



E-BOOK

# O Livro Completo da Engenharia de Dados

Uma coleção de blogs técnicos, incluindo amostras de código e cadernos

# Conteúdo

<b>SEÇÃO 1</b>	<b>Introdução à Engenharia de Dados na Databricks</b>	<b>3</b>
<b>SEÇÃO 2</b>	<b>Casos de uso reais na plataforma Lakehouse de Databricks</b>	<b>8</b>
	2.1 Análise de ponto de venda em tempo real com o Data Lakehouse	9
	2.2 Criar um Lakehouse de segurança cibernética para eventos CrowdStrike Falcon	14
	2.3 Desbloquear o poder dos dados de saúde com um Data Lakehouse moderno	19
	2.4 Pontualidade e confiabilidade na transmissão de relatórios regulatórios	24
	2.5 Soluções AML em escala usando a plataforma Databricks Lakehouse	30
	2.6 Crie um modelo de IA em tempo real para detectar comportamentos tóxicos em jogos	41
	2.7 Impulsionar a Transformação na Northwestern Mutual (Insights Platform) movendo para uma arquitetura de Lakehouse aberta e escalável	44
	2.8 Como a equipe de dados da Databricks construiu um Lakehouse em três nuvens e mais de 50 regiões	48
<b>SEÇÃO 3</b>	<b>Histórias de clientes</b>	<b>51</b>
	3.1 Atlassian	52
	3.2 ABN AMRO	54
	3.3 J.B. Hunt	56

SEÇÃO

# 01

## Introdução à Engenharia de Dados na Databricks

As organizações percebem que os dados de valor atuam como um ativo estratégico para várias iniciativas relacionadas aos negócios, como o aumento da receita, a melhoria da experiência do cliente, a operação eficiente ou a melhoria de um produto ou serviço. No entanto, o acesso e o gerenciamento de dados para essas iniciativas têm se tornado cada vez mais complexos. A maior parte da complexidade surgiu com a explosão de volumes e tipos de dados, com organizações acumulando uma estimativa de **80% dos dados em formato não estruturado e semiestruturado**. À medida que a coleta de dados continua aumentando, 73% deles não são usados para análise ou tomada de decisão. Para tentar diminuir essa porcentagem e tornar os dados mais utilizáveis, as equipes de engenharia de dados são responsáveis por criar pipelines de dados para entregá-los de forma eficiente e confiável. Mas o processo de criação desses pipelines de dados complexos traz uma série de dificuldades:

- Para colocar dados em um data lake, os engenheiros de dados são obrigados a gastar um tempo imenso codificando manualmente tarefas de ingestão de dados repetitivas
- Uma vez que as plataformas de dados mudam continuamente, os engenheiros de dados gastam tempo construindo e mantendo e, em seguida, reconstruindo, uma infraestrutura escalável complexa.
- Com a crescente importância dos dados em tempo real, são necessários pipelines de dados de baixa latência, que são ainda mais difíceis de construir e manter.
- Por fim, com todos os pipelines escritos, os engenheiros de dados precisam se concentrar constantemente no desempenho, ajustando pipelines e arquiteturas para atender aos SLAs.

## Como a Databricks pode ajudar?

Com a Plataforma Databricks Lakehouse, os engenheiros de dados têm acesso a uma solução de engenharia de dados de ponta a ponta para ingerir, transformar, processar, agendar e entregar dados. A Plataforma Lakehouse automatiza a complexidade de construir e manter pipelines e executar cargas de trabalho ETL diretamente em um data lake para que os engenheiros de dados possam se concentrar na qualidade e confiabilidade para gerar insights valiosos.

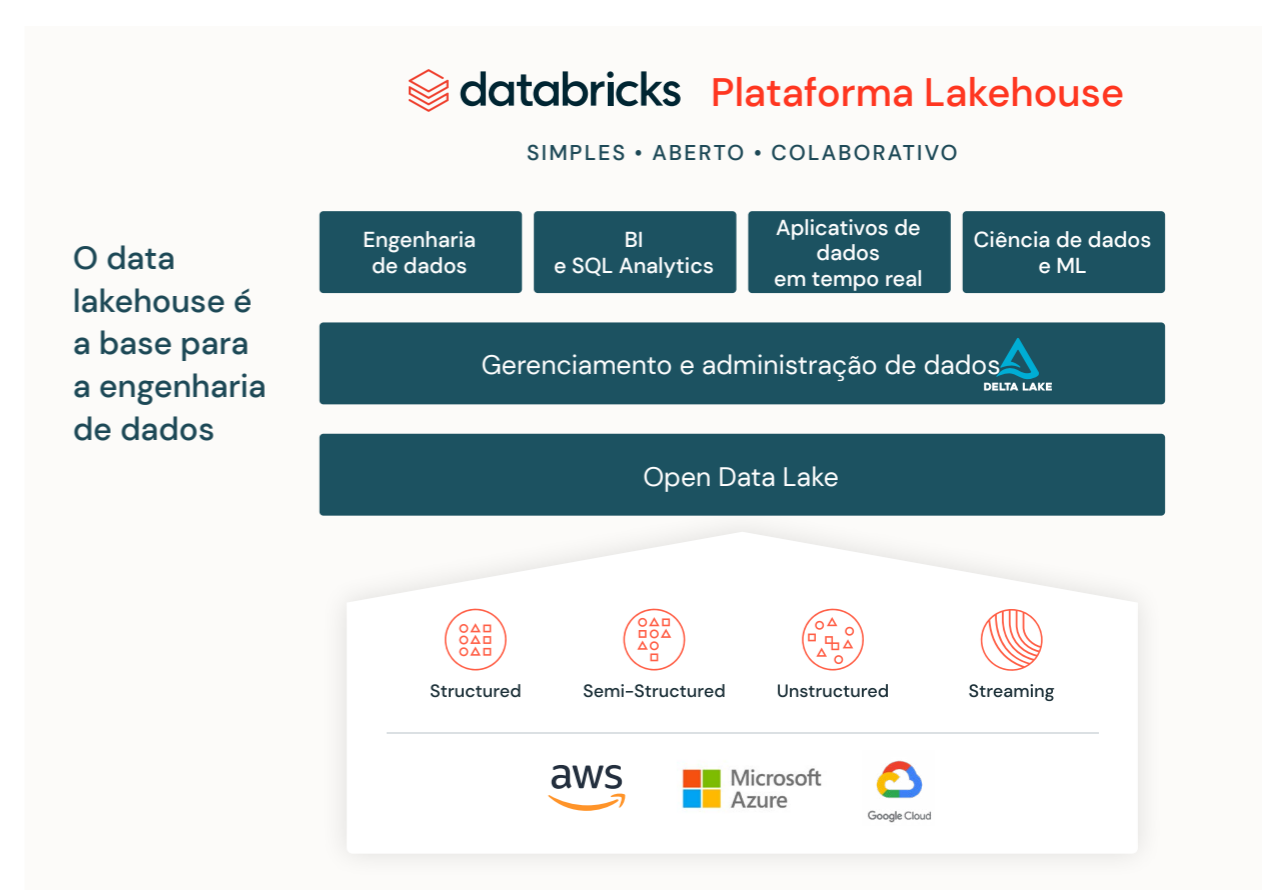


Figura 1

A Plataforma Databricks Lakehouse unifica seus dados, análises e IA em uma plataforma comum para todos os seus casos de uso de dados

## Principais diferenciais para engenharia de dados bem-sucedida com Databricks

Ao simplificar uma arquitetura de lakehouse, os engenheiros de dados precisam de uma abordagem pronta e de nível empresarial para criar pipelines de dados. Para ter sucesso, uma equipe de soluções de engenharia de dados deve adotar estes oito principais recursos diferenciais:

### Ingestão contínua ou programada de dados

Com a capacidade de ingerir petabytes de dados com esquemas de evolução automática, os engenheiros de dados podem fornecer dados rápidos, confiáveis, escaláveis e automáticos para análise, ciência de dados ou machine learning. Isso inclui:

- Processar dados de forma progressiva e eficiente à medida que chegam de arquivos ou fontes de streaming como Kafka, DBMS e NoSQL
- Inferir automaticamente o esquema e detectar alterações de coluna para formatos de dados estruturados e não estruturados
- Rastrear dados de forma automática e eficiente à medida que eles chegam sem intervenção manual
- Evitar a perda de dados resgatando colunas de dados

### Pipelines de ETL declarativos

Os engenheiros de dados podem reduzir o tempo e o esforço de desenvolvimento e se concentrar na implementação de lógica de negócios e verificações de qualidade de dados no pipeline de dados usando SQL ou Python. Isso pode ser alcançado por meio de:

- Uso do desenvolvimento declarativo orientado por intenção para simplificar o “como” e definir “o que” resolver
- Criação automática de linhagem de alta qualidade e gerenciamento de dependências de tabela em todo o pipeline de dados
- Verificação automática de dependências ausentes ou erros de sintaxe e gerenciamento de recuperação do pipeline de dados

### Validação e monitoramento da qualidade dos dados

Melhore a confiabilidade dos dados em todo o data lakehouse para que as equipes de dados possam confiar nas informações para iniciativas de downstream através de:

- Definição de controles de integridade e qualidade de dados dentro do pipeline com expectativas de dados definidas
- Abordagem de erros de qualidade de dados com políticas predefinidas (falha, queda, alerta, quarentena)
- Alavancagem das métricas de qualidade de dados que são capturadas, rastreadas e comunicadas para todo o pipeline de dados

### Recuperação automática e tolerante a falhas

Manuseie erros transitórios e recupere-se das condições de erro mais comuns que ocorrem durante a operação de um pipeline com recuperação automática rápida e escalonável, que inclui:

- Mecanismos tolerantes a falhas para recuperar consistentemente o estado dos dados
- Capacidade de rastrear automaticamente o progresso da fonte com pontos de verificação
- Capacidade de recuperar e restaurar automaticamente o estado do pipeline de dados

### Observabilidade do pipeline de dados

Monitore o status geral do pipeline de dados de um painel de gráfico de fluxo de dados e monitore visualmente a integridade do pipeline de ponta a ponta para desempenho, qualidade e latência. Os recursos de observabilidade do pipeline de dados incluem:

- Um diagrama de linhagem de alta qualidade e alta fidelidade que fornece visibilidade sobre como os dados fluem para análise de impacto
- Registro granular com desempenho e status do pipeline de dados em nível de linha
- Monitoramento contínuo dos trabalhos de pipeline de dados para garantir operação contínua

### Processamento em batch e fluxo de dados

Permita que engenheiros de dados ajustem a latência de dados com controles de custo sem a necessidade de conhecer o processamento de fluxo complexo ou implementar lógica de recuperação.

- Execute cargas de trabalho de pipeline de dados em compute clusters baseados em Apache Spark™ flexíveis e automaticamente provisionados para escala e desempenho
- Use clusters de otimização de desempenho que paralelizam trabalhos e minimizam o movimento de dados

### Implementações e operações automáticas

Garanta uma entrega confiável e previsível de dados para casos de uso de funções analíticas e machine learning, permitindo implementações e reversões de pipeline de dados fáceis e automáticas para minimizar o tempo de inatividade. Os benefícios incluem:

- Implementação completa, parametrizada e automatizada para a entrega contínua de dados
- Orquestração, testes e monitoramento de ponta a ponta da implementação do pipeline de dados em todos os principais provedores de nuvem

### Pipelines e fluxos de trabalho programados

Orquestração simples, clara e confiável de tarefas de processamento de dados para pipelines de dados e machine learning com a capacidade de executar várias tarefas não interativas como um gráfico acíclico direcionado (DAG) em um cluster de processamento Databricks.

