

FIAP GRADUAÇÃO

# DATA SCIENCE

## BIG DATA ARCHITECTURING & DATA INTEGRATION

Prof. Dr. Renê de Ávila Mendes



# CHECKPOINT 4



## Checkpoint 4

---

- **Data:** 14/08 (entrega até 20/08)
- **Local:** remoto

# Objetivos da disciplina

**DISCIPLINA:** Big Data Architecturing & Data Integration

**OBJETIVOS:** Entenda as principais **arquiteturas** para ingestão, processamento e análise de grandes volumes de dados. Conheça as principais **ferramentas** open-source de Big Data como Hadoop, MapReduce, Spark, Sqoop, NiFi, Flume, Kafka, Zookeeper, HBase, Hive e as **integre** com as ferramentas de **extração, transformação e carga** de dados em modelos dimensionais. Entenda conceitos sobre **computação paralela e distribuída**, aplicação do Hadoop e bases Apache e arquiteturas *serverless* e desacopladas. Veja como **visualizar os dados** estruturados ou não estruturados com ferramentas de Self-Service Business Intelligence como PowerBI, utilizando as melhores práticas de **visualização de dados**.

## **Assuntos – 2º Semestre**

- Arquiteturas para Big Data
- Data Pipelines
- Conceito de Data Lake
- PIG, HIVE, Spark, Data Streaming

# Metodologia e Ferramenta

- Material oficial do Knime > Certificação (!)
- Conteúdo da disciplina

**14/08 – 19:30H**

- Entrevista com **Aline Bessa**
- Brasileira, Cientista de Dados na Knime (EUA)
- Professora e Pesquisadora na New York University (2014 – 2021)





## Objetivo dessa aula

**OBJETIVOS:** Entender o conceito de pipeline; Data Lake; relembrar processamento em lote, em fluxo; conceituar arquitetura e framework; entender os requisitos de arquiteturas para integração de dados Big Data.

# Alguns conceitos iniciais



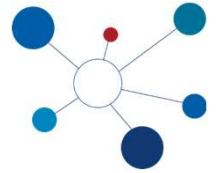
## Sistema

Pode ser definido como aquilo que é feito por pessoas e que possa ser configurado com hardware, software, dados, humanos, processos, procedimentos, facilidades, materiais e outras entidades que ocorram naturalmente ou como uma combinação de **elementos que interagem para alcançar um objetivo**, cujo entendimento pode ser tornado mais claro pela utilização de um substantivo associativo, a exemplo de "sistema de vôo", que pode ser substituído simplesmente por seu sinônimo, tal como avião, helicóptero ou outro sistema capaz de voar.

ISO. *ISO/IEC 12207:2008 Systems and software engineering—software life cycle processes*. [S.l.]: International Organization for Standardization, 2008.

IEEE. *IEEE 15288-2004 Adoption of ISO/IEC 15288:2002 Systems Engineering—System Life Cycle Processes*. [S.l.]: IEEE Computer Society, 2004.

# Alguns conceitos iniciais



## Arquitetura

Pode ser definida como aqueles **conceitos ou** aquelas **propriedades fundamentais de um sistema** incorporados em seus elementos, relacionamentos ou ainda em seus princípios de projeto e evolução. A definição utiliza a disjunção “ou” para abarcar as filosofias de arquitetura como conceito de um sistema na mente de uma pessoa e de arquitetura como percepção das propriedades de um sistema. Neste sentido uma arquitetura é abstrata, requerendo artefatos que a descrevam e documentem, o que é então definido como Descrição da Arquitetura.

ISO. *ISO/IEC 42010:2011 Systems and software engineering — Architecture description*. [S.l.]: International Organization for Standardization, 2011.

# Alguns conceitos iniciais



## Framework

No contexto de Tecnologia da Informação, pode ser definido como **classes ou quadros cooperativos que tornam um projeto reutilizável** para uma classe específica de software.

IEEE. *IEEE 1517-2010 IEEE Standard for Information Technology - System and Software Life Cycle Processes - Reuse Processes*. [S.l.]: IEEE Computer Society, 2010.

# O pipeline de dados analíticos



Os dados não se formatam por si mesmos para fins de análise. Eles passam por uma série de passos que envolvem a coleta a partir das origens, o tratamento para as formas necessárias à análise e finalmente sua disponibilização na forma de resultados para serem consumidos. Estes passos podem ser pensados como um “tubo”. Este modelo ajuda a entender onde aplicar cada tecnologia.

TEJADA, Zoiner. **Mastering Azure Analytics: Architecting in the Cloud with Azure Data Lake, HDInsight, and Spark.** O'Reilly Media, Inc., 2017.

# O pipeline de dados analíticos



**A localização a partir  
da qual novos dados  
brutos são obtidos**

TEJADA, Zoiner. **Mastering Azure Analytics: Architecting in the Cloud with Azure Data Lake, HDInsight, and Spark.** O'Reilly Media, Inc., 2017.

