e automatiza a implantação de modelos de aprendizado de máquina em ambientes de produção em grande escala. É mais fácil alinhar os modelos às necessidades de negócios, bem como aos requisitos regulamentares. Dados não podem ser utilizados de qualquer forma.
- Fluxo de operação em aprendizado de máquina:

Pipeline de Dados



- Sempre treinar o modelo para garantir que a performance seja mantida;
- Treina vários tipos de modelo até ver qual a melhor performance;
- Ciclo de preparação do modelo:



- Cada vez que uma empresa desenvolve software, ela segue esse mesmo fluxo: identificar o problema, desenvolver o software, criar a primeira, segunda, terceira versão, cada uma delas tem que ser gerenciada e entregue ao usuário final, se

tiver alguma correção, volta, prepara uma nova versão (release do software). Machine Learning, em sua essência, é desenvolvimento.

MLOps = ML + DEV + OPS



- MLOps visa unificar o desenvolvimento de sistemas de ML (dev) e a implantação de sistemas ML (ops) para padronizar e agilizar a entrega contínua de modelos de alto desempenho em produção.

QUAIS PROBLEMAS PODEM SER RESOLVIDOS COM MLOPS?

- Há uma **escassez de Cientistas de Dados** que sejam bons em análise de dados e que tenham conhecimento em desenvolvimento e implantação de aplicações web. Atualmente, o perfil de Engenheiro de Machine Learning visa atender a essa necessidade. É um ponto ideal na interseção da Data Science e do DevOps;
- Mudança dos objetivos de negócios no modelo -existem muitas dependências com os dados mudando continuamente, sendo difícil manter os padrões de desempenho do modelo e garantir governança de IA. É difícil acompanhar o treinamento contínuo do modelo e os objetivos de negócios em constante evolução;
- Lacunas de comunicação entre as equipes técnicas e de negócios que não possuem uma linguagem comum. Na maioria das vezes, essa lacuna se torna o motivo do fracasso de grandes projetos;
- Avaliação de risco - há muito debate em torno da natureza da caixa preta de sistemas de Machine Learning. Frequentemente, os modelos tendem a se distanciar do que foram inicialmente planejados. Avaliar o risco/custo de tais falhas é uma etapa muito importante e meticulosa;

- Com MLOps as empresas podem resolver, ou pelo menos amenizar, os problemas acima citados, ao mesmo tempo que exploram todas as vantagens do uso de Big Data em sistemas de Machine Learning.
- O Engenheiro de Machine Learning é o profissional que se concentra em pesquisar, construir e projetar sistemas de inteligência artificial (IA) autoexecutáveis para automatizar modelos preditivos. Os Engenheiros de Machine Learning são os profissionais responsáveis pela publicação (deploy) de modelos de Machine Learning e normalmente os responsáveis por implementar MLOps. Ou seja, dominar soluções de Cloud Computing e ferramentas de automação.
 - Projetar, desenvolver e pesquisar sistemas, modelos e esquemas de Machine Learning(ML)
 - Estudar, transformar e converter protótipos de ciência de dados.
 - Pesquisa e seleção de conjuntos de dados apropriados
 - Execução de análises estatísticas e uso de resultados para melhorar os modelos
 - Treinamento e reciclagem de sistemas e modelos de ML, conforme necessário
 - Identificar diferenças na distribuição de dados que podem afetar o desempenho do modelo em situações do mundo real
 - Visualização de dados para insights mais profundos
 - Analisar os casos de uso de algoritmos de ML e classificá-los por sua probabilidade de sucesso
 - Compreender quando suas descobertas podem ser aplicadas a decisões de negócios
 - Verificar a qualidade dos dados por meio da limpeza de dados.

DevOps, MLOps, AIOps, DataOps

- **DEVOps**

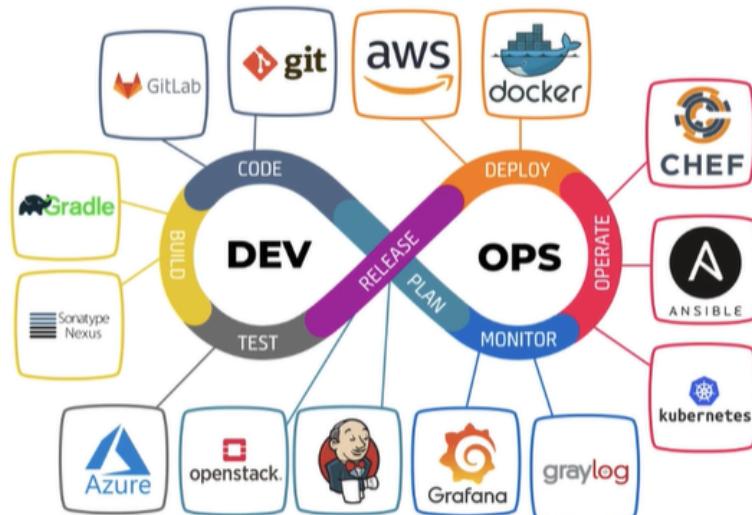
É uma abordagem para desenvolvimento de software que acelera o ciclo de vida de sua construção usando automação. O DevOps se concentra na implementação contínua do software, aproveitando os recursos de TI sob demanda e automazinando a integração, o teste e a implementação do código.