- Organize facilmente tarefas em um DAG usando a interface e a API da Databricks
- Crie e gerencie múltiplas tarefas em trabalhos via UI ou API e recursos, como alertas por e-mail para monitoramento
- Organize qualquer tarefa que tenha uma API fora da Databricks e através de todas as nuvens

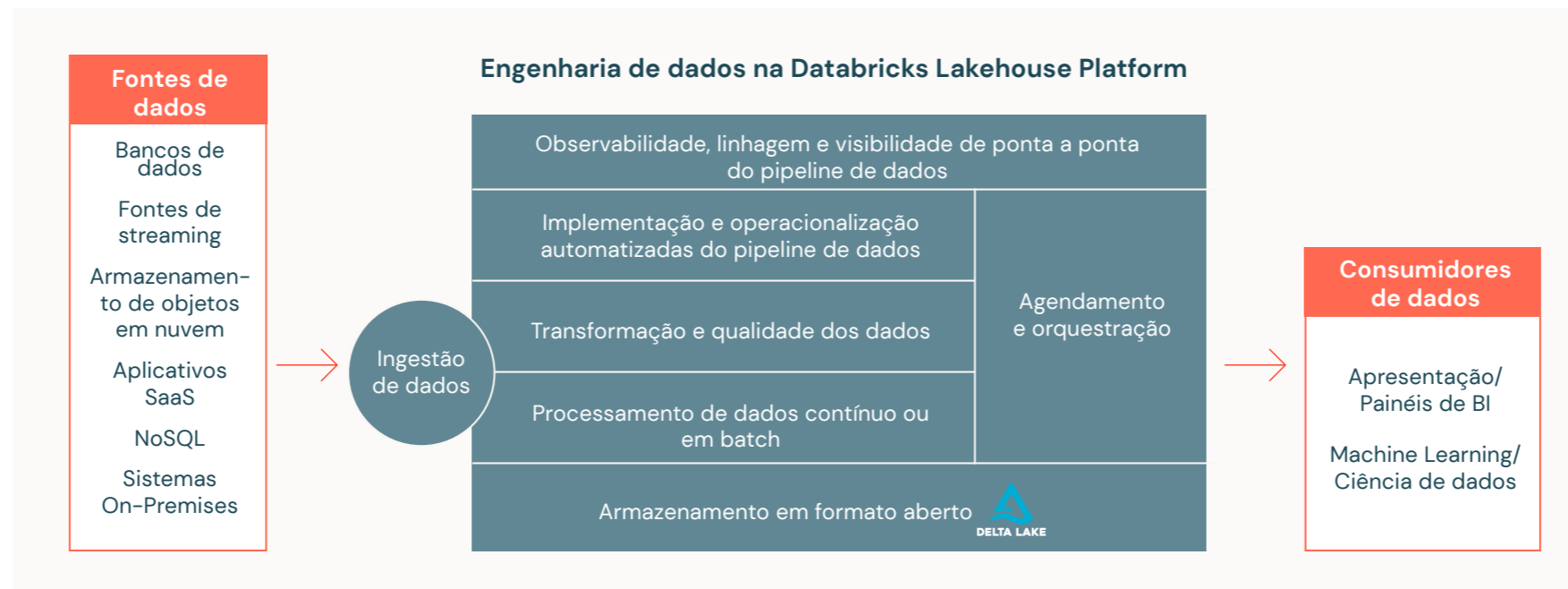


Figura 2  
Engenharia de dados na arquitetura de referência da Databricks

## Conclusão

À medida que as organizações se esforçam para se tornarem orientadas por dados, a engenharia de dados é um ponto focal para o sucesso. Para fornecer dados confiáveis, os engenheiros de dados não precisam gastar tempo desenvolvendo e mantendo manualmente um ciclo de vida ETL de ponta a ponta. As equipes de engenharia de dados precisam de uma maneira eficiente e escalável de simplificar o desenvolvimento de ETL, melhorar a confiabilidade dos dados e gerenciar as operações.

Conforme descrito, os oito principais recursos de diferenciação simplificam o gerenciamento do ciclo de vida de ETL, automatizando e mantendo todas as dependências de dados, aproveitando os controles de qualidade integrados com monitoramento e fornecendo visibilidade profunda das operações de

pipeline com recuperação automática. As equipes de engenharia de dados agora podem se concentrar na criação fácil e rápida de pipelines de dados prontos para produção de ponta a ponta e confiáveis usando apenas SQL ou Python em batch e streaming que fornecem dados de alto valor para funções analíticas, ciência de dados ou machine learning.

## Casos de uso

Na próxima seção, descrevemos as melhores práticas para casos de uso de ponta a ponta da engenharia de dados extraídos de exemplos do mundo real. Da ingestão e do processamento de dados até funções analíticas e machine learning, você aprenderá como converter dados brutos em dados acionáveis. Nós o orientaremos com os conjuntos de dados e amostras de código, para que você possa colocar a mão na massa enquanto explora todos os aspectos do ciclo de vida dos dados na Plataforma Databricks Lakehouse.

SEÇÃO

# 02

## Casos de uso reais na plataforma Databricks Lakehouse

Análise de ponto de venda em tempo real com o Data Lakehouse

Construindo um Lakehouse de cibersegurança para eventos CrowdStrike Falcon

Desbloqueando o poder dos dados de saúde com um Data Lakehouse moderno

Prontidão e confiabilidade na transmissão de relatórios regulamentares

Soluções AML em escala usando a plataforma Databricks Lakehouse

Crie um modelo de IA em tempo real para detectar comportamentos tóxicos em jogos

Promover a transformação na Northwestern Mutual (Plataforma Insights) movendo-se em direção a uma arquitetura de Lakehouse escalável e aberta

Como a equipe de dados da Databricks construiu um lakehouse em três nuvens e mais de 50 regiões



## SEÇÃO 2.1 **Análise de ponto de venda em tempo real com o Data Lakehouse**

de **BRYAN SMITH** e **ROB SAKER**

9 de setembro de 2021

Interrupções na cadeia de suprimentos — da redução da oferta de produtos e da diminuição da capacidade de armazenar — juntamente com as expectativas dos consumidores em rápida mudança por **experiências omnichannel** perfeitas estão levando os varejistas a repensar como eles usam dados para gerenciar suas operações. Antes da pandemia, **71% dos varejistas** apontavam a falta de visibilidade do estoque em tempo real como um dos principais obstáculos para alcançar suas metas omnichannel. A pandemia só aumentou **a demanda por experiências integradas online e na loja**, colocando ainda mais pressão sobre os varejistas para apresentar disponibilidade precisa de produtos e gerenciar mudanças de pedidos em tempo real. Um melhor acesso às informações em tempo real é essencial para atender às demandas dos consumidores no novo normal.

Neste blog, abordaremos a necessidade de dados em tempo real no varejo, e como superar os desafios de movimentação em tempo real do fluxo de dados no ponto de venda em escala com um data lakehouse.

### O sistema de ponto de venda

Há muito tempo, o sistema de ponto de venda (PDV) tem sido a peça central da infraestrutura na loja, registrando a troca de produtos e serviços entre o varejista e o cliente. Para sustentar essa troca, o PDV normalmente rastreia os estoques de produtos e facilita a reposição à medida que a contagem de unidades cai

abaixo dos níveis críticos. A importância do PDV para as operações na loja não pode ser exagerada, e como o sistema de registro de operações de vendas e estoque, o acesso aos seus dados é de interesse fundamental para os analistas de negócios.

Historicamente, a conectividade limitada entre lojas individuais e escritórios corporativos significava que o sistema de PDV (não apenas suas interfaces de terminal) residia fisicamente na loja. Durante o horário de pico, esses sistemas podem telefonar para casa para transmitir dados de resumo que, quando consolidados em um data warehouse, fornecem uma visão antiga do desempenho das operações de varejo que fica cada vez mais obsoleta até o início do ciclo da próxima noite.



Figura 1

Disponibilidade do inventário com padrões ETL tradicionais orientados por batch

As modernas melhorias de conectividade permitiram que mais varejistas mudassem para um sistema de PDV centralizado e baseado em nuvem, enquanto muitos outros estão desenvolvendo integrações quase em tempo real entre os sistemas da loja e o back office corporativo. A disponibilidade de informações quase em tempo real significa que os varejistas podem atualizar continuamente suas estimativas de disponibilidade de itens. A gestão de negócios não está mais contra o conhecimento das condições do inventário como já esteve antes. Em vez disso, está adotando medidas baseadas em seu conhecimento das condições de inventário como elas são agora.

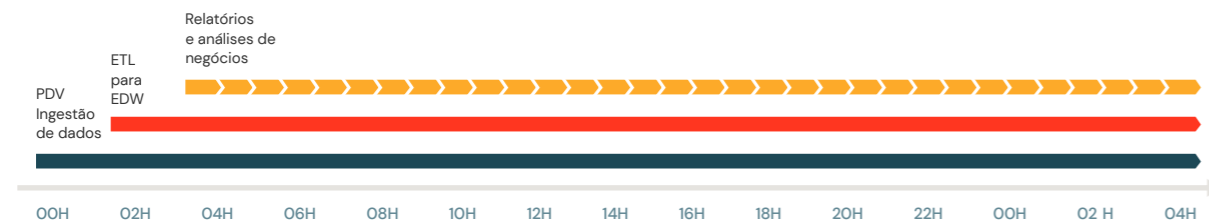


Figura 2  
Disponibilidade de inventário com padrões ETL de streaming

## Insights quase em tempo real

Por mais impactante que seja a percepção quase em tempo real da atividade da loja, a transição dos processos noturnos para o fluxo contínuo de informações traz desafios particulares não apenas para o engenheiro de dados, que deve projetar um tipo diferente de fluxo de trabalho de processamento de dados, mas também para o consumidor de informações. Neste post, nós compartilhamos algumas lições aprendidas de clientes que embarcaram recentemente nesta

jornada e examinamos como padrões e capacidades essenciais disponíveis através do **padrão** de lakehouse podem permitir o sucesso.

### AULA 1 Considere o escopo cuidadosamente

Os sistemas de PDV muitas vezes não se limitam apenas a vendas e gerenciamento de estoque. Em vez disso, eles podem fornecer uma ampla gama de funcionalidades, incluindo processamento de pagamentos, gerenciamento de crédito de loja, realização de faturamento e pedidos, gerenciamento de programas de fidelidade, agendamento de funcionários, rastreamento de horários e até folha de pagamento, tornando-os um verdadeiro canivete suíço de funcionalidade na loja.

Como resultado, os dados alojados dentro do PDV são tipicamente espalhados por uma grande e complexa estrutura de banco de dados. Se tiver sorte, a solução PDV disponibiliza uma camada de acesso aos dados, o que os torna acessíveis através de estruturas de interpretação mais fácil. Caso contrário, o engenheiro de dados deve analisar o que pode ser um conjunto opaco de tabelas para determinar o que é valioso e o que não é.

Independentemente de como os dados são expostos, a orientação clássica se aplica: identifique uma justificativa de negócio convincente para sua solução e use-a para limitar o escopo dos ativos de informação que você consome inicialmente. Tal justificativa muitas vezes vem de um forte responsável pelo negócio, que é encarregado de enfrentar um desafio corporativo específico e vê a disponibilidade de informações mais oportunas como algo fundamental para seu sucesso.

Para ilustrar isso, considere um desafio fundamental para muitas organizações varejistas hoje: a habilitação de soluções omnichannel. Tais soluções, que permitem transações BOPIS ("buy- online, pick up in-store"; ou compre online, retire na loja, em português) e cross-store dependem de informações razoavelmente precisas sobre o inventário da loja. Se limitarmos nosso escopo inicial a essa necessidade, os requisitos de informação para nosso sistema de monitoramento e análise se tornam drasticamente reduzidos. Uma vez que uma solução de inventário em tempo real é entregue e o valor é reconhecido pelo negócio, podemos expandir nosso escopo para considerar outras necessidades, como monitoramento de promoções e detecção de fraudes, ampliando a variedade de ativos de informação alavancados a cada iteração.

## AULA 2 Alinhe a transmissão com padrões de geração de dados e suscetibilidades de tempo

Processos diferentes geram dados de forma diferente dentro do PDV. É provável que as transações de vendas deixem uma trilha de novos registros anexados às tabelas relevantes. As devoluções podem seguir vários caminhos, desencadeando atualizações nos registros de vendas anteriores, inserção de novos registros de vendas reversíveis e/ou inserção de novas informações em estruturas específicas de devoluções. Documentação do fornecedor, conhecimento tribal e até mesmo algum trabalho investigativo independente podem ser necessários para descobrir exatamente como e onde informações específicas do evento chegam ao PDV.

Compreender esses padrões pode ajudar a construir uma estratégia de transmissão de dados para tipos específicos de informações. Padrões de frequência mais elevados, mais finos e orientados a inserções podem ser idealmente adequados para o fluxo contínuo. Eventos menos frequentes e de maior escala podem se alinhar melhor com estilos de transmissão de dados em massa orientados para batch. Mas se esses modos de transmissão de dados representam duas extremidades de um espectro, é provável que você encontre a maioria dos eventos capturados pelo PDV em algum lugar entre elas.