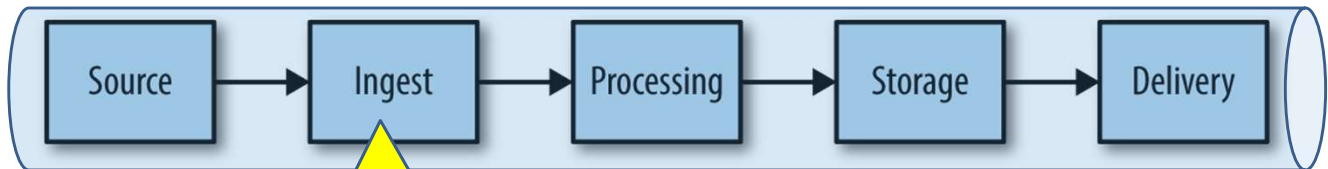
# O pipeline de dados analíticos



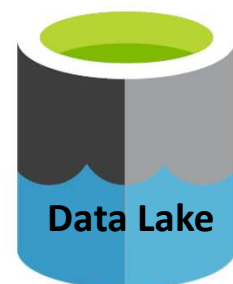
**Os processos computacionais responsáveis pelo recebimento dos dados brutos da origem para que possam ser processados.**

TEJADA, Zoiner. **Mastering Azure Analytics: Architecting in the Cloud with Azure Data Lake, HDInsight, and Spark.** O'Reilly Media, Inc., 2017.

# O pipeline de dados analíticos



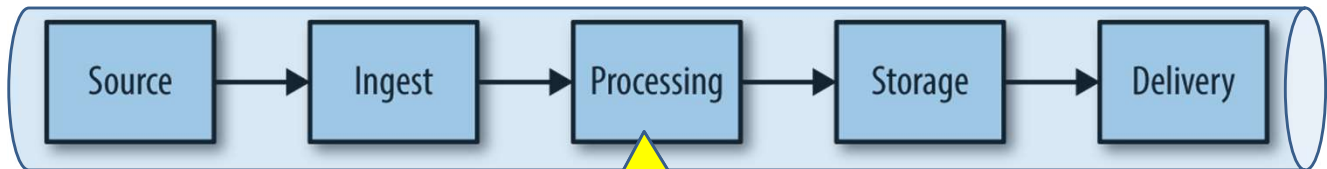
**Os processos computacionais responsáveis pelo recebimento dos dados brutos da origem para que possam ser processados.**



A fase de ingestão deste modelo relaciona-se ao conceito Data Lake, abordado em seguida.



# O pipeline de dados analíticos



**Processos computacionais que controlam como o dado é preparado e processado para ser entregue.**

TEJADA, Zoiner. **Mastering Azure Analytics: Architecting in the Cloud with Azure Data Lake, HDInsight, and Spark.** O'Reilly Media, Inc., 2017.

# O pipeline de dados analíticos



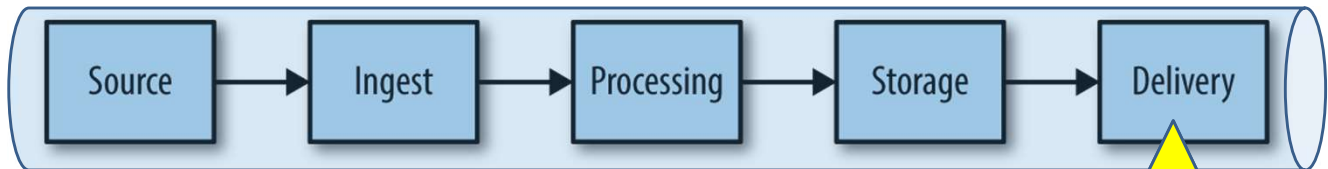
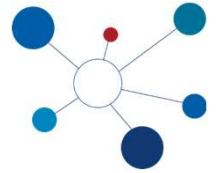
**TRANSIENTE X PERSISTENTE**

**Os vários locais onde os dados ingeridos, os dados intermediários e os resultados finais são armazenados.**

O armazenamento pode ser transiente (o dado reside na memória somente por um período finito de tempo) ou persistente (o dado é armazenado definitivamente).

TEJADA, Zoiner. **Mastering Azure Analytics: Architecting in the Cloud with Azure Data Lake, HDInsight, and Spark.** O'Reilly Media, Inc., 2017.

# O pipeline de dados analíticos

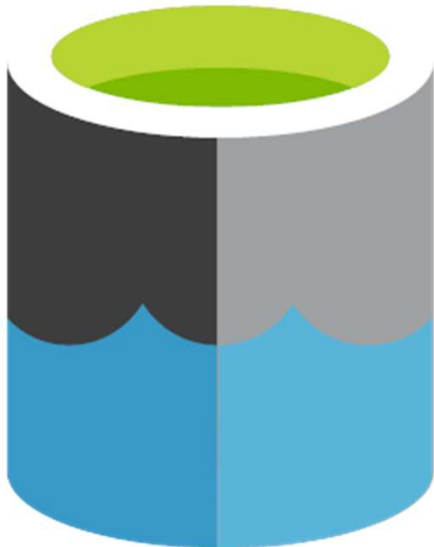


**Como o dado é entregue ao consumidor, seja para análise ou para ser consumido por outros processos.**

TEJADA, Zoiner. **Mastering Azure Analytics: Architecting in the Cloud with Azure Data Lake, HDInsight, and Spark.** O'Reilly Media, Inc., 2017.