Assim, reduz o tempo de implementação, lançamento no mercado, minimiza defeitos e diminui o tempo necessário para resolver problemas.

Usando o DevOps empresas conseguiram reduzir o tempo do ciclo de lançamento de meses para segundos. Empresas como Google e Amazon agora lançam software muitas vezes por dia.



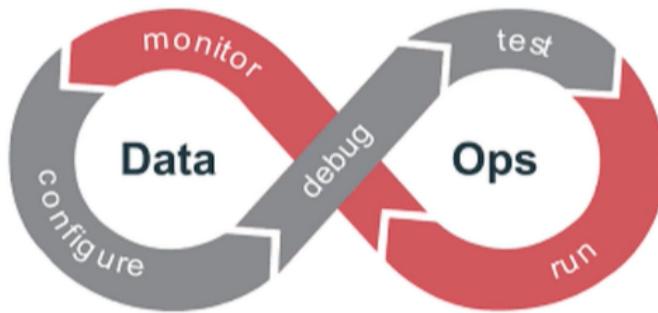
Várias empresas se especializaram em DevOps ao longo do tempo e diversas ferramentas surgiram:



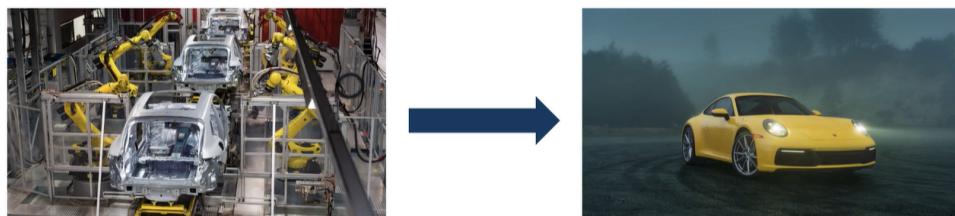
Então foi levado o conceito para a Ciência de Dados, e assim nasceram:

- **MLOps** - operação de fluxo de trabalho em Machine Learning;
- **AIOps** - basicamente uma extensão do MLOps, porém para operações do fluxo de trabalho da IA;

- **DataOps** - conceito mais recente que abrange toda a operação de dados de uma empresa.
- **DataOps**
 - É o amadurecimento da ciência de dados. Pense em DataOps como se fosse uma linha de produção para produzir analytics e a matéria prima são os dados:



- DataOps (Operação de Dados) é uma metodologia ágil e orientada a processos para desenvolver e entregar análises;
- Fornece as ferramentas, processos e estruturas organizacionais para apoiar a empresa focada em dados;
- DataOps é a capacidade de habilitar soluções, desenvolver produtos de dados e ativar dados para valor comercial em todas as camadas da tecnologia, da infraestrutura à experiência do usuário final;
- Dados entram brutos e saem como relatórios, dashboards, análise, etc..
- **Objetivo: agilizar o design, desenvolvimento e manutenção de aplicativos com bases em dados e análise de dados.**
- Busca melhorar a forma como os dados são gerenciados e os produtos são criados e coordenar essas melhorias com o objetivo do negócio;
- As equipes de DataOps também buscam orquestrar dados, ferramentas, código e ambientes do início ao fim, com o objetivo de fornecer resultados reproduzíveis. Essas equipes tendem a ver os pipelines como análogos a linha de produção de uma fábrica, sendo aqui o BigData a matéria prima;



Operações de Dados



Produto Final



Operações de Dados



Produto Final

- O Engenheiro DataOps são profissionais técnicos que se concentram principalmente ou exclusivamente no ciclo de vida de desenvolvimento e implantação. A linha comum entre o Engenheiro de Dados e o Engenheiro DataOps é disponibilizar dados para uso por Cientistas de Dados, Analistas de Dados e outros. Mas os Engenheiros DataOps apoiam o ciclo de fornecimento e utilização de dados definindo o processo e as tecnologias que outros usam para originar, transformar e utilizar dados. O Engenheiro DataOps tem um perfil mais amplo que o Engenheiro de Dados.