A vantagem da abordagem de data lakehouse para a arquitetura de dados é que **vários modos de transmissão de dados** podem ser empregados em paralelo. Para dados naturalmente alinhados com a transmissão contínua, o streaming pode ser empregado. Para dados mais bem alinhados com a transmissão em massa, processos em batch podem ser usados. E para esses dados que estão entre os dois, você pode se concentrar na prontidão dos dados necessários para tomada de decisão, e permitir que isso mostre o caminho a seguir. Todos esses modos podem ser enfrentados com uma abordagem consistente para a implementação de ETL, um desafio que provocou muitas implementações anteriores do que era frequentemente referido como **arquiteturas Lambda**.

## AULA 3

## Obtenha os dados em etapas

Os dados chegam dos sistemas de PDV na loja com diferentes frequências, formatos e expectativas de disponibilidade oportuna. Aproveitando o **padrão de design Bronze, Silver e Gold**, popular em lakehouses, você pode separar a limpeza inicial, a reformatação e a persistência dos dados das transformações mais complexas necessárias para entregas específicas alinhadas aos negócios.

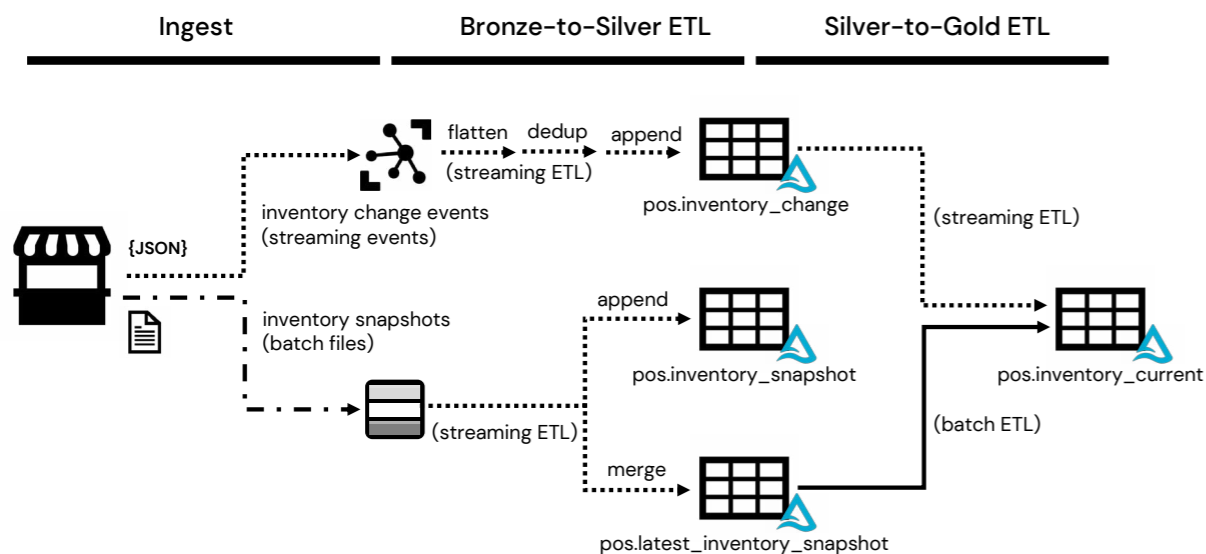


Figura 3

Uma arquitetura de data lakehouse para o cálculo do inventário atual utilizando o padrão Bronze, Silver e Gold de persistência de dados

## AULA 4

## Gerenciar expectativas

A mudança para funções analíticas quase em tempo real exige uma mudança organizacional. Gartner descreve isso por meio de seu **modelo de Maturidade em Analytics**, em que a análise de dados de streaming se integra à estrutura das operações diárias. Isso não acontece de um dia para o outro.

Em vez disso, os engenheiros de dados precisam de tempo para reconhecer os desafios inerentes à entrega de streaming de lojas físicas para um back office centralizado e baseado em nuvem. Melhorias na conectividade e confiabilidade do sistema, juntamente com fluxos de trabalho ETL cada vez mais robustos, obtêm dados com maior prontidão, confiabilidade e consistência. Isso muitas vezes implica aprimorar as parcerias com engenheiros de sistemas e desenvolvedores de aplicativos para oferecer suporte a um nível de integração que normalmente não está presente nos dias de fluxos de trabalho ETL somente em batch.

Os analistas de negócios precisarão se familiarizar com o ruído inerente aos dados que estão sendo atualizados continuamente. Eles precisarão reaprender como realizar trabalho de diagnóstico e validação de um conjunto de dados, como quando uma consulta que ocorreu segundos antes que agora retorna com um resultado ligeiramente diferente. Eles devem adquirir maior consciência dos problemas nos dados que muitas vezes estão escondidos quando apresentados em agregados diários. Tudo isso exigirá ajustes, tanto em suas análises quanto em sua resposta aos sinais detectados em seus resultados.

Tudo isso ocorre apenas nos primeiros estágios de maturação. Em etapas posteriores, a capacidade da organização de detectar sinais significativos dentro do fluxo pode levar a uma maior capacidade de sentido e resposta automatizada. Aqui, os níveis mais altos de valor nos fluxos de dados são desbloqueados. Mas o monitoramento e a governança devem ser implementados e comprovados antes que o negócio confie suas operações a essas tecnologias.

## Implementação de streaming de PDV

Para ilustrar como a arquitetura lakehouse pode ser aplicada aos dados de PDV, desenvolvemos um fluxo de trabalho de demonstração no qual calculamos um inventário quase em tempo real. Nele, visualizamos dois sistemas PDV separados transmitindo informações relevantes de estoque associadas a vendas, reabastecimentos e dados de redução, juntamente com transações de compra online, retire na loja (BOPIS) (iniciadas em um sistema e completadas em outro) como parte de um feed de alteração de inventário de streaming. As contagens periódicas (instantâneos) de unidades de produtos na prateleira são capturadas pelo PDV e transmitidas em massa. Esses dados são simulados por um período de um mês e reproduzidos em velocidade 10x maior para aumentar a visibilidade das alterações de inventário.

Os processos de ETL (como ilustrado na Figura 3) representam uma mistura de técnicas de streaming e batch. Uma abordagem em duas etapas com dados minimamente transformados capturados em tabelas Delta representando nossa camada Silver separa nossa abordagem de ETL inicial e mais tecnicamente alinhada com a abordagem mais alinhada aos negócios necessária para os cálculos de inventário atuais. A segunda etapa foi implementada usando os recursos tradicionais de streaming estruturado, algo que podemos revisar com a nova funcionalidade de **Delta Live Tables** à medida que ela se torna mais disponível.

A demonstração usa o Azure IOT Hubs e o Azure Storage para ingestão de dados, mas funcionaria da mesma forma nas nuvens AWS e GCP com substituições de tecnologia apropriadas.

### Comece a experimentar com estes notebooks Databricks gratuitos



- **PDV 01: Configuração do ambiente**
- **PDV 02: Geração de dados**
- **POS 03: Ingestão ETL**
- **PDV 04: Inventário atual**

## SEÇÃO 2.2 Criar um lakehouse de segurança cibernética para eventos CrowdStrike Falcon

de AEMRO AMARE, ARUN PAMULAPATI,  
YONG SHENG HUANG e JASON POHL

20 de maio de 2021

Os dados de endpoints são exigidos pelas equipes de segurança para detecção e caça de ameaças, investigações de incidentes e para atender aos requisitos de conformidade. Os volumes de dados podem ser terabytes por dia ou petabytes por ano. A maioria das organizações luta para coletar, armazenar e analisar logs de endpoints devido aos custos e complexidades associados a esses grandes volumes de dados. Mas não precisa ser assim.

Nesta série de blogs em duas partes, abordaremos como você pode operacionalizar petabytes de dados de endpoint com a Databricks para melhorar sua postura de segurança com análises avançadas de forma econômica. A Parte 1 (deste blog) abordará a arquitetura da coleta de dados e a integração com um SIEM (Splunk). No final deste blog, com notebooks fornecidos, você estará pronto para usar os dados para análise. A Parte 2 discutirá casos de uso específicos, como criar modelos de ML e enriquecimentos e funções analíticas automatizadas. Ao final da parte 2, você poderá implementar os notebooks para detectar e investigar ameaças usando dados de endpoint.

Usaremos os logs Falcon do CrowdStrike como exemplo. Para acessar logs Falcon, é possível usar o Falcon Data Replicator (FDR) para enviar dados brutos de eventos da plataforma CrowdStrike para o armazenamento em nuvem, como

o Amazon S3. Esses dados podem ser ingeridos, transformados, analisados e armazenados usando a plataforma Databricks Lakehouse juntamente com o restante de sua telemetria de segurança. Os clientes podem ingerir dados CrowdStrike Falcon, aplicar detecções em tempo real baseadas em Python, pesquisar dados históricos com Databricks SQL e consultar ferramentas SIEM como Splunk com extensão Databricks para Splunk.

### Desafio de operacionalizar dados do CrowdStrike

Embora os dados do CrowdStrike Falcon ofereçam detalhes completos de registro de eventos, é uma tarefa assustadora ingerir, processar e operacionalizar volumes complexos e grandes de dados de segurança cibernética quase em tempo real e de forma econômica. Estes são alguns dos desafios mais conhecidos:

- **Ingestão de dados em tempo real em escala:** é difícil manter o controle dos arquivos de dados brutos processados e não processados, que são gravados pelo FDR no armazenamento em nuvem quase em tempo real.
- **Transformações complexas:** o formato de dados é semiestruturado. Cada linha de cada arquivo de registro contém centenas de tipos de payloads diferentes, e a estrutura dos dados de eventos pode mudar ao longo do tempo.



- **Governança de dados:** esse tipo de dado pode ser confidencial, e o acesso deve ser limitado somente aos usuários que precisam deles.
- **Análise de segurança simplificada de ponta a ponta:** ferramentas escaláveis são necessárias para fazer a engenharia de dados, o ML e análises sobre esses conjuntos de dados rápidos e de alto volume.
- **Colaboração:** Uma colaboração eficaz pode aproveitar a expertise dos engenheiros de dados, analistas de segurança cibernética e engenheiros de ML. Assim, ter uma plataforma colaborativa melhora a eficiência da análise de segurança cibernética e cargas de trabalho de resposta.

Como resultado, engenheiros de segurança de todas as empresas se encontram em uma situação difícil, lutando para gerenciar o custo e a eficiência operacional. Ou eles aceitam estar trancados em sistemas proprietários muito caros ou se esforçam para construir suas próprias ferramentas de segurança de endpoint enquanto lutam por escalabilidade e desempenho.

## Lakehouse de segurança cibernética Databricks

A Databricks oferece às equipes de segurança e aos cientistas de dados uma nova esperança de desempenhar seus trabalhos de forma eficiente e eficaz, além de um conjunto de ferramentas para combater os crescentes desafios de big data e ameaças sofisticadas.

**Lakehouse**, uma arquitetura aberta que combina os melhores elementos de data lakes e data warehouses, simplifica a criação de um pipeline de engenharia

de dados multi-hop que adiciona estrutura aos dados progressivamente. O benefício de uma arquitetura multi-hop é que os engenheiros de dados podem construir um pipeline que começa com dados brutos como uma "fonte única da verdade", a partir da qual tudo flui. Os dados brutos semiestruturados do CrowdStrike podem ser armazenados por anos, e transformações e agregações subsequentes podem ser feitas em streaming de ponta a ponta, para refinar os dados e introduzir estrutura específica de contexto para analisar e detectar riscos à segurança em diferentes cenários.

- **Ingestão de dados:** o **Auto Loader** (**AWS** | **Azure** | **GCP**) ajuda a ler dados imediatamente assim que um novo arquivo é escrito pelo CrowdStrike FDR no armazenamento de dados brutos. Ele aproveita os serviços de notificação em nuvem para processar incrementalmente novos arquivos à medida que eles chegam à nuvem. O Auto Loader também configura e ouve automaticamente o serviço de notificação para novos arquivos e pode dimensionar até milhões de arquivos por segundo.
- **Fluxo unificado e processamento em batch:** **Delta Lake** é uma abordagem aberta para trazer o gerenciamento e a governança de dados a data lakes que aproveita o poder da computação distribuída do Apache Spark™ para enormes volumes de dados e metadados. O Databricks Delta Engine é um motor altamente otimizado que pode processar milhões de registros por segundo.
- **Governança de dados:** com o Controle de Acesso à Tabela Databricks (**AWS** | **Azure** | **GCP**), os administradores podem conceder diferentes níveis de acesso às tabelas Delta com base na função de negócios de um usuário.