## Data Lake - Conceituação

Ingest



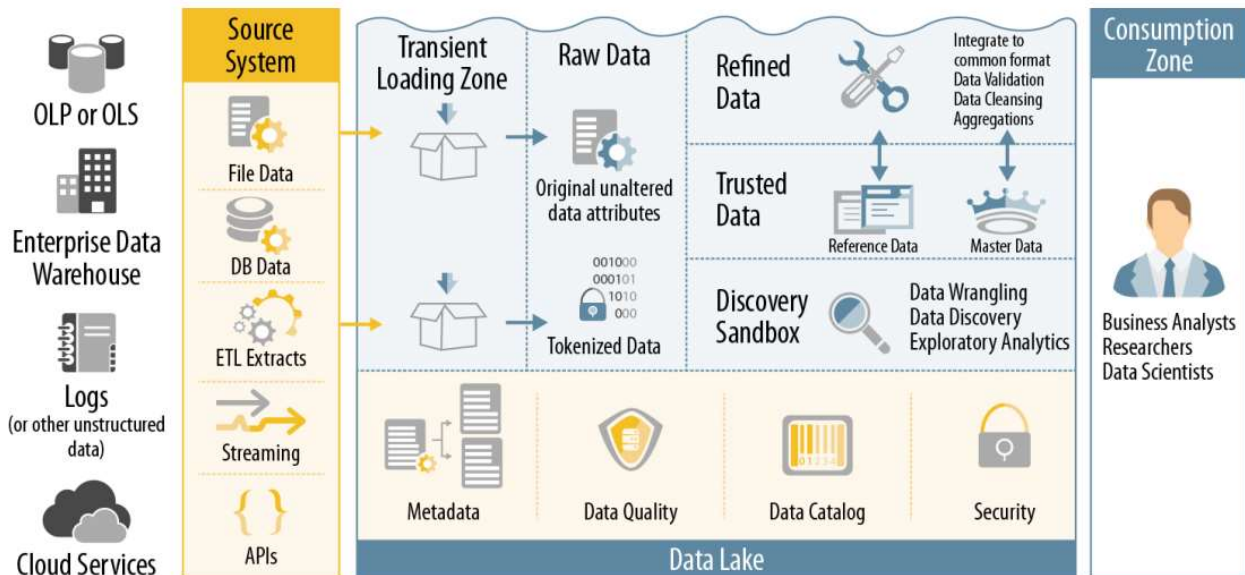
Repositório **centralizado** para armazenamento de todos os dados de uma corporação, independentemente da origem e do formato. Recebe dados estruturados, semiestruturados e desestruturados.

Como todos os dados são bem-vindos, os data lakes são uma abordagem emergente e poderosa para os desafios da integração de dados em um EDW (Enterprise Data Warehouse) tradicional, especialmente quando as organizações se voltam para aplicativos móveis e baseados em nuvem e IoT.

LAPLANTE, Alice e SHARMA, Ben. **Architecting Data Lakes: Data Management Architectures for Advanced Business Use Cases**. O'Reilly Media, Inc., 2016.

# Data Lake - Arquitetura

Ingest



LAPLANTE, Alice e SHARMA, Ben. **Architecting Data Lakes: Data Management Architectures for Advanced Business Use Cases**. O'Reilly Media, Inc., 2016, p14.

## Data Lake - Benefícios

Ingest

Pode receber dados de qualquer fonte de dados.

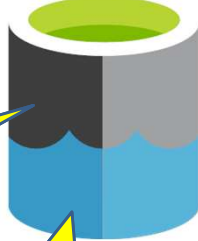


Você pode armazenar todos os tipos de dados estruturados e não estruturados em um data lake, desde dados de CRM até postagens de mídia social.

LAPLANTE, Alice e SHARMA, Ben. **Architecting Data Lakes: Data Management Architectures for Advanced Business Use Cases**. O'Reilly Media, Inc., 2016.

## Data Lake - Benefícios

Ingest



Pode receber dados de qualquer fonte de dados.

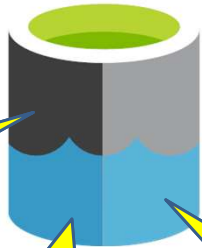
Não requer conhecimento prévio das perguntas a serem respondidas pelo modelo.

Simplesmente armazene dados brutos: você pode refiná-los à medida que sua compreensão e percepção melhoram.

LAPLANTE, Alice e SHARMA, Ben. **Architecting Data Lakes: Data Management Architectures for Advanced Business Use Cases**. O'Reilly Media, Inc., 2016.

## Data Lake - Benefícios

Ingest



Pode receber dados de qualquer fonte de dados.

Não requer conhecimento prévio das perguntas a serem respondidas pelo modelo.

Permite o uso de ferramentas diferentes para processamento do dado.

