

MBA⁺

**Data Science &
Artificial Intelligence**



**MBA⁺****Data Architecture,
Integration and Ingestion**

Prof.: Ivan Gancev

Email: profivan.gancev@fiap.com.br



Data Architecture, Integration and Ingestion

(O que vamos explorar?)

Aula 1 – 12/abr (qua)

- Pilares de arquitetura: persistência, integração e consumo
- Estratégias de arquitetura
- Tipos de tratamentos e arquiteturas

Aula 2 – 19/abr (qua)

- Exemplos de Bancos, diferenças e usos:
 - Bancos Relacionais
 - Bancos Colunares

Aula 3 – 26/abr (qua)

- Exemplos de Bancos, diferenças e usos:
 - Bancos de documentos
 - Bancos chave-valor
 - Bancos de Grafos

Aula 4 – 03/mai (qua)

- Ingestão de dados, tratamentos e manipulações
- Pipeline de dados, governança e qualidade
- Integração de dados
 - Cargas batch, ETL, vantagens e desvantagens

Aula 5 – 10/mai (qua)

- Eventos, APIs, NRT e casos de uso
- Arquiteturas para analytics
- Boas práticas, recomendações e cuidados

Ingestão de dados

Ingestão de dados consiste na transferência de dados de uma ou mais fontes para dentro de um banco de dados.

Além disso, uma ingestão de dados irá tratar aspectos do dado como:

- **Qualidade**
- **Compleitude**
- **Transformações**
- **Organização dentro da plataforma de dados**

Tipos de ingestão de dados

Vamos considerar 2 tipos de ingestão: BATCH e STREAMING

BATCH

São transferências de dados em lotes com 1 ou mais origem, normalmente com grandes volumes de dados e recorrências pré-definidas. Ex:

- Uma carga de vendas diária para os sistemas gerenciais
- Uma atualização dos dados de uma linha de produção a cada turno
- Coleta dos estoques a cada hora
- Atualização minuto a minuto de uma fila de incidentes (*micro-batch*)

45697056

Tipos de ingestão de dados

Vamos considerar 2 tipos de ingestão: BATCH e STREAMING

STREAMING

São transferências normalmente disparadas por um evento na origem e visam transferir a informação em pequenas quantidades, o mais rápido possível. Ex:

- Transferência de arquivos pesados em pequenas porções
- Disparo de um alerta de um sensor de presença em uma casa monitorada
- Inclusão de publicidade em sites personalizadas para cada cliente
- Logs de acesso de um site público

Características

Um fluxo de ingestão de dados tem como desafios:

- ✓ A **latência do dado**, que é o tempo necessário desde a geração do dado até sua disponibilização para um uso de negócio
- ✓ A **qualidade do dado** oferecido para que seja confiável e decisões de negócio possam ser tomadas com base em dados
- ✓ A **disponibilidade do dado** no momento em que é consultado. Uma indisponibilidade pode afetar gravemente o time-to-market do negócio.

Tratamentos de dados

Após a ingestão dos dados, estes dados precisarão ser organizados e tratados para atender ao negócio da empresa. Para que possam ser usados, é importante que estejam catalogados e disponíveis.

A partir de um catálogo de dados, o cientista de dados poderá utilizar estes dados para obtenção de informações e disponibilização de novos dados

Alguns exemplos de ferramentas

The Alteryx logo, featuring the word "alteryx" in a blue, lowercase, sans-serif font.The Pentaho logo, consisting of a blue circular icon with a white spiral inside, followed by the word "pentaho" in a black, lowercase, sans-serif font.The Debezium logo, featuring a green and blue icon with four curved lines forming a square shape, followed by the word "debezium" in a black, lowercase, sans-serif font.The Apache Kafka logo, featuring a black icon with four circles connected by lines, followed by the word "kafka" in a black, lowercase, sans-serif font, with "APACHE" in a smaller font above it.The Talend logo, featuring the word "talend" in a red, lowercase, sans-serif font.The Informatica logo, featuring an orange and blue icon with a white triangle inside, followed by the word "Informatica" in a black, uppercase, sans-serif font.The Apache NiFi logo, featuring the word "nifi" in a blue, lowercase, sans-serif font, with "APACHE" in a smaller font above it, and a blue icon of a water drop with a grid pattern inside it.

Exercício: Ingestão de dados



Capturar um arquivo através do NiFi e inseri-lo no HDFS

Iniciem os dockers do Hadoop

1. Abrir o prompt de comando
2. Abrir o diretório: C:\docker\5dts
 - `C:\`
 - `cd \docker\5dts`
3. Verificar a pasta "bigdata_docker"
 - `dir`
4. Acessar a pasta "bigdata_docker"
 - `cd bigdata_docker`
5. Iniciar os dockers
 - `docker-compose up -d`

Apache nifi



- ✓ Plataforma open-source para construção de fluxos de dados
- ✓ Escalável, seguro e tolerante á falhas
- ✓ Interface web-based-browser e API para desenvolvimento e monitoramento dos fluxos
- ✓ Os fluxos podem ser alterados em tempo real de execução

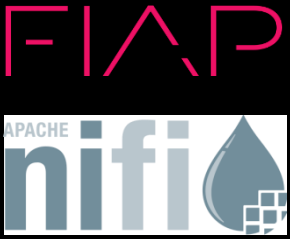
45697056

Funcionalidades do nifi



- ✓ Extração e carga de dados de diversas fontes no modelo “drag and drop”
(programação pode ser necessária)
- ✓ Controle em tempo real para gerenciar o fluxo de dados entre qualquer origem e destino
- ✓ Foi projetado para escalar em Clusters, oferecendo escalabilidade e entrega garantida dos dados
- ✓ Criação de Templates de Fluxos
- ✓ Data Provenance (Data Lineage)
- ✓ Extensibilidade, Escalabilidade e Segurança

Flow File



- ✓ Um FlowFile representa cada objeto que se move pelo mapa de fluxo, para cada um, o NiFi rastreia um mapa de strings de atributos de par chave/valor e seu conteúdo associado de zero ou mais bytes.
- ✓ Basicamente, aqui estamos falando do componente de fluxo de dados em si.
- ✓ Essa é uma funcionalidade Core do NiFi.