- **Ferramentas de análise de segurança: o Databricks SQL** ajuda a criar um painel interativo com alertas automáticos quando padrões incomuns são detectados. Da mesma forma, ele pode ser facilmente integrado a ferramentas de BI amplamente adotadas, como Tableau, Microsoft Power BI e Looker.
- **Colaboração em notebooks Databricks: os notebooks colaborativos** Databricks permitem que as equipes de segurança colaborem em tempo real. Vários usuários podem executar consultas em vários idiomas, compartilhar visualizações e fazer comentários dentro do mesmo espaço de trabalho para manter as investigações avançando sem interrupção.

## Arquitetura de Lakehouse para dados da CrowdStrike Falcon

Recomendamos a seguinte arquitetura de lakehouse para cargas de trabalho de segurança cibernética, como os dados da CrowdStrike Falcon. O Auto Loader e o Delta Lake simplificam o processo de leitura de dados brutos do armazenamento e escrita em nuvem para uma tabela Delta a baixo custo e com o mínimo de trabalho DevOps.

Nessa arquitetura, os dados semiestruturados da CrowdStrike são carregados no armazenamento em nuvem do cliente na zona de destino. Em seguida, o Auto Loader usa serviços de notificação em nuvem para acionar automaticamente o processamento e a ingestão de novos arquivos nas tabelas Bronze do cliente, que funcionarão como a única fonte de verdade para todos os downstream jobs. O Auto Loader rastreará arquivos processados e não processados usando pontos de verificação para evitar o processamento de dados duplicados.

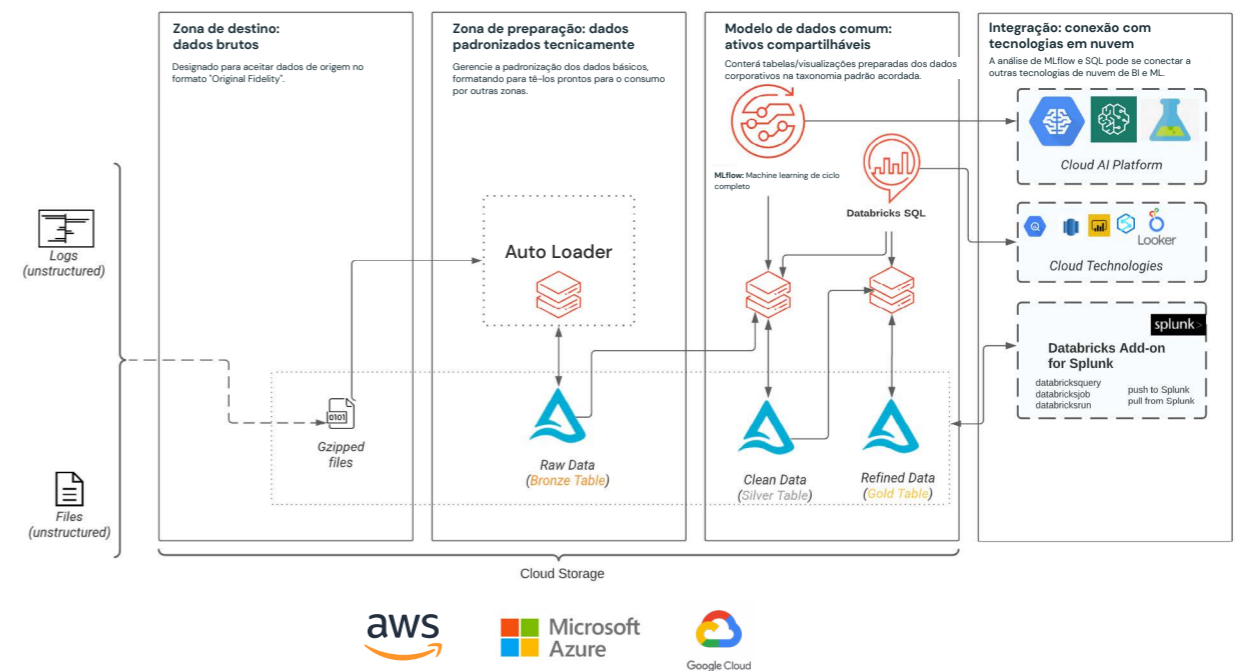


Figura 1

Arquitetura da lakehouse para dados da CrowdStrike Falcon

À medida que passamos do estágio Bronze para o Silver, o esquema será adicionado para fornecer estrutura aos dados. Como estamos lendo de uma única fonte da verdade, podemos processar todos os diferentes tipos de eventos e aplicar o esquema correto à medida que são escritos em suas respectivas tabelas. A capacidade de impor esquemas na camada Silver fornece uma base sólida para construir cargas de trabalho de ML e analíticas.

O estágio Gold, que agrega dados para consultas mais rápidas e desempenho em painéis e ferramentas de BI, é opcional, dependendo do caso de uso e dos volumes de dados. Os alertas podem ser definidos para disparar quando tendências inesperadas são observadas.



Outro recurso opcional é o **complemento Databricks para Splunk**, que permite que as equipes de segurança aproveitem o modelo econômico Databricks e o poder da IA sem ter que deixar o conforto do Splunk. Os clientes podem executar consultas ad hoc contra Databricks a partir de um painel Splunk ou barra de pesquisa com o complemento. Os usuários também podem iniciar notebooks ou trabalhos na Databricks por meio de um painel Splunk ou em resposta a uma pesquisa Splunk. A integração Databricks é bidirecional, permitindo que os clientes resumam dados ruidosos ou executem detecções em Databricks que aparecem no Splunk Enterprise Security. Os clientes podem até executar pesquisas Splunk a partir de um notebook Databricks para evitar a necessidade de duplicar dados.

A integração Splunk e Databricks permite que os clientes reduzam os custos, expandam as fontes de dados que analisam e forneçam os resultados de um mecanismo de análise mais robusto, sem alterar as ferramentas usadas pela equipe diariamente.

## Passo a passo do código

Como o Auto Loader abstrai a parte mais complexa da ingestão de dados baseada em arquivo, um pipeline de ingestão raw-to-Bronze pode ser criado dentro de algumas linhas de código. Abaixo, encontra-se um exemplo de código Scala para um pipeline de ingestão Delta. Os registros de eventos do CrowdStrike Falcon têm um nome de campo comum: "event\_simpleName".

```
val crowdstrikeStream = spark.readStream
  .format("cloudFiles")
  .option("cloudFiles.format", "text") // o arquivo de texto não precisa de esquema
  .option("cloudFiles.region", "us-west-2")
  .option("cloudFiles.useNotifications", "true")
  .load(rawDataSource)
  .withColumn("load_timestamp", current_timestamp())
  .withColumn("load_date", to_date($"load_timestamp"))
  .withColumn("eventType", from_json($"value", "struct", Map.empty[String, String]))
  .selectExpr("eventType.event_simpleName", "load_date", "load_timestamp", "value")
  .writeStream
  .format("delta")
  .option("checkpointLocation", checkpointLocation)
  .table("demo_bronze.crowdstrike")
```

Na camada raw-to-Bronze, apenas o nome do evento é extraído dos dados brutos. Ao adicionar um timestamp de carregamento e colunas de data, os usuários armazenam os dados brutos na tabela Bronze. A tabela Bronze é particionada por nome de evento e data de carregamento, o que ajuda a tornar os trabalhos Bronze-to-Silver mais eficientes, especialmente quando há interesse por um número limitado de intervalos de datas de eventos. Em seguida, um trabalho de streaming Bronze-to-Silver lê eventos de uma tabela Bronze, impõe um esquema e escreve em centenas de tabelas de eventos com base no nome do evento. Veja abaixo um exemplo de código Scala:

```
spark
  .readStream
  .option("ignoreChanges", "true")
  .option("maxBytesPerTrigger", "2g")
  .option("maxFilesPerTrigger", "64")
  .format("delta")
  .load(bronzeTableLocation)
  .filter($"event_simpleName" === "event_name")
  .withColumn("event", from_json(US$"value", schema_of_json(sampleJson)) )
  .select($"event.*", $"load_timestamp", $"load_date")
  .withColumn("silver_timestamp", current_timestamp())
  .writeStream
  .format("delta")
  .outputMode("append")
  .option("mergeSchema", "true")
  .option("checkpointLocation", checkpoint)
  .option("path", tableLocation)
  .start()
```

Cada esquema de evento pode ser armazenado em um registro de esquema ou em uma tabela Delta, caso um esquema precise ser compartilhado em vários serviços orientados por dados. Observe que o código acima usa uma amostra de string JSON lida da tabela Bronze, e o esquema é inferido a partir do JSON usando `schema_of_json()`. Posteriormente, a string JSON é convertida em uma estrutura usando `from_json()`. Em seguida, a estrutura é achatada, exigindo a adição de uma coluna de timestamp. Essas etapas fornecem um DataFrame com todas as colunas necessárias para serem anexadas a uma tabela de eventos. Por fim, escrevemos esses dados estruturados em uma tabela de eventos com o modo anexar.

Também é possível exibir eventos em várias tabelas com um stream com o `foreachBatch` definindo uma função que lidará com microbatches. Usando o `foreachBatch()`, é possível reutilizar fontes de dados em batches existentes para filtrar e escrever em várias tabelas. No entanto, o `foreachBatch()` oferece apenas garantias de escrita no mínimo uma vez. Portanto, uma implementação manual é necessária para aplicar semântica exatamente uma vez.

Neste estágio, os dados estruturados podem ser consultados com qualquer uma das linguagens suportadas em notebooks Databricks e jobs: Python, R, Scala e SQL. Os dados da camada Silver são convenientes de utilizar para análise de ML e ataque cibernético.

O próximo pipeline de streaming seria Silver-to-Gold. Nesse estágio, é possível agregar dados para criação de painéis e alertas. Na segunda parte desta série do blog, forneceremos mais algumas informações sobre como construímos painéis usando o Databricks SQL.

## O que vem por aí

Fique ligado em mais postagens de blog que geram ainda mais valor neste caso de uso aplicando ML e usando Databricks SQL.

Você pode usar esses [notebooks](#) em sua própria implantação da Databricks. Cada seção dos notebooks tem comentários. Convidamos você a nos enviar um e-mail para [cybersecurity@databricks.com](mailto:cybersecurity@databricks.com). Aguardamos suas perguntas e sugestões para tornar esse notebook mais fácil de entender e implantar.



Comece a experimentar esses **notebooks Databricks gratuitos**.

## SEÇÃO 2.3 **Desbloqueando o poder dos dados de saúde com um Lakehouse de dados moderno**

de MICHAEL ORTEGA, MICHAEL SANKY e AMIR KERMANY

19 de julho DE 2021

### Como superar os desafios de data warehouses e data lakes nos setores de saúde e life sciences

Um único paciente produz aproximadamente **80 megabytes de dados médicos** por ano. Multiplique isso por milhares de pacientes ao longo da vida e você estará olhando para petabytes de dados de pacientes que contêm insights valiosos. Desbloquear esses insights pode ajudar a simplificar as operações clínicas, acelerar a P&D de medicamentos e melhorar os resultados de saúde do paciente. Mas, em primeiro lugar, os dados precisam estar preparados para funções analíticas e IA downstream. Infelizmente, a maioria das organizações de saúde e life sciences gastam um tempo enorme simplesmente coletando, limpando e estruturando seus dados.