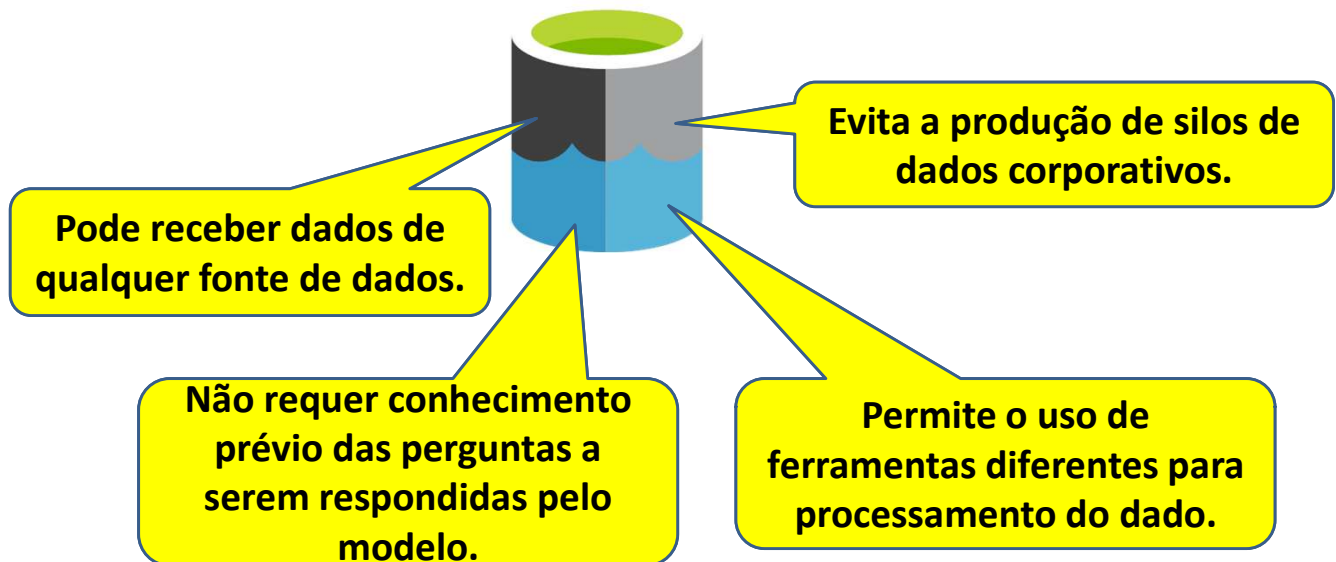
Você pode usar uma variedade de ferramentas para obter informações sobre o que os dados significam.

LAPLANTE, Alice e SHARMA, Ben. **Architecting Data Lakes: Data Management Architectures for Advanced Business Use Cases**. O'Reilly Media, Inc., 2016.



## Data Lake - Benefícios

Ingest



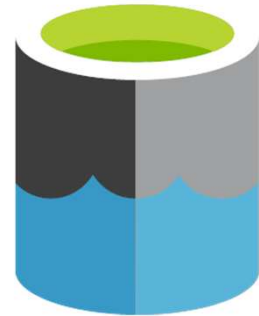
Você obtém um acesso democratizado com uma visão única e unificada dos dados em toda a organização.

LAPLANTE, Alice e SHARMA, Ben. **Architecting Data Lakes: Data Management Architectures for Advanced Business Use Cases**. O'Reilly Media, Inc., 2016.

## Data Lake – Características essenciais

Ingest

Deve ser um repositório único de dados, normalmente armazenado em HDFS (*Hadoop Distributed File System*).



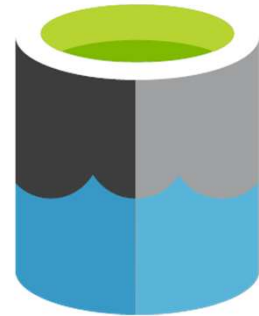
Ao armazenar os dados em HDFS o formato original do dado é preservado e as mudanças do ciclo de vida do dado são capturadas. Esta característica é especialmente útil para atender as necessidades de compliance e de auditoria.

LAPLANTE, Alice e SHARMA, Ben. **Architecting Data Lakes: Data Management Architectures for Advanced Business Use Cases**. O'Reilly Media, Inc., 2016.

## Data Lake – Características essenciais

Ingest

Deve ter recursos de orquestração e agendamento de tarefas. Pode utilizar o Apache YARN, que faz parte do framework Hadoop.



A execução da carga de trabalho é um pré-requisito para o Enterprise Hadoop, e o YARN fornece gerenciamento de recursos e uma plataforma central para fornecer operações consistentes, segurança e ferramentas de governança de dados em clusters Hadoop, garantindo que os fluxos de trabalho analíticos tenham acesso aos dados e ao poder de computação de que precisam.

LAPLANTE, Alice e SHARMA, Ben. **Architecting Data Lakes: Data Management Architectures for Advanced Business Use Cases**. O'Reilly Media, Inc., 2016.

## Data Lake – Características essenciais

Ingest

Deve conter recursos (aplicações e *workflows*) para consumir, processar e agir sobre os dados.