Processor



- ✓ Componentes de execução de tarefas individuais (conexão em FTP, Alertas, Conexão em DB)
- ✓ Cada Processor tem uma finalidade de execução, como roteamento (transferência de dados), comunicação entre sistemas. Eles conseguem herdar atributos e conteúdos do Flow File.
- ✓ Existem mais de 200 Processors existentes, cada um para uma finalidade
- ✓ É possível criar seus próprios Processors customizados

45697056

Controller Service



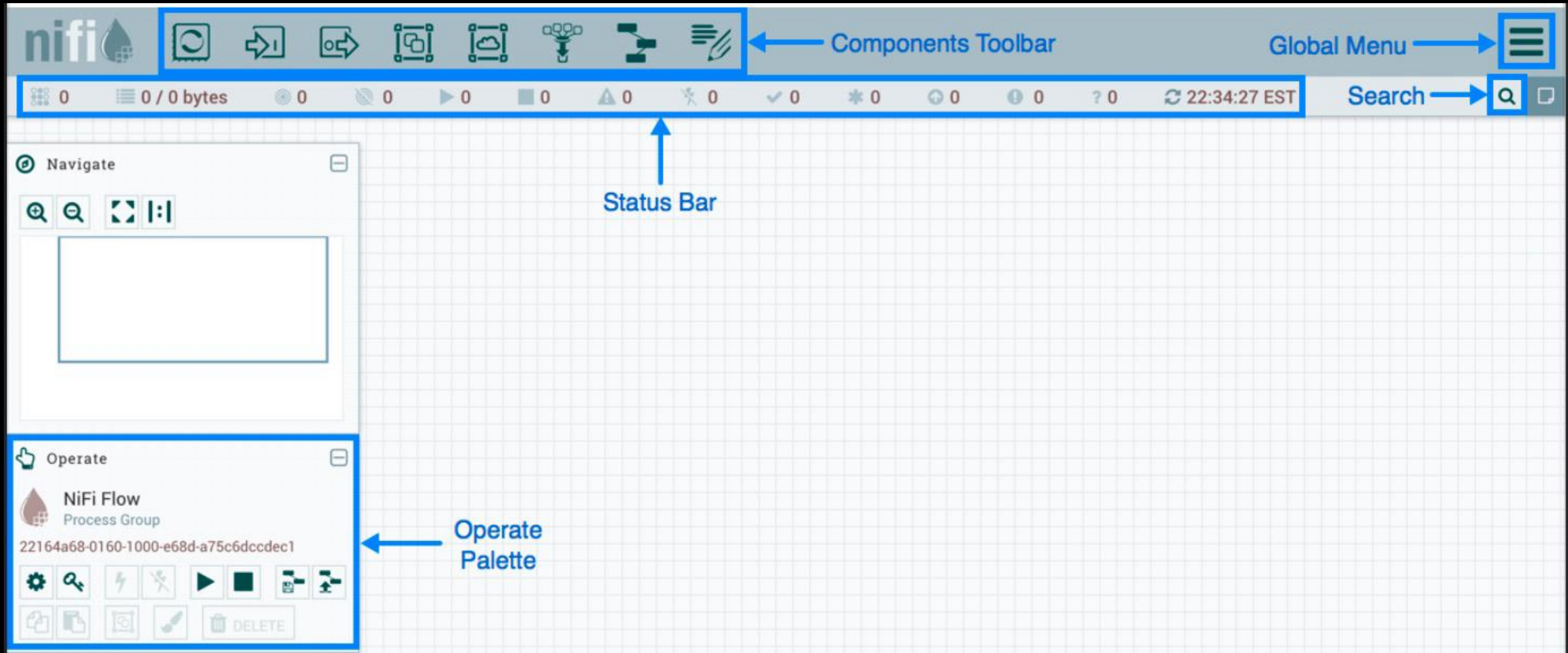
- ✓ Atua como Controlador de Serviços gerenciando os recursos compartilhados
- ✓ Conexão em bancos de dados
- ✓ Gestão do Cache
- ✓ Credenciais Cloud
- ✓ ContextMap (API)

Connections



- ✓ Connections basicamente existem para conectar os Processors entre si
- ✓ Atua como uma fila de execução e podem ser configurado para priorizar dinamicamente a execução da tarefa
- ✓ Possuem limite de carga de dados trafegadas para evitar sobrecarga no Processor

Interface nifi



Exercício nifi

Há um conjunto de imagens docker com componentes Hadoop. Siga os passos iniciais abaixo:

Script disponibilizado com os comandos

1. Copiar o arquivo .csv do exercício para o container nifi
2. Copiar os arquivos hdfs-site.xml e core-site.xml para o container nifi
3. Acessar o nifi pelo browser
4. Adicionar um processor do tipo GetFile
5. Configurar o processor para ler o arquivo .csv (*)
6. Adicionar um processor do tipo PutHDFS
7. Configurar o processor para se conectar ao HDFS
8. Conectar os dois processors e marcar as opções de auto-terminate
9. Executar os 2 processors e desliga-los em seguida (*)
10. Verificar o diretório do HDFS se o arquivo foi carregado

Atenção para o tempo de execução do scheduler.