BIG DATA x SMALL DATA



PRINCIPAIS FERRAMENTAS:

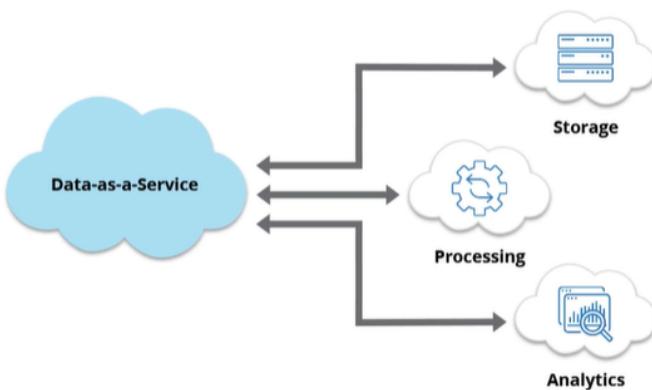
- **MLOps**
 - DVC: DVC, ou Data Version Control, é um sistema de controle de versão de código aberto para projetos de aprendizado de máquina. É uma ferramenta de experimentação que ajuda a definir o pipeline, independentemente da linguagem usada.
 - Pachyderm: plataforma que combina linhagem de dados com pipelines de ponta a ponta. Disponível em código aberto ,plataforma completa com versão controlada ou Hub Edition (combina características das duas versões anteriores).
 - Airflow: plataforma de código aberto que permite monitorar, agendar e gerenciar osfluxos de trabalho usando appweb. Ele fornece uma visão sobre o status das tarefas concluídas e em andamento, juntamente com uma visão dos logs.
 - Neptune: Repositório de metadados desenvolvido para equipes de pesquisa e produção que realizam muitos experimentos.
 - MLflow: plataforma de código aberto que ajuda a gerenciar todo o ciclo de vida do aprendizado de máquina que inclui experimentação, reproduzibilidade, implantação e um registro de modelo central. É adequado para indivíduos e equipes de qualquer tamanho. A ferramenta é

independente de biblioteca. Você pode usá-lo com qualquer biblioteca de aprendizado de máquina e em qualquer linguagem de programação.

- **DataOps**

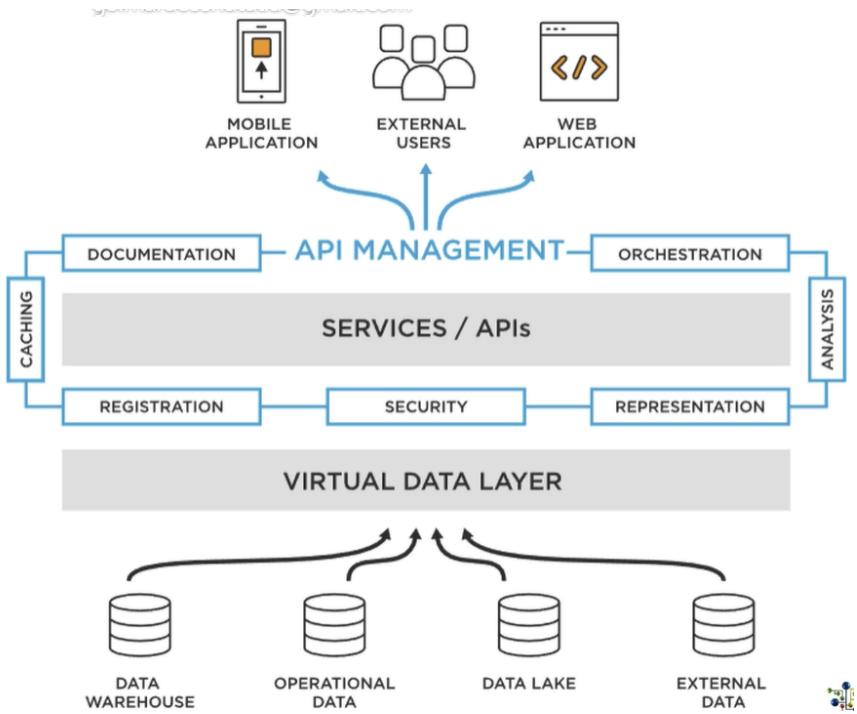
- DataKitchen: Uma das ferramentas DataOps mais populares, é a melhor para automatizar e coordenar pessoas, ambientes e ferramentas em análise de dados de toda a organização. O DataKitchen cuida de tudo -do teste à orquestração, ao desenvolvimento e à implantação. Permite que as organizações criem ambientes de trabalho em questão de minutos para que as equipes possam experimentar sem interromper os ciclos de produção. O pipeline de qualidade do DataKitchen é baseado em três seções principais; dados, produção e valor. É essencial entender que, com esta ferramenta, você pode acessar o pipeline com o código Python, transformá-lo via LinguagemSQL, projetar o modelo em R, visualizar na pasta de trabalho e obter relatórios noTableau.
- Genie: Desenvolvida pela Netflix, essa ferramenta DataOps é um mecanismo de código aberto que oferece serviços de orquestração de trabalhos distribuídos. Essa ferramenta fornece APIs para desenvolvedores que desejam executar uma ampla variedade de trabalhos com Big Data, usandoHive, Hadoop, Presto e Spark. Genie também fornece APIs para gerenciamento de metadados em clusters de processamento distribuído.
- Piper: Piper é um pacote de ferramentas de DataOps baseadas em aprendizado de máquina que permite que as organizações leiam dados de maneira mais suave e eficiente. Esta solução expõe os dados por meio de um conjunto de APIs que se integram facilmente aos ativos digitais da organização.
- Airflow: Airflow é uma plataforma de DataOps(e também MLOps) de código aberto que gerencia fluxos de trabalho complexos em qualquer organização, considerando os processos de dados como DAG (Directed Acyclic Graphs). Projetado pelo Airbnb para agendar e monitorar seus fluxos de trabalho, agora as empresas podem utilizar essa ferramenta de código aberto para gerenciar seu processo de dados no MacOS, Linux e Windows.