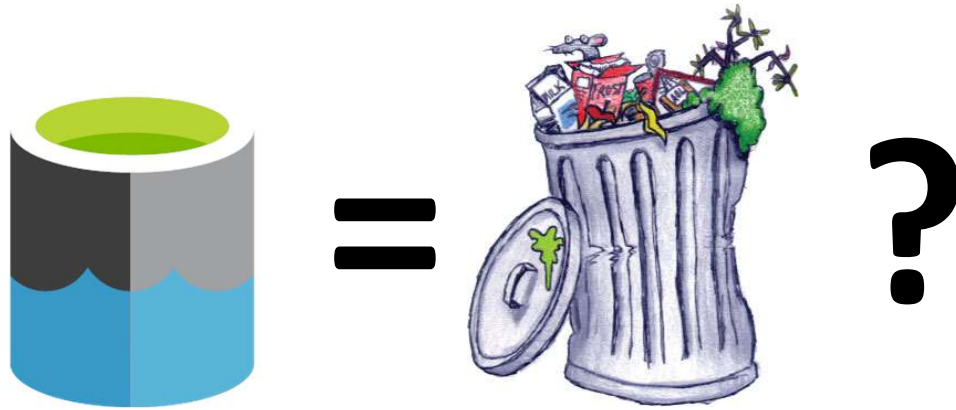


O fácil acesso do usuário é uma das marcas de um data lake, devido ao fato de que as organizações preservam os dados em sua forma original. Sejam estruturados, não estruturados ou semiestruturados, os dados são carregados e armazenados como estão. Os proprietários de dados podem facilmente consolidar dados de clientes, fornecedores e operações, eliminando obstáculos técnicos — e até mesmo políticos — ao compartilhamento de dados.

LAPLANTE, Alice e SHARMA, Ben. **Architecting Data Lakes: Data Management Architectures for Advanced Business Use Cases**. O'Reilly Media, Inc., 2016.

## Data Lake – Uso

Ingest



LAPLANTE, Alice e SHARMA, Ben. **Architecting Data Lakes: Data Management Architectures for Advanced Business Use Cases**. O'Reilly Media, Inc., 2016.

## Data Lake – Uso

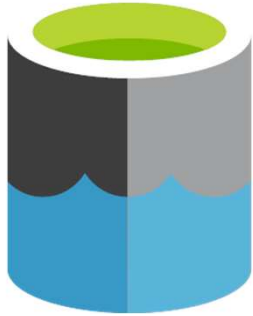
Ingest



LAPLANTE, Alice e SHARMA, Ben. **Architecting Data Lakes: Data Management Architectures for Advanced Business Use Cases**. O'Reilly Media, Inc., 2016.

## Data Lake – Abordagens de GD

Ingest



### Resolver depois

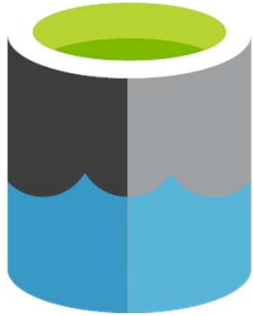
Carregar os dados livremente e aplicar processos de limpeza no momento da análise.

**Risco:** o que restar do processo de limpeza permanece no Data Lake e passa a ser ignorado por sua baixa qualidade.

LAPLANTE, Alice e SHARMA, Ben. **Architecting Data Lakes: Data Management Architectures for Advanced Business Use Cases**. O'Reilly Media, Inc., 2016.

## Data Lake – Abordagens de GD

Ingest



### Adaptar o ETL

Adaptar as ferramentas e processos atualmente utilizados no ETL para carregar dados no Data Lake.

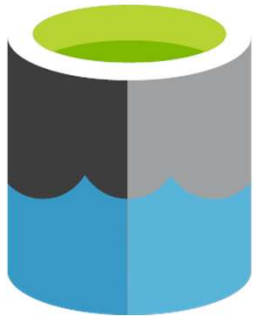
**Desvantagem:** o ETL passa a ser executado fora do Data Lake, diminuindo o desempenho e aumentando custos.

LAPLANTE, Alice e SHARMA, Ben. **Architecting Data Lakes: Data Management Architectures for Advanced Business Use Cases**. O'Reilly Media, Inc., 2016.



## Data Lake – Abordagens de GD

Ingest



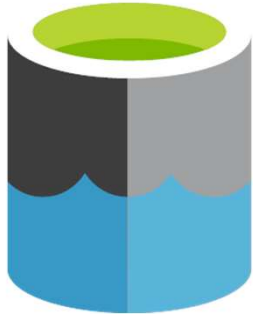
### Scripts customizados

Desenvolver um fluxo utilizando scripts customizados para conectar processos, aplicações, verificações de qualidade e transformações dos dados. **Desvantagem:** maior risco de erros e necessidade de mão-de-obra altamente qualificada. Alto custo.

LAPLANTE, Alice e SHARMA, Ben. **Architecting Data Lakes: Data Management Architectures for Advanced Business Use Cases**. O'Reilly Media, Inc., 2016.