O motivo de desligar o processor é para não ficar carregando o arquivo repetidamente

Processor

Displaying 284 of 284		Filter
Type ▲	Version	Tags
AttributeRollingWindow	1.11.3	rolling, data science, Attribute ...
AttributesToCSV	1.11.3	flowfile, csv, attributes
AttributesToJSON	1.11.3	flowfile, json, attributes
Base64EncodeContent	1.11.3	encode, base64
CalculateRecordStats	1.11.3	stats, record, metrics
CaptureChangeMySQL	1.11.3	cdc, jdbc, mysql, sql
CompareFuzzyHash	1.11.3	fuzzy-hashing, hashing, cyber-...
CompressContent	1.11.3	lzma, decompress, compress, ...
ConnectWebSocket	1.11.3	subscribe, consume, listen, We...
ConsumeAMQP	1.11.3	receive, amqp, rabbit, get, cons...
ConsumeAzureEventHub	1.11.3	cloud, streaming, streams, eve...
ConsumeFWS	1.11.3	FWS Exchange Email Consu

O nifi possui uma lista de
processors suportados que
interagem com vários
componentes dentro e fora do
ecossistema Hadoop

45697056

Desafio



1. Usar o processor *SplitRecord* para quebrar um flowfile em vários
2. Usar os processors *EvaluateJsonPath* e *UpdateAttribute* para criar atributos com base no conteúdo

Desafio

SplitRecord



SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field			
+			
Property	Value		
Record Reader	?	CSVReader	→
Record Writer	?	splitJson	→
Records Per Split	?	1	

CSVReader



Treat First Line as Header	?	true
Ignore CSV Header Column Names	?	false

SplitJson

Suppress Null Values	?	Suppress Missing Values
----------------------	---	-------------------------

EvaluateJsonPath

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field			
Property	Value		
Destination	?	flowfile-attribute	
Return Type	?	auto-detect	
Path Not Found Behavior	?	ignore	
Null Value Representation	?	empty string	
filename	?	\$.Pais	

EvaluateJsonPath

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field			
Property	Value		
Delete Attributes Expression	?	No value set	
Store State	?	Do not store state	
Stateful Variables Initial Value	?	No value set	
Cache Value Lookup Cache Size	?	100	
filename	?	\${filename}.json	

ETL: Extract Transform Load

The background features a large, dark red wireframe sphere on the left side. Scattered across the dark blue and purple gradient background are several white plus signs and a grid of small white dots. On the right side, there are some faint red square markers and a series of white chevron symbols pointing to the right. A small, faint number '45697056' is visible in the bottom right corner of the main image area.

História sobre ETL

Com o crescimento dos bancos de dados na década de 1970, o ETL foi introduzido como um mecanismo de integração e carga de dados para ambientes computacionais onde pudessem ser analisados. Rapidamente tornaram-se o principal método de processamento de dados para **Data Warehouses**.

Foram essenciais para a análise de dados, conseguindo acumular grandes volumes de **dados estruturados**, com aplicação de regras de negócio e tratamentos específicos que viabilizaram o surgimento de **Business Intelligence**.

ETL: Como funciona?

Extração +

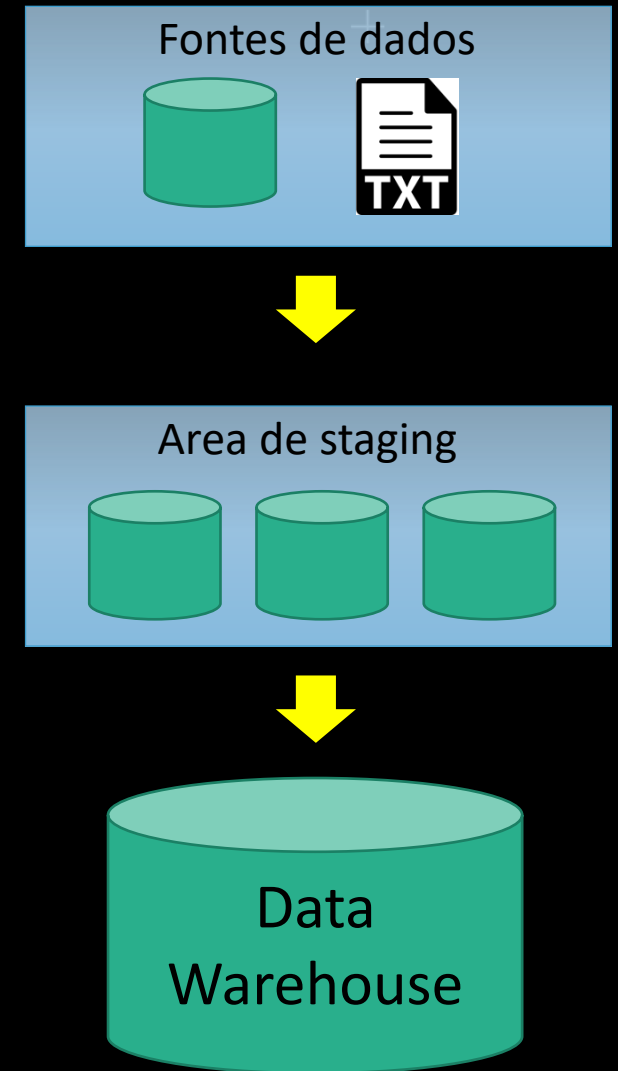
- ✓ Extração e transferência dos dados para uma área de staging
- ✓ Nesta etapa podem haver dados semi ou não estruturados
- ✓ Leituras batch de grandes volumes de dados com processos pesados e demorados

Transformação

- ✓ Nesta etapa os dados são transformados, estruturados, limpos e validados
- ✓ Aplicadas as regras de cálculo, somas, agrupamentos, de/para e demais mudanças
- ✓ Regras de governança e proteção dos dados são aplicadas

Carga +

- ✓ Os dados da staging área prontos e processados são carregados seguindo os critérios do fluxo como carga total, incremental
- ✓ Dados disponibilizados para a área de negócio com a modelagem pré-definida