7. DADOS COMO SERVIÇOS (DaaS)



- Ideia de fornecer os dados corretos no momento em que a empresa precisa para então executar a análise e entregar o resultado tomador de decisão;
- É uma estratégia de gerenciamento de dados que visa alavancar os dados como um ativo de negócios para maior agilidade no processo de análise;
- Faz parte das ofertas “as a service”, ex. planilha e editor de texto do Google;
- Maneira de gerenciar grandes quantidades de dados que as organizações geram todos os dias e fornecer essas informações valiosas em toda a empresa para a tomada de decisões baseada em dados;

ARQUITETURA DaaS



- VIRTUAL DATA LAYER: canal de saída que centraliza os dados
- Nos SERVICES/APIs define-se como os dados vão retornar para o usuário final, que faz uma chamada ao API;
- Já pode inclusive entregar a análise pronta;
- **A arquitetura DaaS se concentra no provisionamento de dados de uma variedade de fontes sob demanda por meio do uso de APIs;**
- Projetado para simplificar, oferece conjunto de dados já tratados ou fluxo de dados para serem consumidos em uma variedade de formatos, geralmente unificados usando virtualização de dados;
- Pode incluir também uma variedade de tecnologias de gerenciamento de dados, serviços de dados, análise de autoatendimento (Self-Service Analytics) e catalogação dos dados

PRINCIPAIS BENEFÍCIOS

- Monetização de dados;
- Redução de custos;

- Caminho mais rápido para inovação (realizar previsões, detectar padrões, etc.). A inovação não é mais um diferencial, é um requisito básico hoje em dia;
- Agilidade no processo de decisão baseado em dados;
- Menor risco no uso de dados;
- Criação de uma Cultura Data-Driven (todas as decisões são orientadas a dados).

ARQUITETURAS MODERNAS DE BIG DATAS

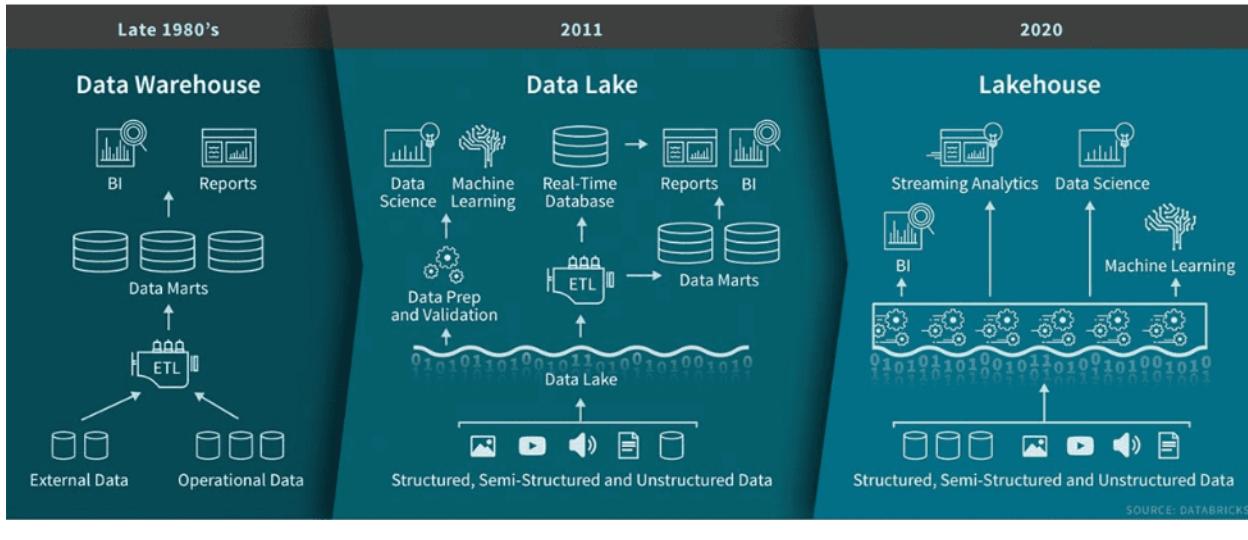
- Objetivo de tornar o acesso aos dados mais fácil, eficiente, seguro e amplamente disponível;
- As duas últimas tendências em arquiteturas de plataforma de dados emergentes são o Data Lakehouse o Data Mesh;

DATA LAKEHOUSE

Um Data Lakehouse é uma **nova arquitetura de gerenciamento de dados que combina a flexibilidade, economia e escala de Data Lakes com o gerenciamento de dados e transações ACID (Atomicidade, Consistência, Isolamento, Durabilidade) de Data Warehouses**, permitindo Business Intelligence (BI) e Machine Learning(ML) em todos dados armazenados em um único repositório.