## Data Lake – Abordagens de GD

Ingest

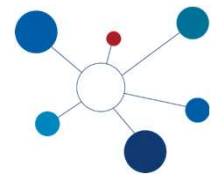


### Plataforma de Gerenciamento

Utilizar ferramentas apropriadas para ingerir grandes volumes de dados. Estas soluções fazem a catalogação dos dados e controlam o fluxo de dados mantendo a qualidade.

LAPLANTE, Alice e SHARMA, Ben. **Architecting Data Lakes: Data Management Architectures for Advanced Business Use Cases**. O'Reilly Media, Inc., 2016.

## Tipos de processamento



**Lote (*batch*)**

**Fluxo (*stream*)**

Uma observação final importante: o conceito de data lake como é usado hoje é destinado ao processamento em lote, onde a alta latência (tempo até os resultados ficarem prontos) é apropriada. Dito isso, o suporte para processamento de latência mais baixa é uma área natural de evolução para data lakes, portanto, essa definição pode evoluir com o cenário tecnológico.

TEJADA, Zoiner. **Mastering Azure Analytics: Architecting in the Cloud with Azure Data Lake, HDInsight, and Spark.** O'Reilly Media, Inc., 2017.

# Tipos de processamento

- Processamento em lote



# Tipos de processamento

- Processamento em fluxo

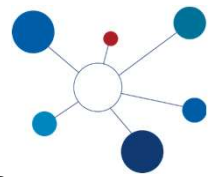




## Tipos de processamento

- Processamento em lote
  - Computa o **volume completo de dados** gerando visões a partir dele
  - Os volumes de dados são **grandes**
  - As **restrições de tempo** de resposta para a análise são menos exigentes
- Processamento em fluxo
  - Computa apenas o **volume de dados mais recente**
  - Requerido por aplicações de análise em **tempo real, ou quase real**
  - A variável tempo de resposta, ou latência, torna-se um **fator crítico** para o sucesso

## Big Data – Por que uma arquitetura ?

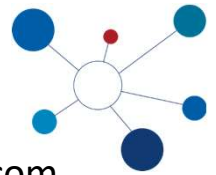


- Necessidade de inserir e pesquisar grandes volumes de dados com uma latência aceitável;

A escolha de arquitetura para sistemas Big Data surge da necessidade de inserir e pesquisar dados em grandes volumes e com uma latência aceitável para o contexto do sistema. Mesmo o escalamento dos tradicionais bancos de dados e até mesmo a inclusão de recursos de processamento assíncrono não são capazes de responder satisfatoriamente a problemas tais como corrupção de dados, tolerância a falhas e alta disponibilidade.

(MARZ; WARREN, 2015).

## Big Data – Por que uma arquitetura ?



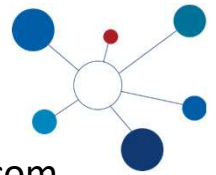
- Necessidade de inserir e pesquisar grandes volumes de dados com uma latência aceitável;
- O escalamento dos tradicionais bancos de dados e mesmo a inclusão de recursos de processamento assíncrono não respondem satisfatoriamente a problemas de corrupção de dados, tolerância a falhas e alta disponibilidade;

A escolha de arquitetura para sistemas Big Data surge da necessidade de inserir e pesquisar dados em grandes volumes e com uma latência aceitável para o contexto do sistema. Mesmo o escalamento dos tradicionais bancos de dados e até mesmo a inclusão de recursos de processamento assíncrono não são capazes de responder satisfatoriamente a problemas tais como corrupção de dados, tolerância a falhas e alta disponibilidade.