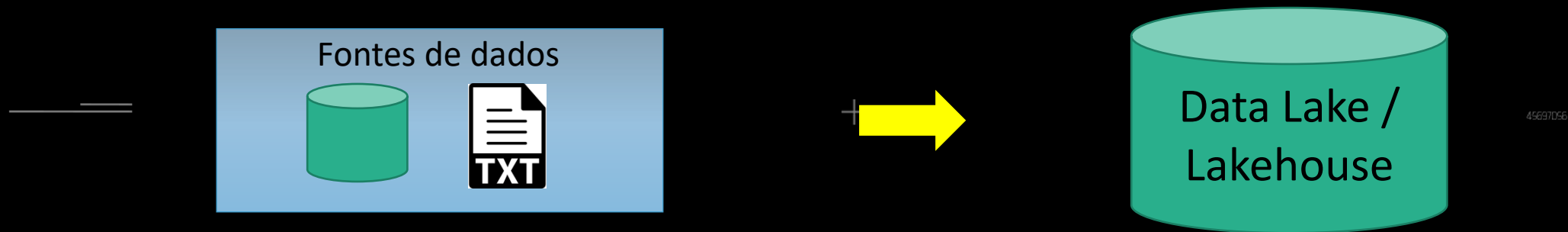


ETL x ELT

Com o surgimento do **Big Data e Data Lakes**, as plataformas de dados ganharam grande capacidade de processamento de dados não estruturados além de suportarem maiores volumes de dados.

A abordagem de ELT (Extract, Load, Transform) remove a necessidade de uma área de *staging* a parte, esse trabalho é feito dentro da própria plataforma de dados. Os dados são transformados e disponibilizados para o consumo final com maior rapidez.

Podendo suportar fluxos de dados em **tempo real**.



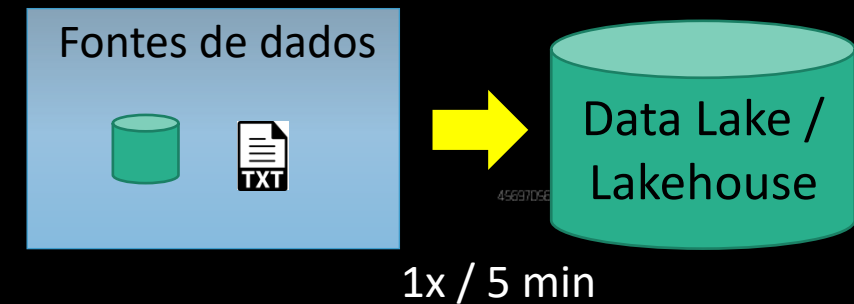
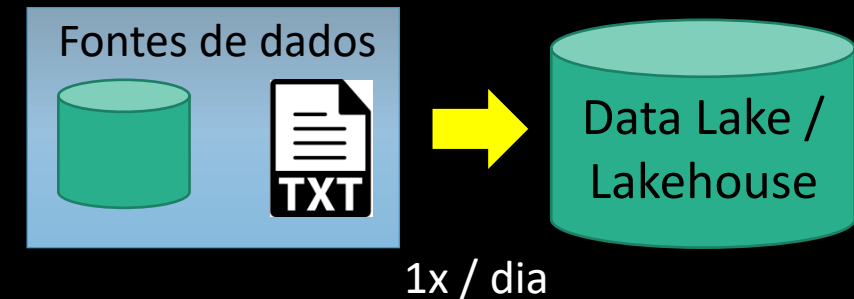
Batch x micro-batch

Micro-batches são processos que fragmentam os dados em porções menores, contudo conseguem manter uma arquitetura batch, balanceando a latência do dado sem precisar adicionar custos de uma arquitetura de streaming.

São processos de curta duração e curto intervalo entre seus ciclos (na casa de minutos).

Quando usar:

- ✓ A latência necessária para o dado pode ser atendida em minutos
- ✓ Uma alternativa mais barata para modernizar processos batch pesados
- ✓ Quando a origem e destino suportam leituras incrementais com ciclos curtos



Cases de ETL

Consolidação de dados:

- ✓ Somatória dos resultados de todos os CNPJs de um grupo de empresas
- ✓ Consolidação dos resultados mensais de uma empresa
- ✓ Agrupamento de produção de diferentes linhas de produtos

Transferência de dados:

- ✓ Integração entre sistemas (gerencial, fiscal, financeiro, contábil etc)
- ✓ Cargas batch de sistemas de front-office com back-office

Preparação de dados:

- ✓ Consolidação de diversas origens no modelo dimensional adotado pela empresa
- ✓ Identificação e tratamento de inconsistências de dados nas origens

Questão

Quais exemplos de ETL existem na empresa em que você trabalha?

Exercício: ETL



Tratar um arquivo CSV e gerar um novo arquivo CSV

Interface talend


The screenshot displays the Talend Open Studio for Data Integration (8.0.1.20211109_1610) interface. The main workspace is the Designer, showing a job design for 'teste_csv 0.1'. The job flow consists of three components: 'tFileInputDelimited_1', 'tMap_1', and 'tFileOutputDelimited_1', connected by links labeled 'row1 (Main)' and 'output (Main)'. The interface is divided into several panels:

- Left Panel (Project Tree):** Labeled 'Árvore do projeto', it shows the project structure under 'LOCAL: MyProject', including 'Job Designs', 'Contexts', 'Code', 'SQL Templates', 'Metadata', 'Documentation', and 'Recycle bin'.
- Right Panel (Component Palette):** Labeled 'Palheta de componentes', it displays a search bar with 'delimited' and a list of components categorized by 'Favorites', 'Recently Used', 'Big Data', 'Business Intelligence', 'Business', 'Cloud', 'Custom Code', and 'Data Quality'.
- Bottom Panel (Properties/Debug):** Labeled 'Propriedades, configurações, debug', it shows the 'Component' tab with the message 'Properties not available.' for the selected component.

Annotations in yellow boxes point to these specific areas: 'Árvore do projeto' (Project Tree), 'Designer' (Main workspace), 'Palheta de componentes' (Component Palette), and 'Propriedades, configurações, debug' (Properties/Debug panel).