Um único paciente produz mais de 80 megabytes de dados médicos todos os anos

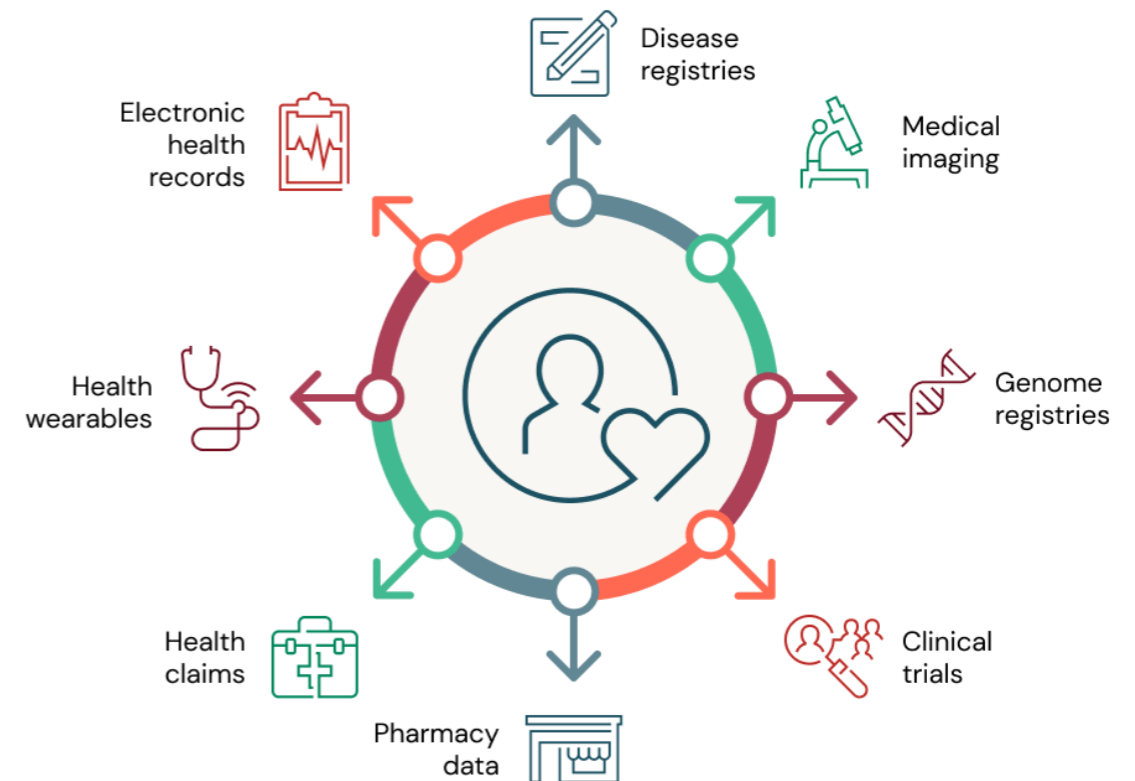


Figura 1

Os dados de saúde estão crescendo exponencialmente, com um único paciente produzindo mais de 80 megabytes de dados por ano

## Desafios com análises de dados na área da saúde e life sciences

Há muitos motivos pelos quais a preparação de dados, a análise e a IA são desafios para as organizações do setor de saúde, mas muitos estão relacionados a investimentos em arquiteturas de dados legadas construídas em data warehouses. Aqui estão os quatro desafios mais comuns que vemos no setor:

### DESAFIO N.º 1: VOLUME

#### Dimensionamento de dados de saúde em rápido crescimento

Talvez a genômica seja o melhor exemplo do enorme crescimento do volume de dados na área da saúde. O primeiro genoma custou mais de US\$ 1 bilhão para ser sequenciado. Devido aos custos proibitivos, os primeiros esforços (e muitos ainda persistem) focaram na genotipagem, um processo para procurar variantes específicas em uma fração muito pequena do genoma de uma pessoa, geralmente em torno de 0,1%. Isso evoluiu para Sequenciamento do exome inteiro, que cobre as porções de codificação de proteínas do genoma, ainda menos de 2% de todo o genoma. As empresas agora oferecem testes diretos ao consumidor para sequenciamento de genomas inteiros (WGS) que são inferiores a US\$ 300 por 30x WGS. Em um nível populacional, o UK Biobank está lançando mais de 200.000 genomas inteiros para pesquisa este ano. Não é apenas genômica. Imagens, dispositivos de saúde e registros médicos eletrônicos também estão crescendo muito.

O dimensionamento é essencial para iniciativas como análises de saúde da população e descoberta de medicamentos. Infelizmente, muitas arquiteturas legadas são construídas no local e projetadas para capacidade máxima. Essa abordagem resulta em poder de compute não utilizado (e, em última análise, dinheiro desperdiçado) durante períodos de baixo uso e não é dimensionada rapidamente quando são necessárias atualizações.

### DESAFIO N.º 2: VARIEDADE

#### Análise de diversos dados de saúde

As organizações de saúde e life sciences lidam com uma enorme variedade de dados, cada uma com suas próprias nuances. É amplamente aceito que mais de 80% dos dados médicos não são estruturados, mas a maioria das organizações ainda concentra sua atenção em data warehouses projetados para dados estruturados e funções analíticas tradicionais baseadas em SQL. Os dados não estruturados incluem dados de imagem, o que é fundamental para diagnosticar e medir a progressão da doença em áreas como oncologia, imunologia e neurologia (as áreas de custo de crescimento mais rápido) e texto narrativo em notas clínicas, que são fundamentais para entender a saúde completa do paciente e a história social. Ignorar esses tipos de dados, ou colocá-los de lado, não é uma opção.

Para complicar ainda mais, o ecossistema de serviços de saúde está se tornando mais interconectado, exigindo que as partes interessadas se familiarizem com novos tipos de dados. Por exemplo, os provedores precisam de dados de sinistros para gerenciar e decidir acordos de compartilhamento de risco, e os pagadores precisam de dados clínicos para dar suporte a processos como autorizações anteriores e para impulsionar medidas de qualidade. Muitas vezes, essas organizações não possuem arquiteturas de dados e plataformas para dar suporte a esses novos tipos de dados.

Algumas organizações investiram em data lakes para dar suporte a dados não estruturados e funções analíticas avançadas, mas isso cria um novo conjunto de problemas. Nesse ambiente, as equipes de dados precisam gerenciar dois sistemas – data warehouses e data lakes – onde os dados são copiados em ferramentas isoladas, resultando em problemas de qualidade de dados e de gerenciamento.

### DESAFIO N.º 3: VELOCIDADE

#### Processamento de dados de streaming para insights do paciente em tempo real

Em muitos cenários, a saúde é uma questão de vida ou morte. As condições podem ser muito dinâmicas, e o processamento de dados em batch — mesmo feito diariamente — geralmente não é bom o suficiente. O acesso às informações mais recentes e atualizadas é essencial para um atendimento intervencionista bem-sucedido. Para salvar vidas, os dados de streaming são usados por hospitais e sistemas de saúde nacionais para tudo, desde a previsão de sepse até a implementação de previsão de demanda em tempo real para camas de UTI.

Além disso, a velocidade dos dados é um componente importante da revolução digital da área da saúde. Os indivíduos têm acesso a mais informações do que nunca e podem influenciar seus cuidados em tempo real. Por exemplo, dispositivos vestíveis — como os monitores contínuos de glicose fornecidos pela [Livongo](#) — transmitem dados em tempo real para aplicativos móveis que fornecem recomendações comportamentais personalizadas.

Apesar de alguns desses sucessos iniciais, a maioria das organizações não projetou sua arquitetura de dados para acomodar a velocidade de transmissão de dados. Problemas de confiabilidade e desafios na integração de dados em tempo real com dados históricos estão inibindo a inovação.

### DESAFIO N.º 4: VERACIDADE

#### Desenvolvimento de confiança nos dados de saúde e na IA

Por último, mas não menos importante, os padrões clínicos e regulatórios exigem o nível mais alto de precisão de dados na área da saúde. As organizações de saúde têm altas exigências de conformidade com a saúde pública que devem ser cumpridas. A democratização de dados nas organizações exige governança.

Além disso, as organizações precisam de uma boa governança de modelo ao trazer inteligência artificial (IA) e machine learning (ML) para um ambiente clínico. Infelizmente, a maioria das organizações tem plataformas separadas para fluxos de trabalho de ciência de dados desconectados de seu data warehouse. Isso cria sérios desafios ao tentar construir confiança e reprodutibilidade em aplicativos baseados em IA.

### Desbloquear dados de saúde com um lakehouse

A [arquitetura lakehouse](#) ajuda as organizações de saúde e life sciences a superarem esses desafios com uma arquitetura de dados moderna que combina baixo custo, escalabilidade e flexibilidade de um data lake em nuvem com o desempenho e governança de um data warehouse. Com um lakehouse, as organizações podem armazenar todos os tipos de dados e potencializar todos os tipos de funções analíticas e ML em um ambiente aberto.



## Construir um lakehouse para a área da saúde e life sciences

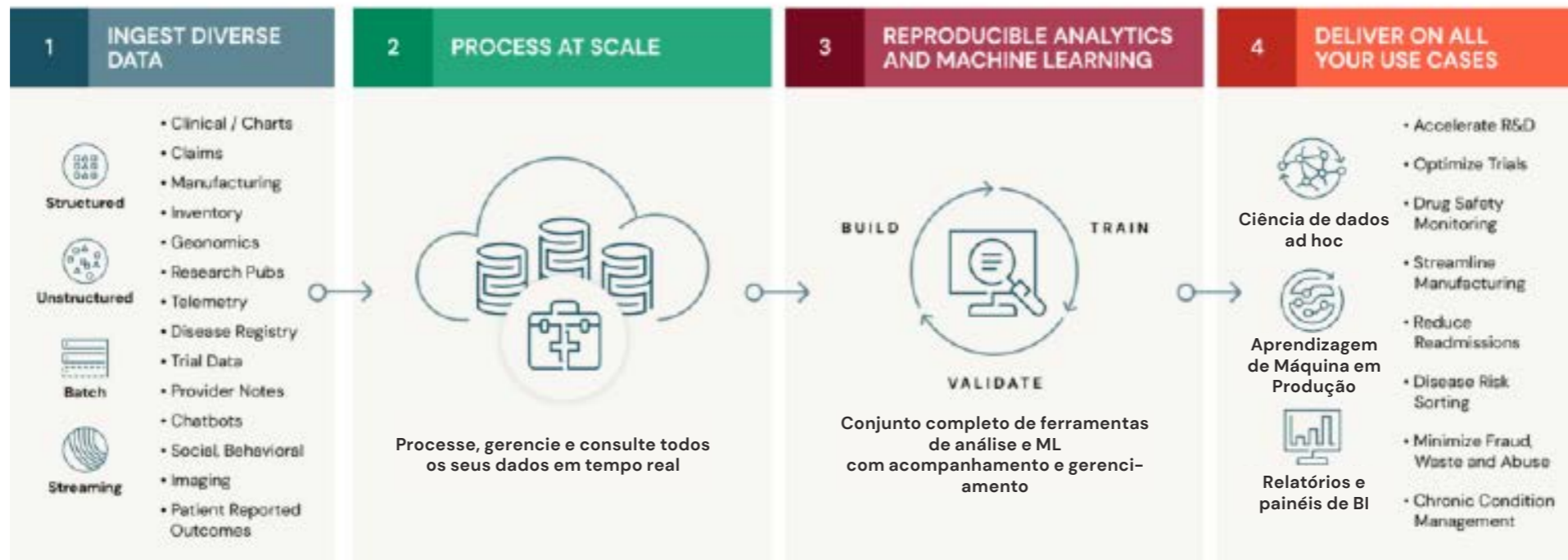


Figura 2

Ofereça todos os seus casos de uso de análise de dados de saúde e life sciences com uma arquitetura moderna de lakehouse

Especificamente, o lakehouse oferece os seguintes benefícios para as organizações de saúde e life sciences:

- **Organize todos os seus dados de saúde em escala.** No centro da plataforma Databricks Lakehouse está o **Delta Lake**, uma camada de gerenciamento de dados de código aberto que fornece confiabilidade e desempenho ao seu data lake. Diferentemente de um data warehouse tradicional, o Delta Lake oferece suporte a todos os tipos de dados estruturados e não estruturados, e para facilitar a ingestão de dados de saúde, o Databricks criou conectores para tipos de dados específicos do domínio, como prontuários médicos eletrônicos e genômica. Esses conectores vêm com modelos de dados padrão do setor em um conjunto de aceleradores de soluções de início rápido. Além disso, o Delta Lake oferece otimizações incorporadas para armazenamento em cache de

dados e indexação para acelerar significativamente as velocidades de processamento de dados. Com esses recursos, as equipes podem obter todos os dados brutos em um único lugar e depois organizá-los para criar uma visão holística da saúde do paciente.

- **Potencialize todas as análises de pacientes e a IA.** Com todos os dados centralizados em um lakehouse, as equipes podem criar análises poderosas de pacientes e modelos preditivos diretamente nos dados. Para aproveitar esses recursos, a Databricks fornece espaços de trabalho colaborativos com um conjunto completo de ferramentas de análise e IA e suporte para um amplo conjunto de linguagens de programação — como SQL, R, Python e Scala. Isso permite que um grupo diversificado de usuários, como cientistas de dados, engenheiros e especialistas em informações clínicas, trabalhem juntos para analisar, modelar e visualizar todos os seus dados de saúde.