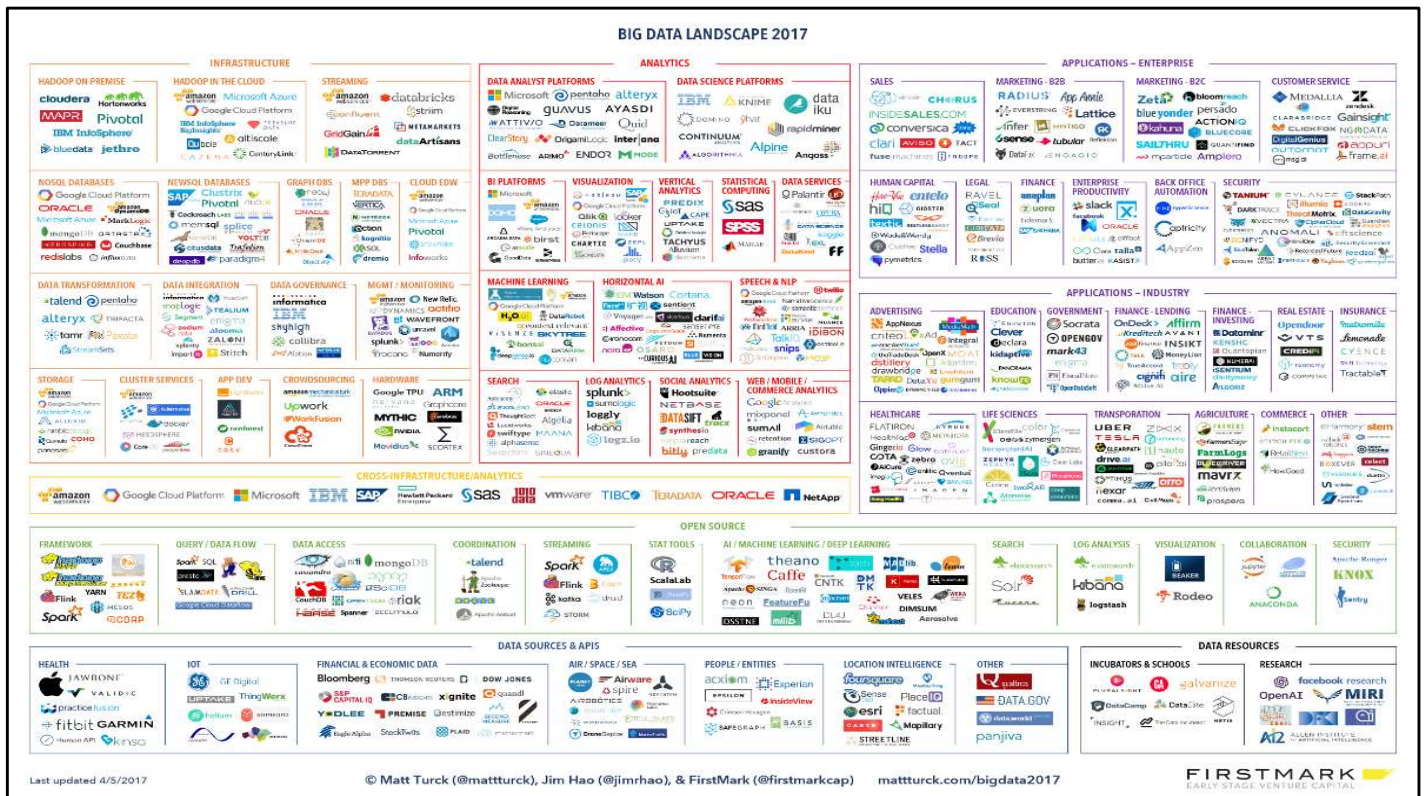
(MARZ; WARREN, 2015).