Exercício talend

Há uma ferramenta nas estações chamada Talend Open Studio:

1. Abrir o talend, crie um novo projeto
2. Um job demo será aberto, crie um novo job chamado myjob
3. Incluir um componente do tipo tFileInputDelimited
4. Definir a localização do arquivo ipca_ibge.csv e seu separador de campos para “;”
5. Incluir um componente do tipo tMap
6. Definir o mapeamento dos campos mesAno e valor
7. Incluir um componente do tipo tFileOutputDelimited
8. Definir o nome do arquivo no destino (*)
9. Executar o job no botão run 
10. Verificar o arquivo gerado (*)

Arquivo com dados
disponibilizado

Crie o arquivo antes de defini-lo como
destino e desmarque a opção de erro
caso o arquivo exista nas opções
avancadas

O arquivo gerado é igual ao arquivo de
entrada

Continuação exercício talend...

FIAP

talend

1. Edite as propriedades do objeto tMap
2. Na categoria StringHandling usar a função EREPLACE para remover o trecho “dez/”
3. Executar o job no botão run
4. Verificar o arquivo gerado (*)

Os trechos “dez/aa” foram substituídos por “aa”

Desafio

1. Substituir o componente tFileOutputDelimited pelo tMysqlOutput
2. Subir a imagem docker do MySQL expondo a porta de conexão
3. Criar um usuário para o usar no job
4. Criar uma tabela com os campos correspondentes
5. Dar permissão ao novo usuário na tabela criada
6. Inserir o resultado do fluxo na tabela do MySQL

Tempo: 30 min

Dicas:

- Subir o docker expondo a porta 3306
- Usar o nano para editar o arquivo /etc/mysql/mysql.conf.d/mysqld.cnf e comentar 2 linhas de bind-address
- Criar o usuário na base no formato 'fiap'@'%'
- Dar permissão *.* para o usuário fiap