São habilitados por um novo design de sistema aberto: implementação de estruturas de dados e recursos de gerenciamento de dados semelhantes aos de um Data Warehouse, diretamente no tipo de armazenamento de baixo custo usado para Data Lakes. Mesclá-los em um único sistema significa que as equipes de dados podem se mover mais rapidamente, pois podem usar os dados sem a necessidade de acessar vários sistemas.

Os Data Lakehouses também garantem que as equipes tenham os dados mais completos e atualizados disponíveis para projetos de ciência de dados, aprendizado de máquina e análise de negócios.O diagrama abaixo ilustra a ideia por trás do Data Lakehouse.



DATA LAKE

DATA MESH

É um tipo de arquitetura de plataforma de dados que abrange a onipresença dos dados na empresa, permitindo um design orientado ao domínio e de autoatendimento. Data Mesh é **amplamente considerado a próxima grande mudança arquitetônica em dados**. É uma nova abordagem para projetar e desenvolver arquiteturas de dados. Ao contrário de uma arquitetura centralizada e monolítica baseada em um Data Warehouse e/ou um Data Lake, Data Mesh é uma **arquitetura de dados altamente descentralizada**.

Tenta resolver três desafios quando temos um Data Lake/Warehouse centralizado:

- Falta de propriedade. Quem é o proprietário dos dados - a equipe da fonte de dados ou a equipe de infraestrutura?
- Falta de qualidade. A equipe de infraestrutura é responsável pela qualidade, mas não conhece bem os dados.
- Escalonamento organizacional. O armazenamento central torna-se o gargalo, como no caso de um Data Lake/Warehouse empresarial.

O objetivo com Data Mesh é **tratar os dados como um produto**, com cada fonte tendo seu próprio gerente/proprietário de produto de dados (que fazem parte de uma equipe multifuncional de Engenheiros de Dados) e sendo seu próprio domínio claramente focado e com uma oferta autônoma, tornando-se os blocos de construção

fundamentais de uma malha(Mesh), levando a uma arquitetura distribuída orientada por domínio.

Observe que, por motivos de desempenho, você pode ter um domínio que agrupa dados de várias fontes. Cada domínio deve ser detectável, endereçável, autoexplicativo, seguro (governado por controle de acesso global), confiável e interoperável (governado por um padrão aberto). Cada domínio armazenará seus dados em um Data Lake e, em muitos casos, também terá uma cópia de alguns dos dados em um banco de dados relacional.

Outro componente do Data Mesh é a infraestrutura de dados como plataforma, que **fornecer armazenamento, pipeline, catálogo de dados e controle de acesso aos domínios**. A ideia principal é **evitar a duplicação de esforços**. Isso permitirá que cada equipe de produto de dados crie seus produtos de dados rapidamente.

Vale ressaltar que Data Mesh ainda é uma tendência e sua implementação tem diversos desafios técnicos.

DATA MESH COMO PARADIGMA DE ARQUITETURA DE DADOS

Data Mesh é um paradigma arquitetônico e organizacional que **desafia a antiga suposição de que devemos centralizar os dados para usá-los**, ter todos os dados em um só lugar ou ter os dados gerenciados por uma equipe de dados centralizada para agregar valor. Para o Big Data fomentar a inovação, sua propriedade deve ser federada entre os proprietários de dados que são responsáveis por fornecer seus dados como produtos (com o suporte de uma plataforma de dados de autoatendimento para abstrair a complexidade técnica envolvida em servir produtos de dados).

Também devemos adotar uma nova forma de governança federada por meio da automação para permitir a interoperabilidade de produtos de dados orientados a domínio. **A descentralização, junto com a interoperabilidade e o foco na experiência dos consumidores de dados, são fundamentais para a democratização da inovação usando dados.**

Se uma organização tem muitos domínios com vários sistemas e equipes gerando dados ou um conjunto diversificado de casos de uso e padrões de acesso orientados a dados, o uso de Data Mesh pode ser uma opção viável.

A implementação de Data Mesh requer investimento na construção de uma plataforma de dados de autoatendimento e adoção de uma mudança organizacional para domínios

a fim assumir a propriedade de longo prazo dos produtos de dados, bem como uma estrutura de incentivos que recompensa domínios que servem e utilizam dados como um produto. Data Mesh marca uma mudança bem-vinda de paradigma arquitetônico e organizacional em como gerenciamos Big Data. O paradigma é baseado em quatro princípios:

- (1) **Descentralização** orientada ao domínio da propriedade e arquitetura de dados;
- (2) Dados orientados ao domínio servidos como um **produto**;
- (3) Infraestrutura de dados de **autoatendimento** como uma plataforma para habilitar equipes de dados autônomas e orientadas para o domínio;
- (4) Governança federada para **permitir ecossistemas e interoperabilidade**.