- **Forneça insights em tempo real sobre o paciente.** O lakehouse oferece uma arquitetura unificada para streaming e dados em batch. Não é necessário dar suporte a duas arquiteturas diferentes nem enfrentar problemas de confiabilidade. Além disso, ao executar a arquitetura do lakehouse na Databricks, as organizações têm acesso a uma plataforma nativa da nuvem que cresce automaticamente com base na carga de trabalho. Isso facilita a ingestão de dados de streaming e a combinação com petabytes de dados históricos para insights quase em tempo real em escala populacional.
- **Forneça qualidade de dados e conformidade.** Para tratar da veracidade dos dados, o lakehouse inclui recursos que não existem nos data lakes tradicionais, como aplicação de esquemas, auditoria, versionamento e controles de acesso com granulometria fina. Um benefício importante do lakehouse é a capacidade de realizar tanto a análise quanto o ML nesta mesma fonte de dados confiável. Além disso, a Databricks fornece recursos de rastreamento e gerenciamento do modelo ML para facilitar a reprodução dos resultados pelas equipes em todos os ambientes e ajudar a cumprir os padrões de conformidade. Todos esses recursos são fornecidos em um ambiente analítico em conformidade com a HIPAA (Lei de Portabilidade e Responsabilidade de Seguro Saúde).

Esse lakehouse é a melhor arquitetura para gerenciar dados de saúde e life sciences. Ao unir essa arquitetura aos recursos da Databricks, as organizações podem oferecer suporte a uma ampla variedade de casos de uso altamente impactantes, desde a descoberta de medicamentos até programas de gerenciamento crônico de doenças.

## Comece a construir seu lakehouse para a saúde e life sciences

Conforme mencionado acima, temos o prazer de disponibilizar uma série de Aceleradores de soluções para ajudar as organizações de saúde e life sciences a começar a construir um lakehouse para suas necessidades específicas. Nossos Aceleradores de soluções incluem dados de amostra, código pré-criado e instruções passo a passo em um notebook Databricks.

### Acelerador de novas soluções: lakehouse para evidências no mundo real.

Os dados do mundo real fornecem às empresas farmacêuticas novos insights sobre a saúde do paciente e a eficácia do medicamento fora de um estudo. Esse acelerador ajuda você a construir um Lakehouse para evidências no mundo real na Databricks. Mostraremos como ingerir dados de exemplo de EHR (prontuário eletrônico) para uma população de pacientes, estruturar os dados usando o modelo de dados comuns OMOP e, em seguida, executar análises em escala para desafios como investigar padrões de prescrição de medicamentos.



**Comece a experimentar esses notebooks Databricks gratuitos.**

**Saiba mais sobre todas as nossas soluções de Saúde e Life Sciences.**

## SEÇÃO 2.4 **Prontidão e confiabilidade na transmissão de relatórios regulatórios**

de ANTOINE AMEND e FAHMID KABIR

17 de setembro de 2021

Gerenciar riscos e compliance regulatório é um esforço cada vez mais complexo e dispendioso. A mudança regulatória aumentou 500% desde a crise financeira global de 2008 e impulsionou os custos regulatórios no processo. Dadas as multas associadas à não conformidade e às violações de SLA (os bancos atingiram um recorde em multas de US\$ 10 bilhões em 2019 para o AML), os relatórios de processamento têm que prosseguir mesmo que os dados sejam incompletos. Por outro lado, um histórico de má qualidade dos dados também é multado por causa de "controles insuficientes". Como consequência, muitas instituições de serviços financeiros (FSIs) frequentemente ficam lutando entre baixa qualidade de dados e SLAs rigorosos, se equilibrando entre confiabilidade e prontidão de dados.

Neste Acelerador de soluções de relatórios regulatórios, demonstramos como o **Delta Live Tables** pode garantir a aquisição e o processamento de dados regulatórios em tempo real para acomodar SLAs regulatórios. Com o **Delta Sharing** e o Delta Live Tables combinados, os analistas ganham confiança em tempo real na qualidade dos dados regulatórios que estão sendo transmitidos. Neste post do blog, demonstramos os benefícios da arquitetura de lakehouse para combinar modelos de dados do setor de serviços financeiros com a flexibilidade da computação em nuvem para permitir altos padrões de governança com baixa sobrecarga de desenvolvimento. Agora, vamos explicar o que é um modelo de dados FIRE e como o Delta Live Tables pode ser integrado para criar pipelines de dados robustos.

### Modelo de dados FIRE

A norma de dados regulatórios financeiros (FIRE) define uma especificação comum para a transmissão de dados granulares entre sistemas regulatórios em finanças. Os dados regulatórios referem-se a dados que fundamentam submissões regulatórias, requisitos e cálculos e são utilizados para fins de política, monitoramento e supervisão. O **padrão de dados FIRE** é apoiado pela **Comissão Europeia**, o **Open Data Institute** e a **Incubadora de Open Data FIRE** padrão de dados para a Europa através do programa de financiamento Horizon 2020. Como parte desta solução, contribuimos com um módulo PySpark que pode interpretar modelos de dados FIRE em pipelines operacionais Apache Spark™.





## Delta Live Tables

Recentemente, a Databricks anunciou um novo produto para orquestração de pipelines de dados, o Delta Live Tables, que facilita a criação e o gerenciamento de pipelines de dados confiáveis em escala empresarial. Com a capacidade de avaliar várias expectativas, descartar ou monitorar registros inválidos em tempo real, os benefícios de integrar o modelo de dados FIRE ao Delta Live Tables são óbvios. Conforme ilustrado na arquitetura a seguir, o Delta Live Tables **compreenderá** os dados regulatórios granulares sobre armazenamento em nuvem, **esquematará** o conteúdo e **validará** os registros para consistência de acordo com a especificação de dados FIRE. Continue lendo para nos ver demonstrando o uso do Delta Sharing para trocar informações granulares entre sistemas regulatórios de maneira segura, escalável e transparente.

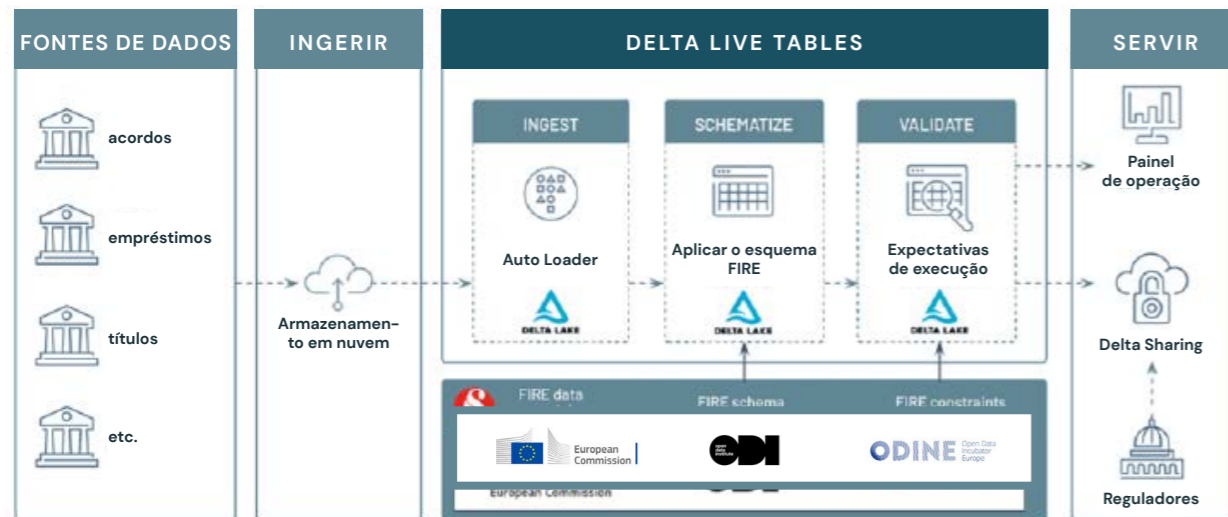


Figura 1

## Aplicação do esquema

Embora alguns formatos de dados possam parecer estruturados (por exemplo, arquivos JSON), a aplicação de um esquema não é apenas uma boa prática de engenharia; em ambientes corporativos, e especialmente no espaço de conformidade regulamentar, a aplicação de esquemas garante que qualquer campo ausente seja esperado, campos inesperados sejam descartados, e tipos de dados sejam totalmente avaliados (por exemplo, uma data deve ser tratada como um objeto de data e não como uma string). Ele também testa seus sistemas para eventuais drifts de dados. Usando o módulo FIRE PySpark, recuperamos programaticamente o esquema Spark necessário para processar uma determinada entidade FIRE (entidade colateral nesse exemplo) que aplicamos em um stream de registros brutos.

```
from fire.spark import FireModel
fire_model = FireModel().load("collateral")
fire_schema = fire_model.esquema
```

No exemplo abaixo, aplicamos o esquema aos arquivos CSV recebidos. Ao decorar este processo usando a anotação `@dlt`, definimos nosso ponto de entrada para nossa Delta Live Table, lendo arquivos CSV brutos de um diretório montado e escrevendo registros esquematizados para uma camada Bronze.

```
@dlt.create_table()
def collateral_bronze():
    return (
        spark
        .readStream
        .option("maxFilesPerTrigger", "1")
        .option("badRecordsPath", "/path/to/invalid/collateral")
        .format("csv")
        .esquema(fire_schema)
        .load("/path/to/raw/collateral")
    )
```

## Avaliar expectativas

Aplicar um esquema é uma coisa, aplicar suas restrições é outra. Dada a **definição de esquema** de uma entidade FIRE (consulte o exemplo da definição de esquema colateral), podemos detectar se um campo é ou não exigido. Dado um objeto de enumeração, garantimos que seus valores sejam consistentes (por exemplo, código de moeda). Além das restrições técnicas do esquema, o modelo FIRE também relata expectativas de negócios, como mínimo, máximo, monetário e maxltens. Todas essas restrições técnicas e comerciais serão recuperadas programaticamente do modelo de dados FIRE e interpretadas como uma série de expressões Spark SQL.

```
from fire.spark import FireModel
fire_model = FireModel().load("collateral")
fire_constraints = fire_model.constraints
```

Com o Delta Live Tables, os usuários têm a capacidade de avaliar múltiplas expectativas ao mesmo tempo, permitindo que abandonem registros inválidos, simplesmente monitorem a qualidade dos dados ou abortem um pipeline inteiro. Em nosso cenário específico, queremos abandonar registros que não atendem a qualquer uma de nossas expectativas, que mais tarde armazenamos em uma tabela de quarentena, conforme relatado nos notebooks fornecidos neste blog.

```
@dlt.create_table()
@dlt.expect_all_or_drop(fire_constraints)
def collateral_silver():
    return dlt.read_stream("collateral_bronze")
```

Com apenas algumas linhas de código, garantimos que nossa tabela Silver seja sintática (esquema válido) e semanticamente (expectativas válidas) correta. Conforme mostrado abaixo, os executivos de conformidade têm visibilidade total do número de registros sendo processados em tempo real. Neste exemplo específico, garantimos que nossa entidade colateral fosse exatamente 92,2% completa (a quarentena lida com os 7,8% restantes).