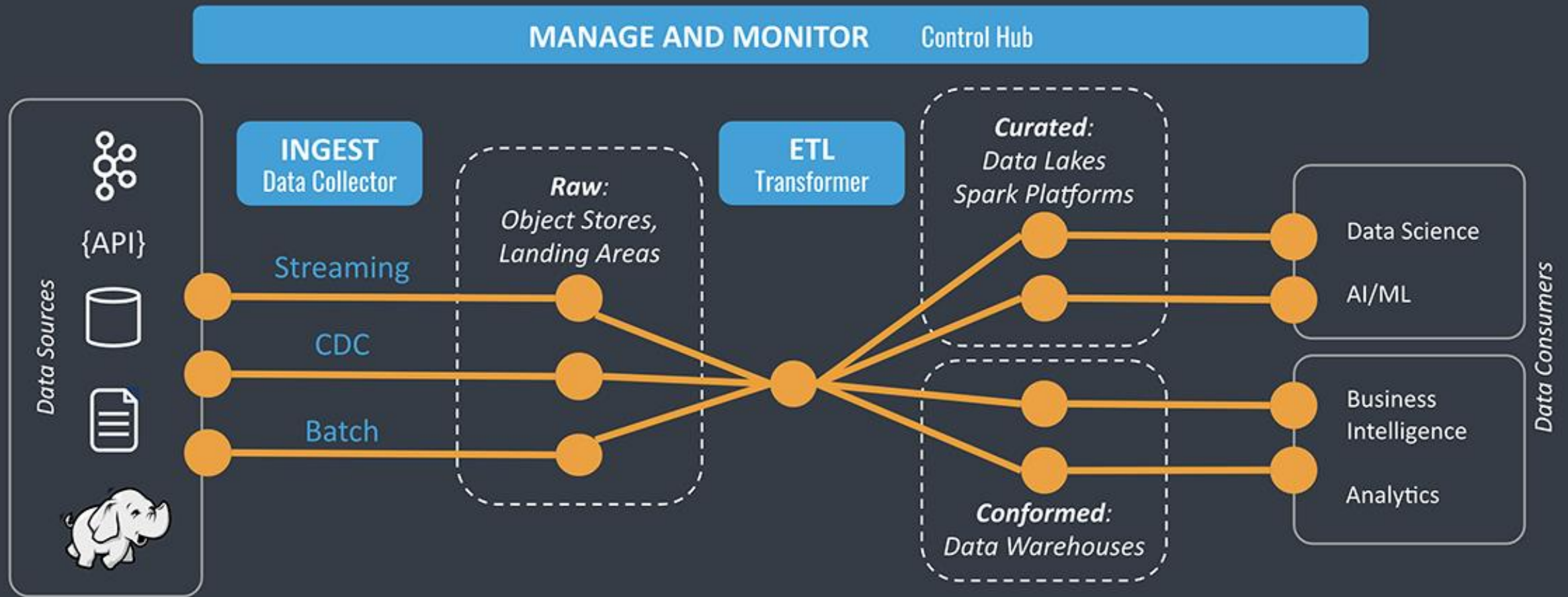
Pipeline de dados

Definição de pipeline de dados

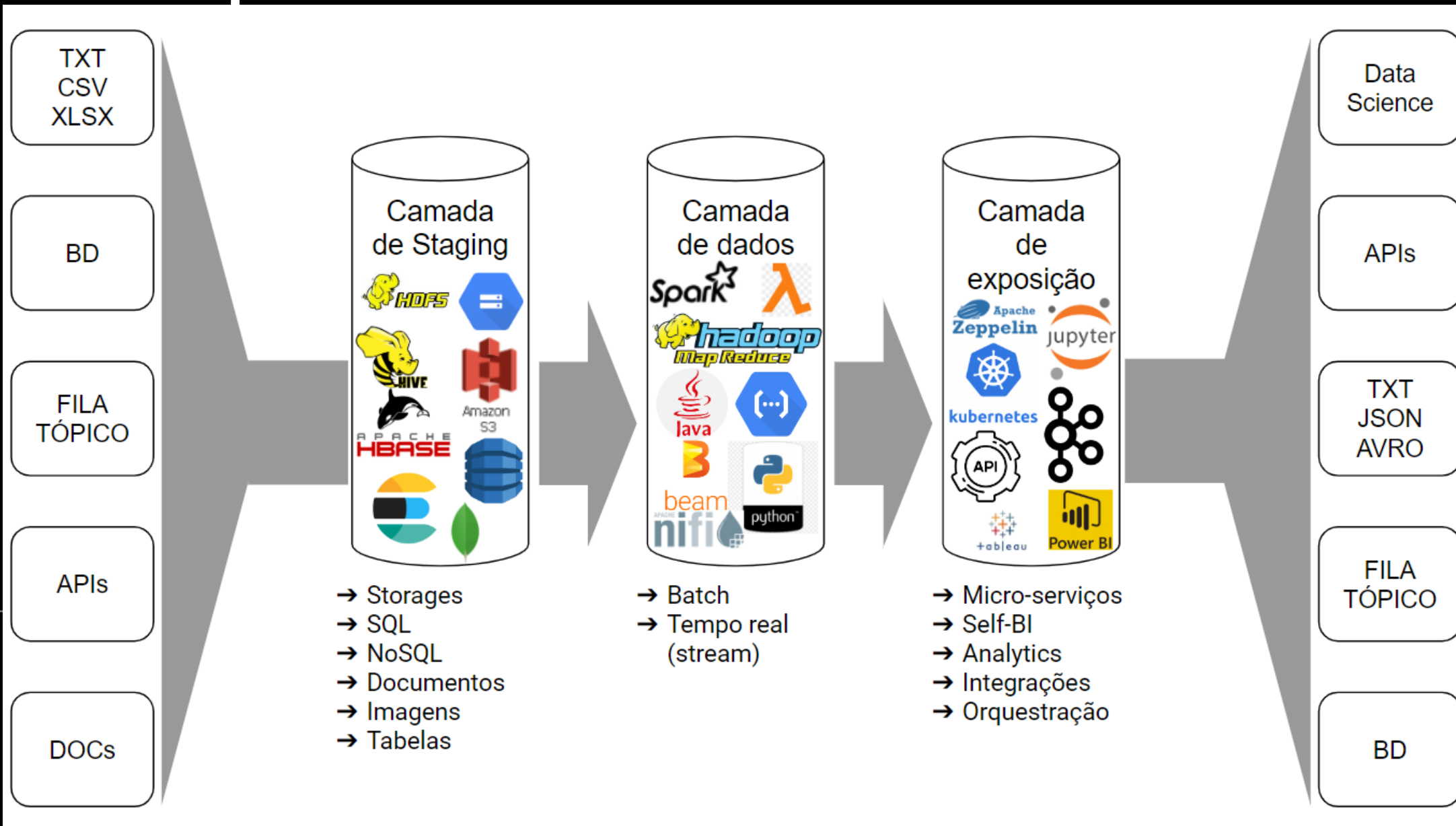
- ✓ Um **pipeline de dados** é uma combinação de passos e ferramentas usadas para automatizar um fluxo de dados desde sua origem até o consumo final do dado.
- ✓ Diferente do processo de ingestão que foca somente na carga do dado, o pipeline compreende a jornada completa do dado. Podemos considerar a ingestão como um passo de um pipeline
- ✓ Batch? Streaming? Micro-batch? ETL? ELT?
São exemplos de pipelines de dados quando tratam os dados da origem ao destino.

45697056

Pipeline de dados



Exemplos de ferramentas



Qualidade de dados

A qualidade dos dados é medida através de conjuntos de indicadores que evidenciam o quanto confiáveis estão os dados para serem utilizados.

Os indicadores mais frequentes são:

- ✓ **Acurácia:** Quanto os dados são precisos (ex: golden record do cliente)
- ✓ **Completeness:** Indica quanto o dado contempla de seu universo (ex: 100% dos PDVs)
- ✓ **Consistência:** Não há conflito entre o valor em diferentes lugares em que ele existe
- ✓ **Recência:** Quando foi sua última atualização

Contudo, mais indicadores podem ser usados como **latência, validação, conformidade, classificação de segurança**

Qualidade de dados

+

+

Benefícios

- ✓ Evitar frustrações de consumos de dados incorretos
- ✓ Elevar a confiança nos dados gerados e decisões tomadas com base nos dados
- ✓ Maior eficiência operacional de sistemas e integração com dados precisos e corretos
- ✓ Redução de custos de correções e reprocessamento de dados incorretos
- ✓ Maior satisfação do cliente através de melhor experiência com dados corretos
- ✓ Mais espaço para inovação com mais espaço para insights de negócios com dados de boa