Há ainda uma grande lacuna de ferramentas comerciais para acelerar a implementação de Data Mesh e hoje o que vemos é a implementação de blocos isolados que então podem ser conectados criando assim uma grande malha de dados e tornando esse tipo de arquitetura uma realidade.

Teremos ainda muita evolução e novas ferramentas surgindo nos próximos anos.

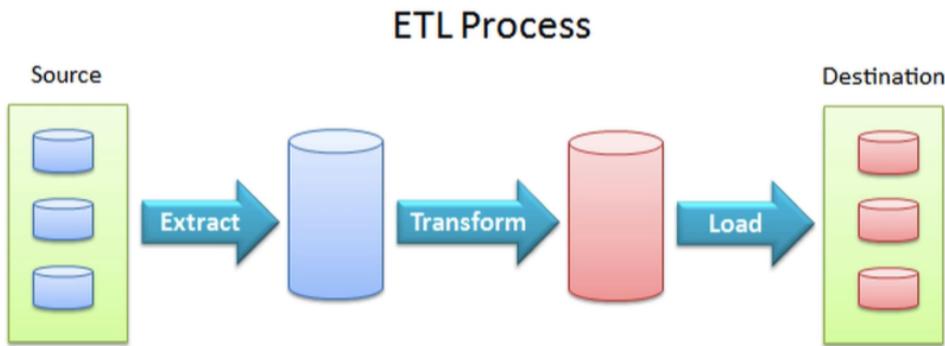
SOLUÇÕES COMERCIAIS

- Para o **Data Lakehouse**, a solução mais comum é o **Databricks**. A DSA foi pioneira no Brasil ao trazer projetos de Databricks;
- **Data Mesh** é um conceito de arquitetura e sua implementação pode envolver diversas tecnologias. Considerando o ambiente de Cloud Computing do Microsoft Azure, o **Azure Purview** seria seu ponto de partida para descobrir dados.

Se você precisar fazer consultas entre domínios, também chamadas de consultas federadas, use o **Synapse** sem servidor com o Azure Virtual Network Peering se estiver consultando dados de contas de armazenamento (vinculando as contas de armazenamento em cada espaço de trabalho do Synapse). Se consultar dados de pools dedicados relacionais do Synapse, isso exigiria atualmente trabalho extra, como usar notebooks Synapse Spark, Databricks, Power BI ou fluxos de dados do

Azure Data Factory para chamar vários bancos de dados hospedados em pools dedicados separados (mas há soluções mais fáceis).

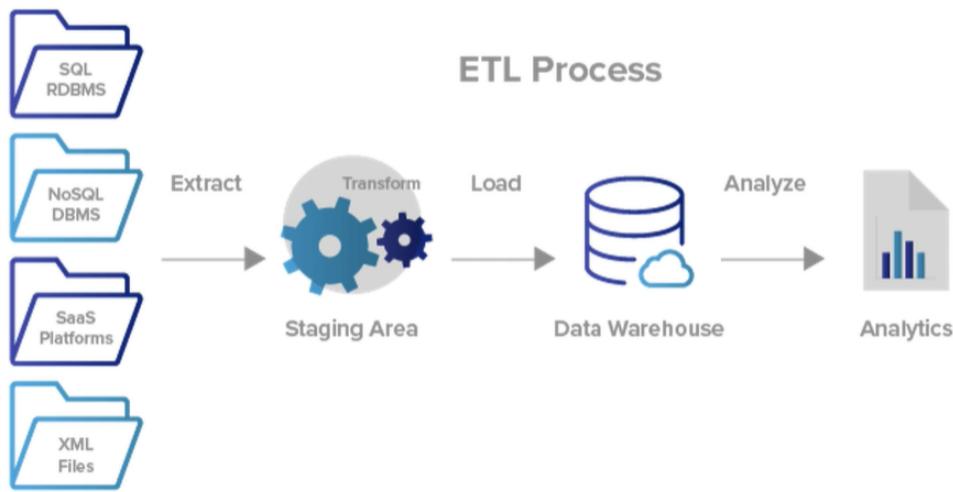
8. ETL - Extração, Transformação e Carga de Dados