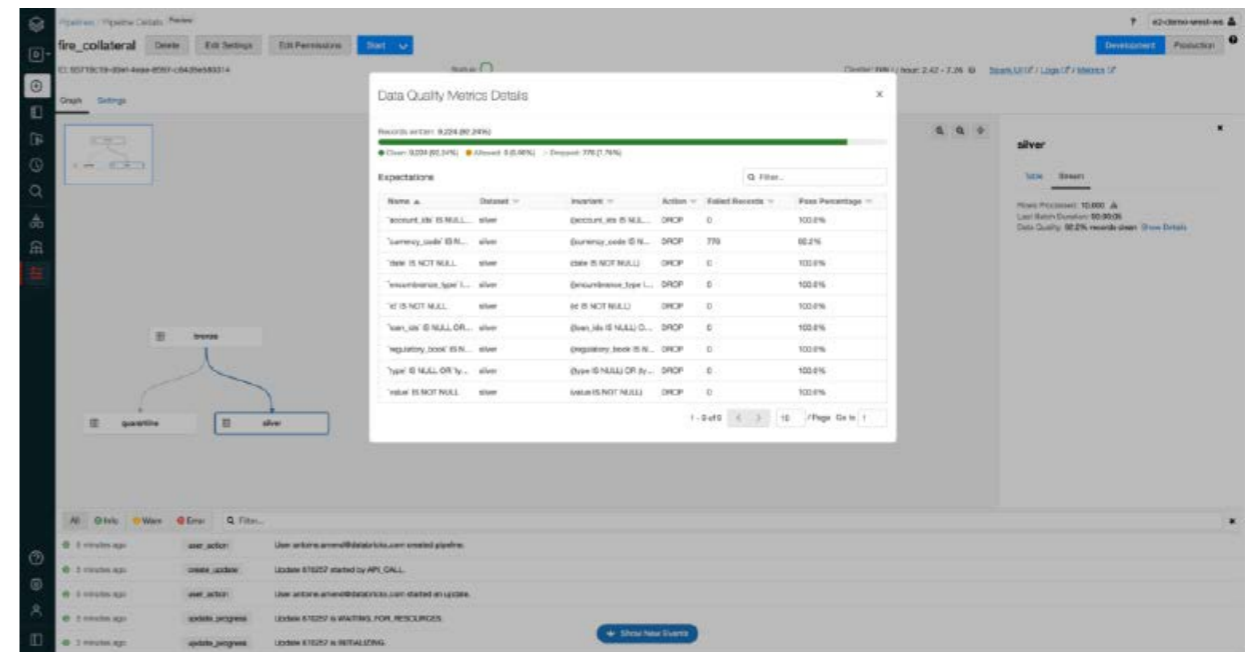


Figura 2

## Armazenamento de Dados de Operações

Além dos dados reais armazenados como arquivos Delta, as Delta Live Tables armazenam métricas de operação como formato "delta" em sistema/eventos. Seguimos um padrão da arquitetura de lakehouse "assinando" novas métricas operacionais usando o Auto Loader, processando eventos do sistema à medida que novas métricas se desdobram — em batch ou em tempo real. Graças ao log de transação do Delta Lake que mantém o controle de qualquer atualização de dados, as organizações podem acessar novas métricas sem precisar construir e manter seu próprio processo de checkpointing.

```
input_stream = spark \
    .readStream \
    .format("delta") \
    .load("/path/to/pipeline/system/events")

output_stream = extract_metrics(input_stream)

output_stream \
    .writeStream \
    .format("delta") \
    .option("checkpointLocation", "/path/to/checkpoint") \
    .table(metrics_table)
```

Com todas as métricas disponíveis centralmente em um armazenamento de operações, os analistas podem usar o [Databricks SQL](#) para criar recursos simples de painel ou mecanismos de alerta mais complexos para detectar problemas de qualidade de dados em tempo real.

	entity	expectation_name	expectation_value
1	adjustment	[id] is mandatory	`id` IS NOT NULL
2	adjustment	[date] is mandatory	`date` IS NOT NULL
3	adjustment	[col] is mandatory	`col` IS NOT NULL
4	adjustment	[contribution_amount] is mandatory	`contribution_amount` IS NOT NULL
5	adjustment	[currency_code] is mandatory	`currency_code` IS NOT NULL
6	adjustment	[currency_code] not allowed value	(`currency_code` IS NULL) OR (`currency_code` IN ('AED', 'AFN', 'ALL', 'AMD', 'ANG', 'AOA', 'ARS', 'AUD', 'AWG', 'AZN', 'BAM', 'BBD', 'BDT', 'BGN', 'BHD', 'BIF', 'BMD', 'BND', 'BOB', 'BOV', 'BRL', 'BSD', 'BTN', 'BWP', 'BYN', 'BZD', 'CAD', 'CDF', 'CHE', 'CHF', 'CHW', 'CLF', 'CLP', 'CNY', 'COP', 'COU', 'CRC', 'CUC', 'CUP', 'CVE', 'CZK', 'DJF', 'DKK', 'DOP', 'DZD', 'EGP', 'ERN', 'ETB', 'EUR', 'FJD', 'FKP', 'GBP', 'GEL', 'GHS', 'GIP', 'GMD', 'GNF', 'GTQ', 'GYD', 'HKD', 'HNL', 'HRK', 'HTG', 'HUF', 'L...'

O aspecto imutabilidade do formato Delta Lake, juntamente com a transparência na qualidade dos dados oferecida pelo Delta Live Tables, permite que as instituições financeiras "viajem no tempo" para versões específicas de seus dados que correspondam tanto ao volume quanto à qualidade necessárias para a conformidade normativa. Em nosso exemplo específico, reproduzir nossos 7,8% dos registros inválidos armazenados em quarentena resultará em uma versão Delta diferente anexada à nossa tabela Silver, uma versão que pode ser compartilhada entre os órgãos reguladores.

```
DESCRIBE HISTORY fire.collateral_silver
```

## Transmissão de dados regulatórios

Com total confiança na qualidade e no volume de dados, as instituições financeiras podem trocar informações com segurança entre sistemas regulatórios usando o **Delta Sharing**, um protocolo aberto para troca de dados corporativos. Não restringindo os usuários finais à mesma plataforma, nem contando com pipelines ETL complexos para consumir dados (acessando arquivos de dados por meio de um servidor SFTP, por exemplo), a natureza de código aberto do Delta Lake torna possível que os consumidores de dados acessem dados esquematizados nativamente a partir do Python, Spark ou diretamente por meio de painéis de IM/BI (como Tableau ou Power BI).

Embora possamos estar compartilhando nossa tabela Silver como está, podemos querer usar regras de negócios que só compartilham dados regulatórios quando um threshold de qualidade de dados predefinido é cumprido. Neste exemplo, clonamos nossa tabela Silver em uma versão diferente e em um local específico separado de nossas redes internas e acessível por usuários finais (zona desmilitarizada ou DMZ).

```
from delta.tables import *

deltaTable = DeltaTable.forName(spark, "fire.collateral_silver")
deltaTable.cloneAtVersion(
    approved_version,
    dmz_path,
    isShallow=False,
    replace=True
)

spark.sql(
    "CREATE TABLE fire.collateral_gold USING DELTA LOCATION '{}'"
    .format(dmz_path)
)
```

Embora a solução de código aberto Delta Sharing dependa de um servidor de compartilhamento para gerenciar permissões, a Databricks utiliza o **Unity Catalog** para centralizar e aplicar políticas de controle de acesso, fornecer aos usuários recursos completos de logs de auditoria e simplificar o gerenciamento de acesso por meio de sua interface SQL. No exemplo abaixo, criamos um SHARE que inclui nossas tabelas regulatórias e um RECIPIENT para compartilhar nossos dados.

```
-- DEFINIR NOSSA ESTRATÉGIA DE COMPARTILHAMENTO
CREATE SHARE regulatory_reports;

ALTER SHARE regulatory_reports ADD TABLE fire.collateral_gold;
ALTER SHARE regulatory_reports ADD TABLE fire.loan_gold;
ALTER SHARE regulatory_reports ADD TABLE fire.security_gold;
ALTER SHARE regulatory_reports ADD TABLE fire.derivative_gold;

-- CREATE RECIPIENTS AND GRANT SELECT ACCESS
CREATE RECIPIENT regulatory_body;


GRANT SELECT ON SHARE regulatory_reports TO RECIPIENT regulatory_body;
```

Qualquer regulador ou usuário com permissões concedidas pode acessar nossos dados subjacentes usando um token de acesso pessoal trocado por meio desse processo. Para obter mais informações sobre o Delta Sharing, visite nossa página do produto e entre em contato com seu representante Databricks.

## Ponha sua conformidade à prova

Através desta série de notebooks e Delta Live Tables jobs, demonstramos os benefícios da arquitetura do lakehouse na ingestão, no processamento, na validação e na transmissão de dados regulamentares. Especificamente, abordamos a necessidade das organizações de assegurar consistência, integridade e pontualidade de pipelines regulatórios que poderiam ser facilmente alcançados usando um modelo de dados comum (FIRE) acoplado a um motor de orquestração flexível (Delta Live Tables). Com as capacidades do Delta Sharing, finalmente demonstramos como as FSIs poderiam trazer total transparência e confiança aos dados regulatórios trocados entre vários sistemas regulatórios, ao mesmo tempo em que atendem às exigências de relatórios, reduzindo os custos de operação e adaptando-se a novos padrões.

Familiarize-se com o pipeline de dados FIRE usando os [notebooks](#) anexos e visite nosso [Hub de Acelerador de soluções](#) para se atualizar com nossas últimas soluções para serviços financeiros.



Comece a experimentar esses **notebooks Databricks gratuitos**.

## SEÇÃO 2.5 **Soluções AML em escala usando a plataforma Databricks Lakehouse**

de SRI GHATTAMANENI, RICARDO PORTILLA e ANINDITA MAHAPATRA

16 de julho DE 2021

### **Resolver os principais desafios para construir uma solução para crimes financeiros**

A conformidade com o AML (combate à lavagem de dinheiro) tem sido, sem dúvida, um dos principais itens da agenda para os reguladores que fornecem supervisão de instituições financeiras em todo o mundo. O AML evoluiu e se tornou mais sofisticado ao longo das décadas, assim como os requisitos regulatórios projetados para combater a lavagem de dinheiro moderna e os esquemas de financiamento terrorista. **A Bank Secrecy Act (Lei de Sigilo Bancário) de 1970** forneceu orientação e estrutura para instituições financeiras estabelecerem um controle adequado para monitorar transações financeiras e relatar atividades fiscais suspeitas às autoridades competentes. Essa lei proporcionou a base para como as instituições financeiras combatem a lavagem de dinheiro e o terrorismo financeiro.

### **Por que a lavagem de dinheiro é tão complexa**

As operações atuais de AML têm pouca semelhança com as da última década. A mudança para o setor bancário digital, com o processamento diário de bilhões de transações por instituições financeiras (IFs), resultou no crescente âmbito de lavagem de dinheiro, mesmo com sistemas mais rigorosos de

monitoramento de transações e soluções robustas Know Your Customer (KYC). Neste blog, compartilhamos nossas experiências ao trabalhar com nossos clientes de IF para desenvolver soluções de AML de escala empresarial na **plataforma lakehouse** que oferecem supervisão forte e soluções inovadoras e escaláveis para se adaptarem à realidade das ameaças modernas de lavagem de dinheiro online.

### **Construir uma solução AML com lakehouse**

A carga operacional do processamento de bilhões de transações por dia vem da necessidade de armazenar os dados de várias fontes e de soluções AML intensivas e de última geração. Essas soluções fornecem análises e relatórios de risco robustos, além de oferecer suporte ao uso de modelos avançados de machine learning para reduzir falsos positivos e melhorar a eficiência da investigação downstream. As IFs já tomaram medidas para resolver os problemas de infraestrutura e dimensionamento, migrando para a nuvem para melhor segurança, agilidade e as economias de escala necessárias para armazenar grandes quantidades de dados.

Mas existe a questão de como dar sentido à enorme quantidade de dados estruturados e não estruturados coletados e guardados em armazenamento de objetos baratos. Enquanto os fornecedores de nuvens oferecem uma maneira



barata de armazenar os dados, entender os dados para o gerenciamento de risco AML downstream e atividades de conformidade começa com o armazenamento dos dados em formatos de alta qualidade e desempenho para o consumo downstream. A **plataforma Databricks Lakehouse** faz exatamente isso. Combinando os benefícios de baixo custo de armazenamento dos data lakes com os robustos recursos de transação dos data warehouses, as IFs podem realmente construir a moderna plataforma AML.

Além dos desafios de armazenamento de dados descritos acima, os analistas de AML enfrentam alguns desafios importantes específicos do domínio:

- Melhore o time-to-value analisando dados não estruturados, como imagens, dados textuais e links de rede
- Reduza a carga de DevOps para suportar recursos críticos de ML, como resolução de entidades, visão computacional e análise de gráficos em metadados de entidades

- Elimine os silos introduzindo a engenharia de análise e uma camada de painel em transações de AML e tabelas enriquecidas

Felizmente, Databricks ajuda a resolver isso utilizando o **Delta Lake** para armazenar e combinar dados não estruturados e estruturados para criar relacionamentos entre entidades; além disso, o Databricks Delta Engine fornece acesso eficiente usando o novo **Photon compute** para acelerar as consultas de BI em tabelas. Além desses recursos, o ML é excelente no lakehouse, o que significa que analistas e cientistas de dados não perdem tempo fazendo uma subamostra ou movendo dados para compartilhar painéis e ficar um passo à frente dos malfeitores.