qualidade

+

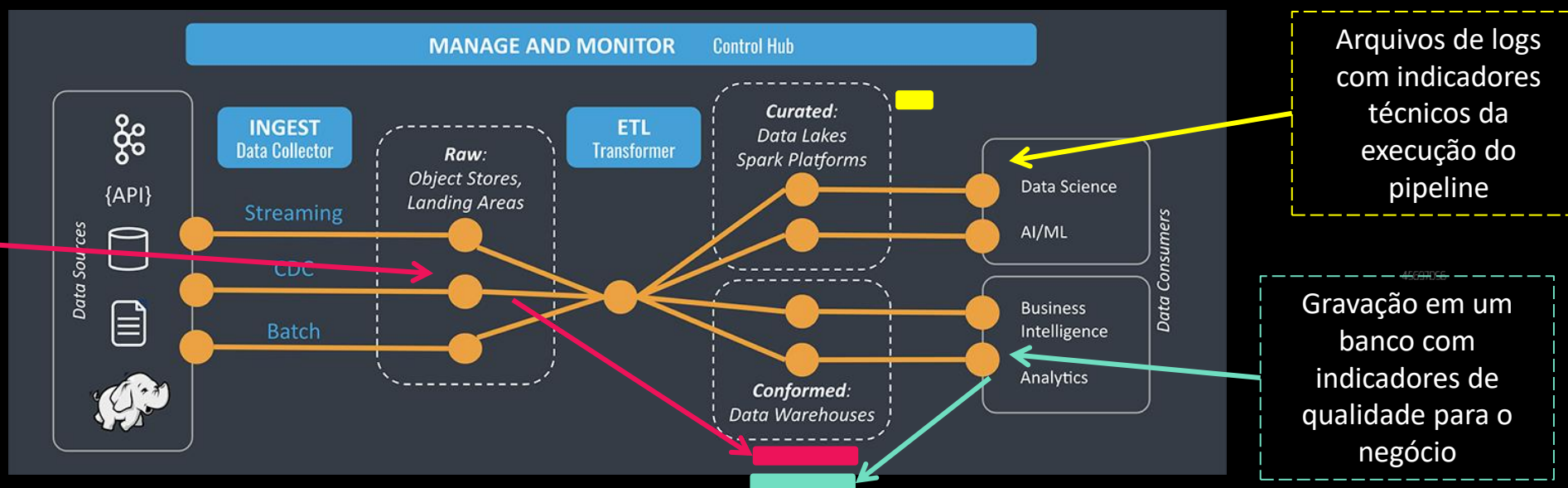
45697056

- ✓ ...

Qualidade de dados

Como é implementada?

- ✓ Em cada etapa de um pipeline podem haver oportunidades de apuração da qualidade de um dado. Por exemplo, com um controle de contagem de registros.
- ✓ A qualidade do dados é aferida através de indicadores dos dados que são armazenados nas bases de dados, catálogo de dados ou logs dos processos



Linhagem de dados

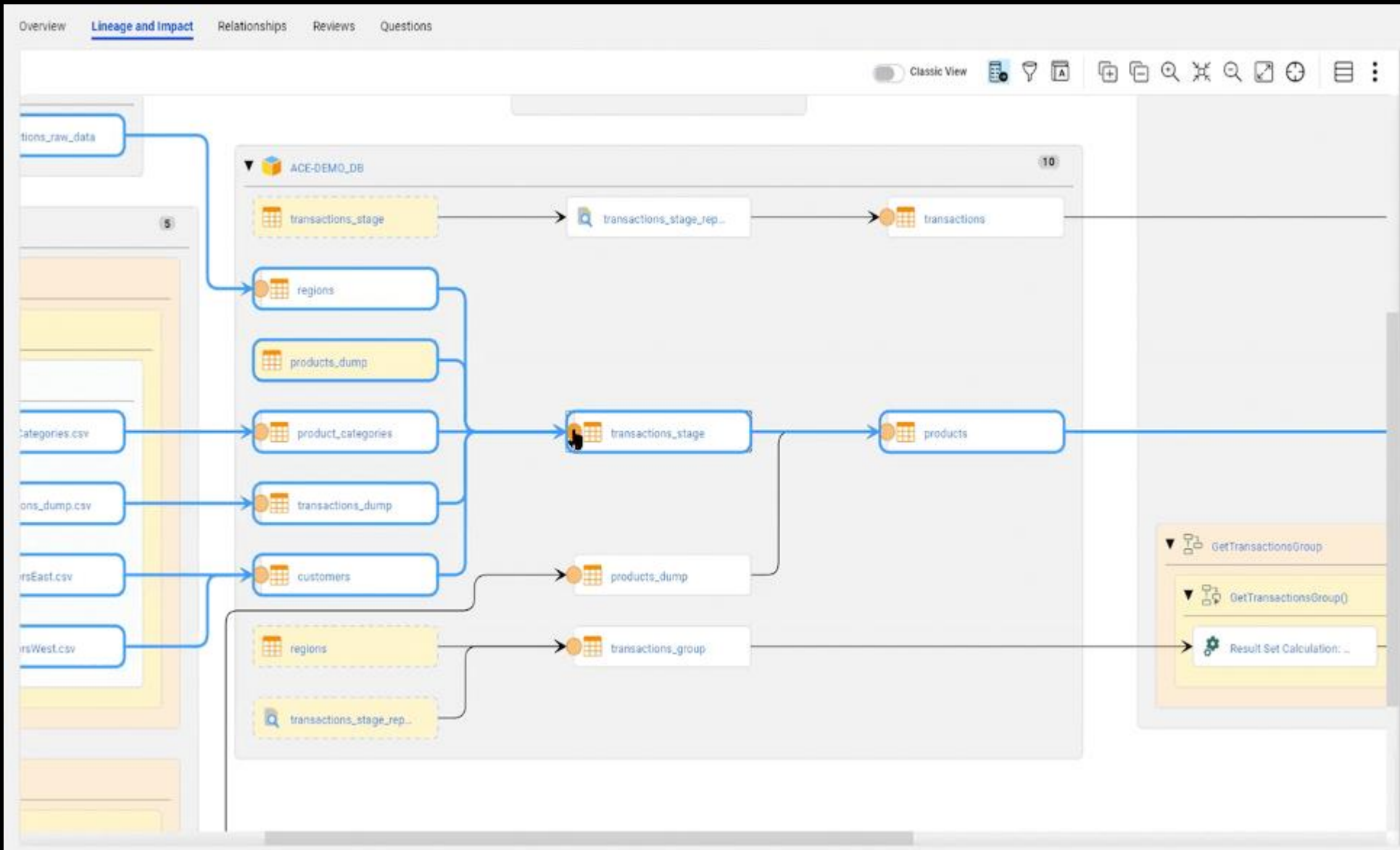
Também conhecida como “*lineage*” é o **registro completo e detalhado** dos processos pelos quais os dados são criados, manipulados, armazenados, transformados e compartilhados em um ambiente de dados. A linhagem de dados é **importante para entender a origem e o histórico dos dados**, bem como para garantir sua qualidade e governança.

A linhagem de dados geralmente é **representada em forma de diagrama** ou fluxograma, que mostra todas as etapas e transformações pelas quais os dados passaram. Isso inclui desde a fonte original dos dados até o destino final, bem como todos os sistemas, aplicativos e processos pelos quais os dados passaram ao longo do caminho.