ETL = Processo de movimentação dos dados

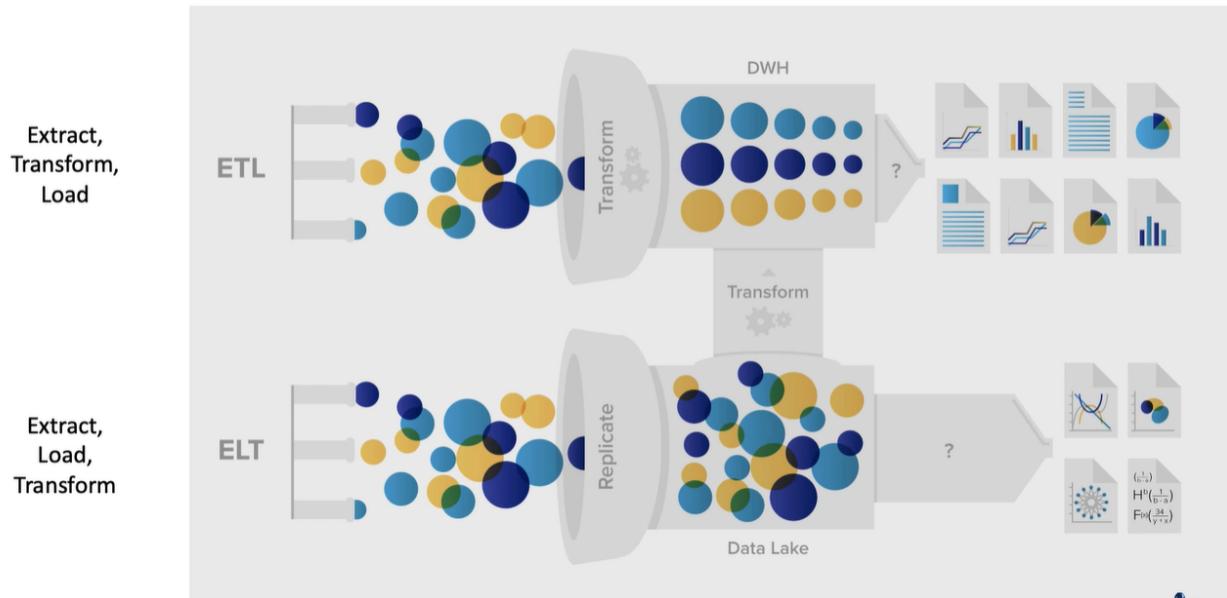
Tem-se uma **Source (fonte)** de onde são **extraídos** os dados para uma **área intermediária (staging area)** e **transformam-se** os dados, dependendo da necessidade da empresa em termos de **limpeza e processamento** de dados. Dentro da staging area podem ser criados procedimentos de validação dos dados (ex. padronização)

Após a aplicação da **transformação**, carregam-se os dados no **destino (por ex. Data Warehouse - DW)**. Depois, estão prontos para **análise (analytics)**.



ETL x ELT (a evolução do ETL)

- ELT = EXTRACT, LOAD AND TRANSFORM



Ou seja, não transforma os dados antes de carregar, garantindo a coleta e armazenamento e mais tarde aplicando a transformação de acordo com a necessidade do processo de análise, pois talvez não precisa-se usar todos os dados simultaneamente. O processo acaba sendo um pouco mais ágil.

Algumas empresas implementam uma arquitetura híbrida.

PRINCIPAIS SOLUÇÕES DE ETL E ELT NO MERCADO

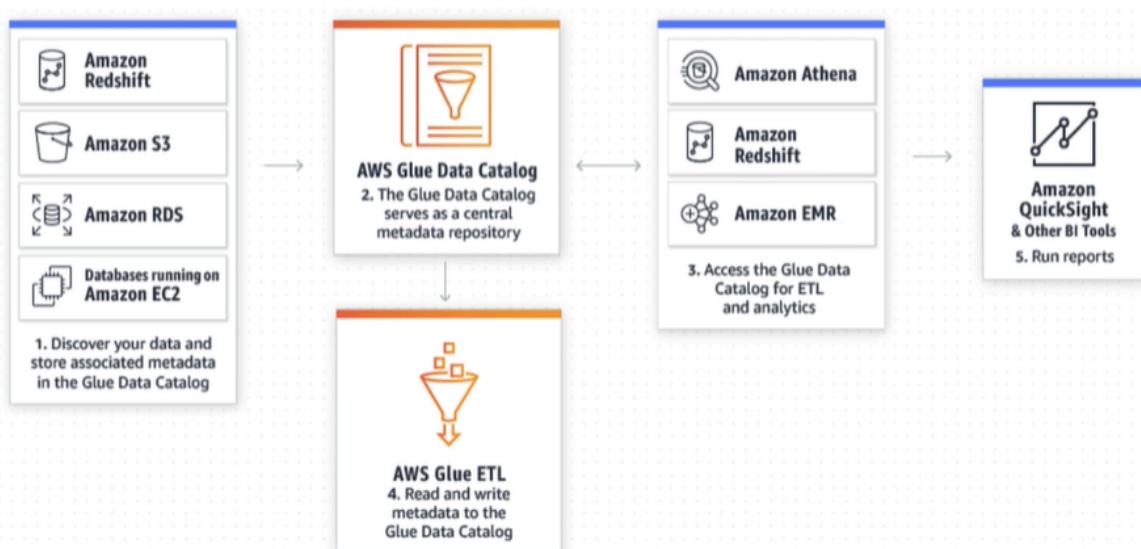
- **Oracle Data Integrator (ODI)**: muito conhecido. Não é gratuito.
- **Pentaho Data Integator (PDI)**: comprado pela HITACHI. Não é gratuito, porém tem uma versão community para teste.
- **Apache NIFI**: open source. Ótima ferramenta gratuita. Interface gráfica amigável.
- **Apache Spark**: framework de processamento de dados. Dá para aplicar a linguagem SQL. Não tem uma interface gráfica como o NIFI, é tudo feito via programação.