## Big Data – Por que uma arquitetura ?

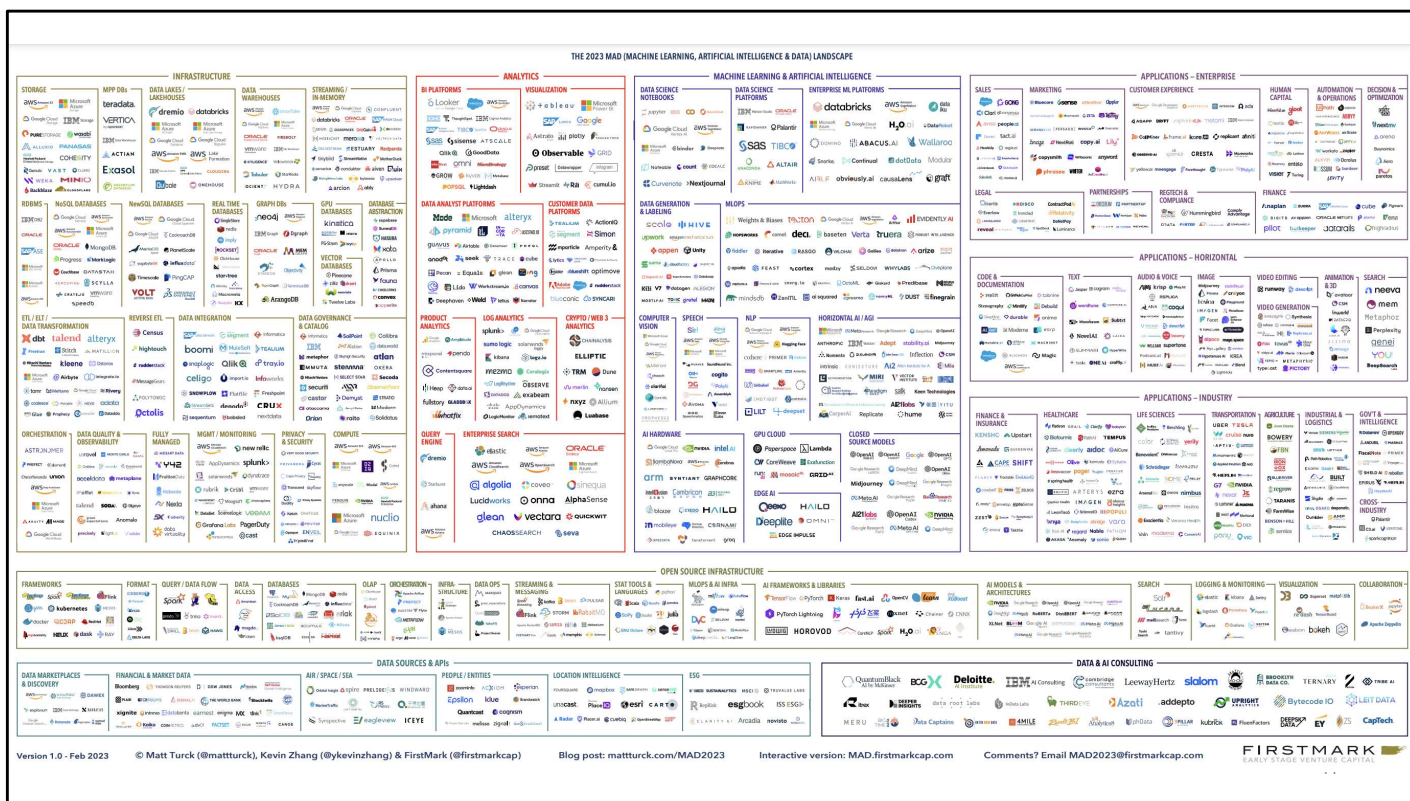


- Necessidade de inserir e pesquisar grandes volumes de dados com uma latência aceitável;
- O escalamento dos tradicionais bancos de dados e mesmo a inclusão de recursos de processamento assíncrono não respondem satisfatoriamente a problemas de corrupção de dados, tolerância a falhas e alta disponibilidade;
- E ...



Um importante aspecto que diferencia o fenômeno Big Data do fenômeno conhecido como Business Intelligence é a oferta de ferramentas Open Source, muitas delas desenvolvidas e disponibilizadas por grandes empresas geradoras de dados, tais como Facebook, Yahoo!, Twitter e LinkedIn, tornando a implementação de soluções e as iniciativas analíticas em Big Data acessíveis (FAN; BIFET, 2013).





Fonte: <https://venturebeat.com/wp-content/uploads/2023/02/Matt-Turck-MAD2023.png?resize=2763%2C1424&strip=all>



# Big Data – Requisitos sistêmicos

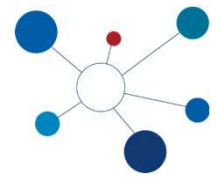


## 1. Robustez e tolerância a falhas

Em um sistema distribuído situações como a queda de um nó, a complexidade de um banco de dados distribuído, concorrência, duplicidade de dados ou mesmo erros humanos são passíveis de ocorrer e devem ser evitadas sempre que possível. O sistema deve ser capaz de se recuperar de situações como estas.

MARZ, N.; WARREN, J. *Big Data: Principles and best practices of scalable realtime data systems*. [S.l.]: Manning Publications Co., 2015.

# Big Data – Requisitos sistêmicos

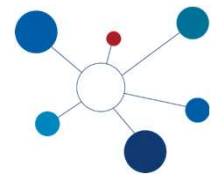


## 2. Baixa latência de leitura e atualização

A latência de leitura de dados em geral está na casa dos milissegundos e a latência de atualização é dependente do tipo de sistema. Qualquer que seja a latência esperada, a solução não deve comprometer a robustez.

MARZ, N.; WARREN, J. *Big Data: Principles and best practices of scalable realtime data systems*. [S.l.]: Manning Publications Co., 2015.

## Big Data – Requisitos sistêmicos



### 3. Escalabilidade

O sistema de dados deve ser capaz de manter o desempenho mesmo em situações de aumento da carga de trabalho ou da quantidade de dados, o que é solucionável pela adição de máquinas (escalabilidade horizontal).

MARZ, N.; WARREN, J. *Big Data: Principles and best practices of scalable realtime data systems*. [S.l.]: Manning Publications Co., 2015.

## Big Data – Requisitos sistêmicos



### 4. Generalização

O sistema de dados deve suportar diferentes tipos de aplicações.

### 5. Extensibilidade

O sistema deve poder ser modificado ou acrescido sem grandes impactos de desenvolvimento.

MARZ, N.; WARREN, J. *Big Data: Principles and best practices of scalable realtime data systems*. [S.l.]: Manning Publications Co., 2015.



## Big Data – Requisitos sistêmicos

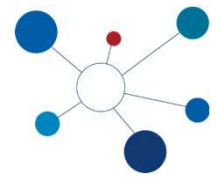


### 6. Suporte a consultas *ad hoc*

O sistema terá valor à medida em que permitir pesquisas arbitrárias em seus dados visando à obtenção de valor.

MARZ, N.; WARREN, J. *Big Data: Principles and best practices of scalable realtime data systems*. [S.l.]: Manning Publications Co., 2015.

# Big Data – Requisitos sistêmicos



## 7. Mínima manutenção

Manter um sistema em funcionamento com uma taxa mínima de manutenção, o que pode ser obtido pela escolha de soluções com a menor complexidade possível. Quanto maior a complexidade de um componente, maior a chance de requerer manutenção.

MARZ, N.; WARREN, J. *Big Data: Principles and best practices of scalable realtime data systems*. [S.l.]: Manning Publications Co., 2015.

## Big Data – Requisitos sistêmicos



### 8. Capacidade de correção (*debuggability*)

O sistema deve permitir a habilitação de traces para a identificação de erros.

MARZ, N.; WARREN, J. *Big Data: Principles and best practices of scalable realtime data systems*. [S.l.]: Manning Publications Co., 2015.

## Big Data – Principais arquiteturas

- Kappa
- Lambda
- Liquid