A linhagem de dados é especialmente importante em ambientes de dados empresariais, onde há **grandes volumes de dados sendo criados, manipulados e compartilhados** por diferentes sistemas e aplicativos. A linhagem de dados permite aos usuários **entender e rastrear as relações e dependências entre os dados**, bem como identificar possíveis problemas e falhas no processo. Isso pode ajudar a garantir a qualidade e a segurança dos dados, bem como melhorar a eficiência e a eficácia das operações de dados.

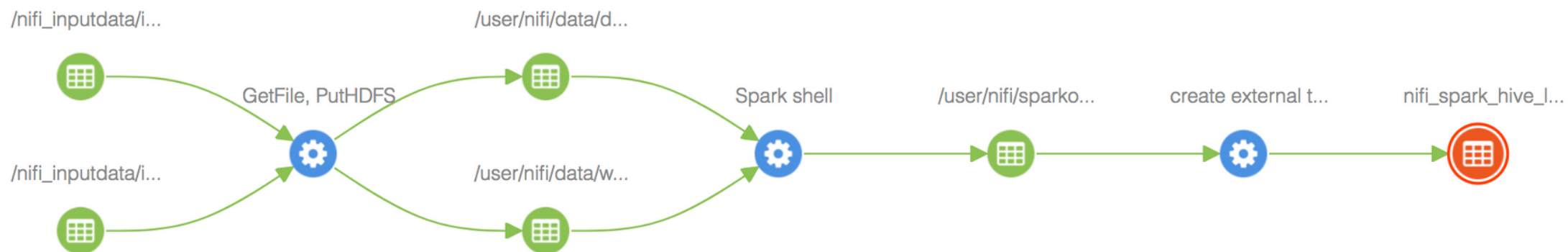
Linhagem de dados

FIAP



45697056

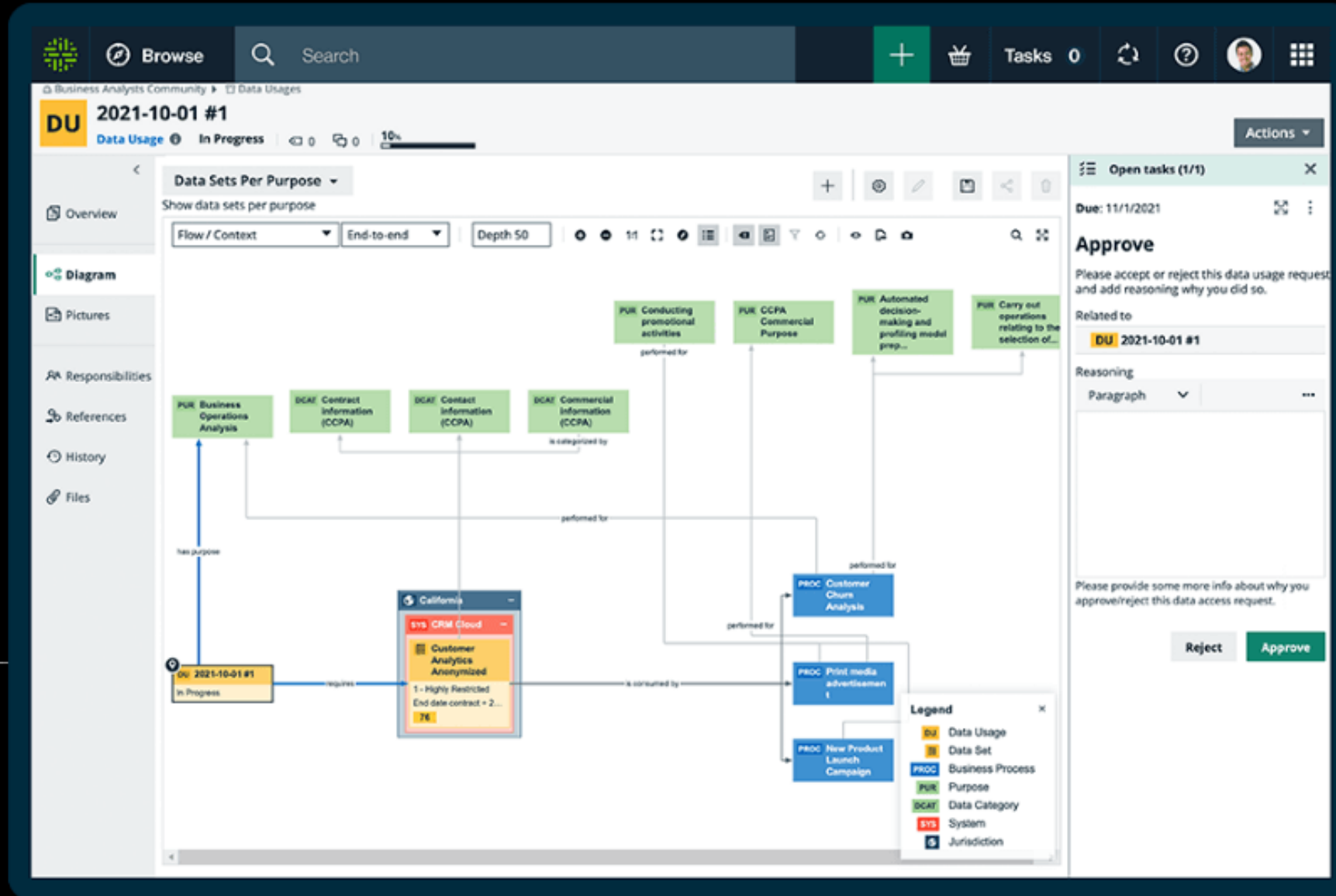
Linhagem de dados



→ Lineage → Impact

Linhagem de dados

FIAP



45697056

Questão

A linhagem de dados ajuda a garantir a qualidade dos dados de um pipeline


MBA⁺

Copyright © **2023** Profs. Ivan Gancev e Leandro Mendes

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, dos Professores Ivan Gancev e Leandro Mendes

profivan.gancev@fiap.com.br

profleandro.mendes@fiap.com.br