Na nuvem:

- **Azure Data Factory**: da Microsoft, ou seja, integra com produtos da Microsoft. Low code, ou seja, não é necessário programação para executar.
- **AWS Glue**: da Amazon. Principal conceito ser *serverless* (sem servidor - ou seja, não requer a instalação e configuração. Você recebe uma interface, faz escolhas com mouse e seu processo ETL/ELT está pronto). Não é gratuito.
- **Amazon Athena**: também da Amazon. Não é uma ferramenta ETL, é um motor de execução de linguagem SQL. Não é necessário instalar, também é *serverless*. Trabalha quase em conjunto com o AWS Glue, se complementam.
- **PowerCenter**: da empresa Informática. Tem bastante ferramentas ligadas a IA.
- **Apache Airflow**: tem a proposta específica de criar, agendar e monitorar fluxos de trabalho, mas pode ser usado dentro de soluções ETL e ELT. É open source/gratuito. Requer programação em python.

DEMO ELT E BIG DATA

- Feita com AWS Glue: integração de dados sem servidor (*serverless*)



9. COMO INICIAR UM PROJETO BIG DATA?

CASOS DE USO DE BIG DATA ANALYTICS

- Ceasar Entertainment (3 m. registros/hora);
- Cerner (monitoramento + 1m. pacientes diários);
- eHarmony (milhões pessoas diárias)
- [...];

COMO INICIAR UM PROJETO DE BIG DATA?

Normalmente de 7 a 9 meses se a empresa tiver uma estrutura bem definida e trabalhar de forma ágil...

- **Definição do *Business Case***
 - Documento que define claramente o objetivo do projeto, a direção do negócio, obstáculos, interessados e papéis, caso de uso mais importante...

- Problemas e obstáculos em termos não técnicos;
 - Definição de métricas;
 - Um projeto desse normalmente deve vir da área de negócios e não da área de tecnologia, pois um BigData é implementado para resolver problemas de negócio;
- **Planejamento do Projeto**
 - Especificar metas esperadas em termos comerciais e mensuráveis;
 - Identificar as questões comerciais com a maior precisão possível;
 - Determinar quais são os outros requisitos de negócio (quantificando);
 - Definir como seria uma implementação bem sucedida;
 - Documentar os critérios de sucesso do projeto;
 - Certificar que cada objetivo comercial tenha um critério mensurável;
 - Compartilhar e obter aprovação dos critérios das partes interessadas (*stakeholders*);
 - Determinar o escopo adequado;
 - Desenvolver orçamento;
 - Definir uma linha do tempo com marcos de sucesso entre 3 meses, 6 meses e 1 ano;
 - **Definição dos requisitos técnicos (criação da arquitetura do projeto):**
 - Definição dos atributos necessários em bancos de dados (quais bancos relacionais serão utilizados da estrutura atual da empresa);
 - Quais serão as possíveis fontes de dados;
 - Definição de como será feita a mescla dos dados coletados (ex. comentários, cliques, acessos, logs, dados CRM...);
 - Quais ferramentas de análise serão utilizadas? Open source, pou solução proprietária (paga)?
 - Quais são as habilidades técnicas necessárias para se trabalhar com todo esse ambiente?

- Vai utilizar *Data Lakes*? Os *Data Lakes* serão em nuvem ou *on premissse* (instados localmente)?
 - Qual será a arquitetura do seu projeto?
 - Você vai montar um *Enterprise DataHub* (conjunto de infraestutura para *BigData*)?
 - Quais serão as ferramentas de relatórios e visualização? *Open Source* (R e Python)? ...
- **Criação de um “*Total Business Value Assessment*”**
 - *Time To Business*, ou seja, quanto tempo realmente terei para que esse projeto comece a gerar resultado? (Normalmente trabalha-se com período de 3 anos após o tempo de implementação);
 - Definição da facilidade de uso;
 - Escalabilidade: precisa poder crescer de forma rápida;
 - Definir os critérios de padrões em termos de estabelecimento de *BigData*;
 - Maturidade da empresa: a empresa já tem uma estrutura analítica orientada para dados?
 - Definir suporte e manutenção após a implementação do projeto;