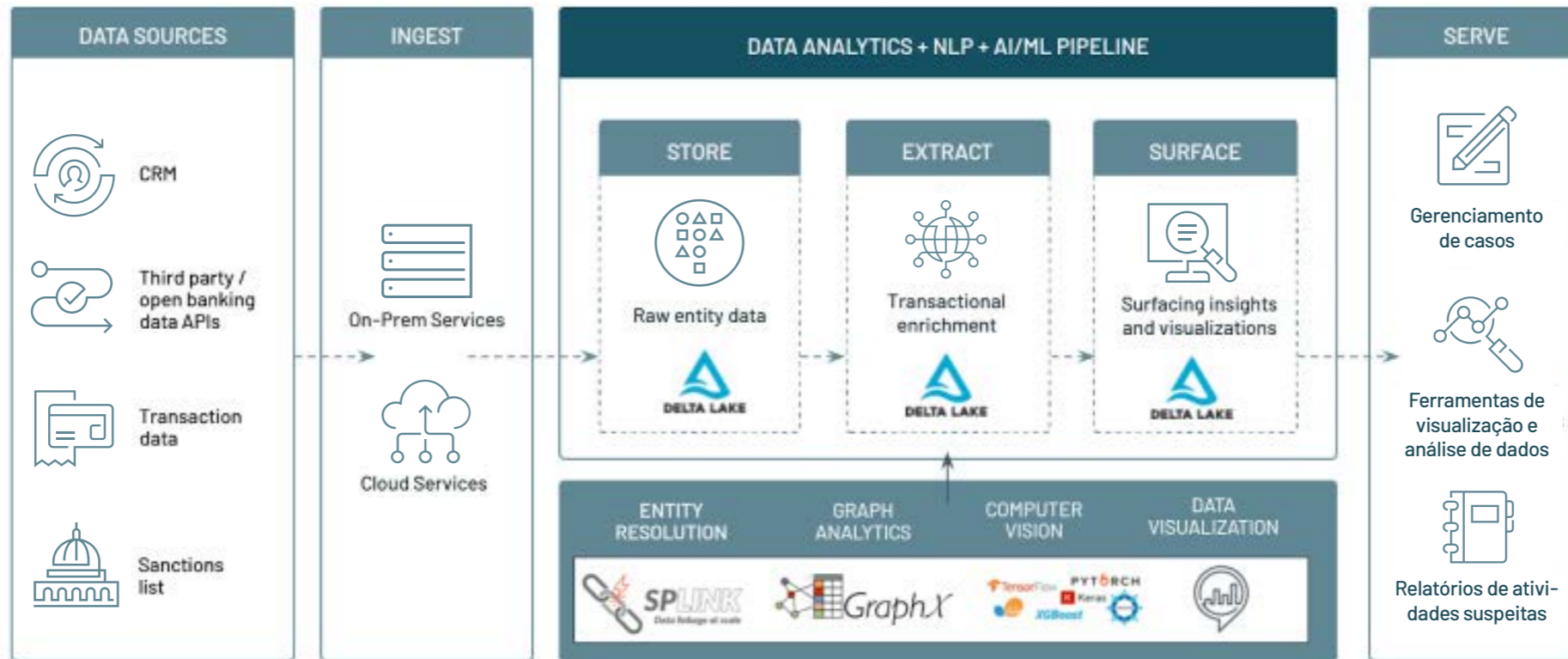


Figura 1

## Detectando padrões AML com recursos de grafos

Uma das principais fontes de dados que os analistas de AML usam como parte de um caso são os *dados de transação*. Embora esses dados sejam tabulares e facilmente acessíveis com SQL, é difícil rastrear cadeias de transações que consistem em três ou mais camadas profundas com consultas SQL. Por esse motivo, é importante ter um conjunto flexível de linguagens e APIs para expressar conceitos simples, como uma rede conectada de indivíduos suspeitos que realizam transações ilegalmente juntos. Felizmente, isso é simples de realizar usando GraphFrames, uma API de gráficos pré-instalada no [Databricks Runtime para Machine Learning](#).

Nesta seção, mostraremos como a análise de grafos pode ser usada para detectar esquemas AML, tais como identidade sintética e estratificação/estruturação de camadas. Vamos utilizar um conjunto de dados composto de transações, assim como entidades derivadas de transações, para detectar a presença destes padrões com Apache Spark™, GraphFrames e Delta Lake. Os padrões persistidos são salvos no Delta Lake, de modo que a [Databricks SQL](#) possa ser aplicada nas versões agregadas de nível Gold destas descobertas, oferecendo o poder da análise gráfica aos usuários finais.

## Cenário 1 – identidades sintéticas

Como mencionado acima, a existência de identidades sintéticas pode ser uma causa de alarme. Usando a análise de gráficos, todas as entidades de nossas transações podem ser analisadas em massa para detectar um nível de risco. Em nossa análise, isso é feito em três fases:

- Com base nos dados da transação, extraia as entidades
- Crie links entre entidades com base no endereço, número de telefone ou e-mail
- Use componentes conectados ao GraphFrames para determinar se várias entidades (identificadas por um ID e outros atributos acima) estão conectadas por meio de um ou mais links

Com base em quantas conexões (ou seja, atributos comuns) existem entre entidades, podemos atribuir uma pontuação de risco menor ou maior e criar um alerta com base em grupos de pontuação alta. Abaixo está uma representação básica desta ideia.

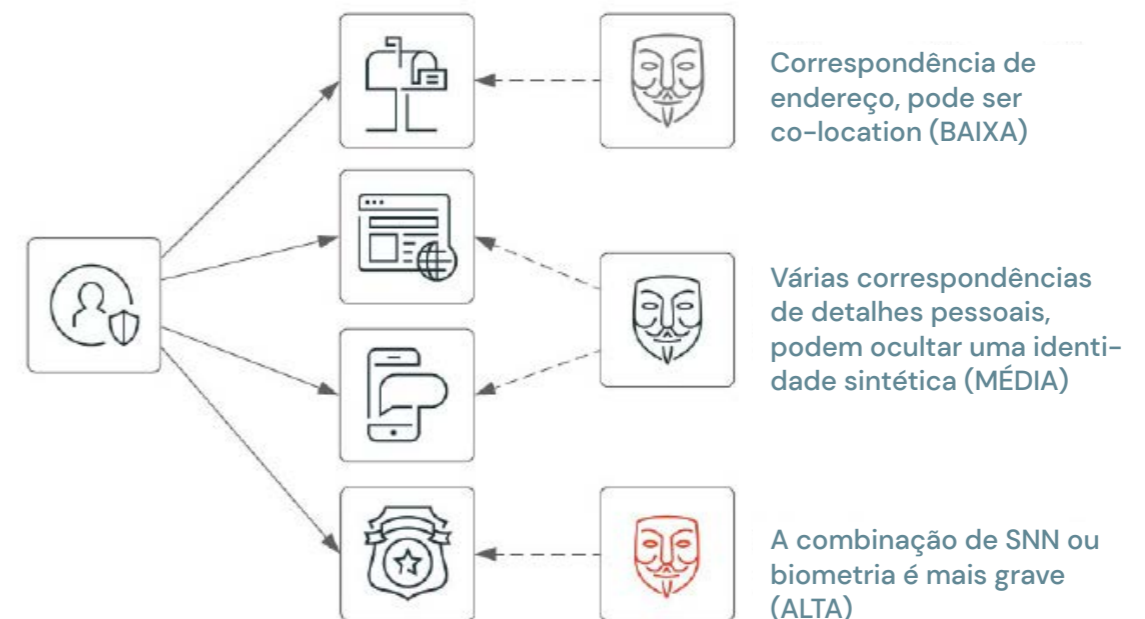


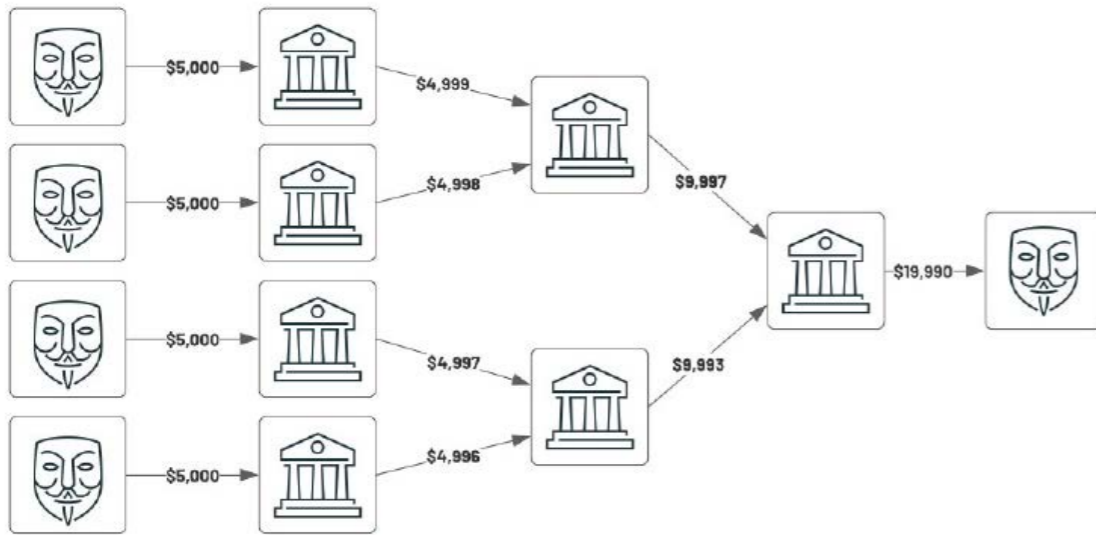
Figura 2





## Cenário 2 – estruturação

Outro padrão comum é chamado de *estruturação*, que ocorre quando várias entidades conspiram e enviam pagamentos menores "fora do radar" para um conjunto de bancos que, posteriormente, direcionam valores agregados maiores para uma instituição final (como descrito abaixo, na extrema direita). Nesse cenário, todas as partes permaneceram abaixo do valor limite de US\$ 10.000, o que normalmente alertaria as autoridades. Não apenas isso é facilmente obtido com análise de grafos, como a *técnica de busca de motivos* pode ser automatizada para se estender a outras permutações de redes e localizar outras transações suspeitas da mesma maneira.



Agora, vamos escrever o código básico de localização de motivos para detectar o cenário acima usando recursos de grafos. Note que a saída aqui é JSON semiestruturado; todos os tipos de dados, incluindo os não estruturados, são facilmente acessíveis no lakehouse – guardaremos esses resultados para relatórios SQL.

```
motif = "(a)-[e1]->(b); (b)-[e2]->(c); (c)-[e3]->(d); (e)-[e4]->(f); (f)-[e5]->(c); (c)-[e6]->(g)"
struct_scn_1 = aml_entity_g.find(motif)

joined_graphs = struct_scn_1.alias("a") \
    .join(struct_scn_1.alias("b"), col("a.g.id") == col("b.g.id")) \
    .filter(col("a.e6.txn_amount") + col("b.e6.txn_amount") > 10000)
```

Usando a descoberta de motivos, extraímos padrões interessantes em que o dinheiro está fluindo através de quatro entidades diferentes e mantido sob um limite de US\$ 10.000. Juntamos nossos metadados gráficos de volta aos conjuntos de dados estruturados para gerar insights para um analista de AML investigar mais.

	top_entity_id	first_entity	second_entity	third_entity	fourth_entity
1	1	Brenda Thomas	Teresa Gibson	Mary Strong	Robert Wilkinson
2	3	Lindsey Barber	Joshua Harris	Mary Strong	Robert Wilkinson
3	5	Bruce White	Kathleen Elliott	Victor Arias	Robert Wilkinson
4	7	Jeffrey Lara	Amy Campbell	Victor Arias	Robert Wilkinson

Figura 5

### Cenário 3 – propagação de pontuação de risco

As entidades de alto risco identificadas terão uma influência (um efeito de rede) em seu círculo. Portanto, a pontuação de risco de todas as entidades com as quais interagem deve ser ajustada para refletir a zona de influência. Usando uma abordagem iterativa, podemos acompanhar o fluxo de transações em qualquer profundidade e ajustar as pontuações de risco de outras pessoas afetadas na rede. Como mencionado anteriormente, executar análises de gráfico evita vários joins do SQL repetidos e lógica de negócios complexa, o que pode afetar o desempenho devido a restrições de memória. Análises de grafos e API de Pregel foram criadas para esse propósito exato. Inicialmente desenvolvido pelo Google, **Pregel** permite que os usuários “propaguem” mensagens recursivamente de qualquer vértice para seus vizinhos correspondentes, atualizando o estado do vértice (sua pontuação de risco aqui) em cada etapa. Podemos representar nossa abordagem de risco dinâmico usando API de Pregel como segue.

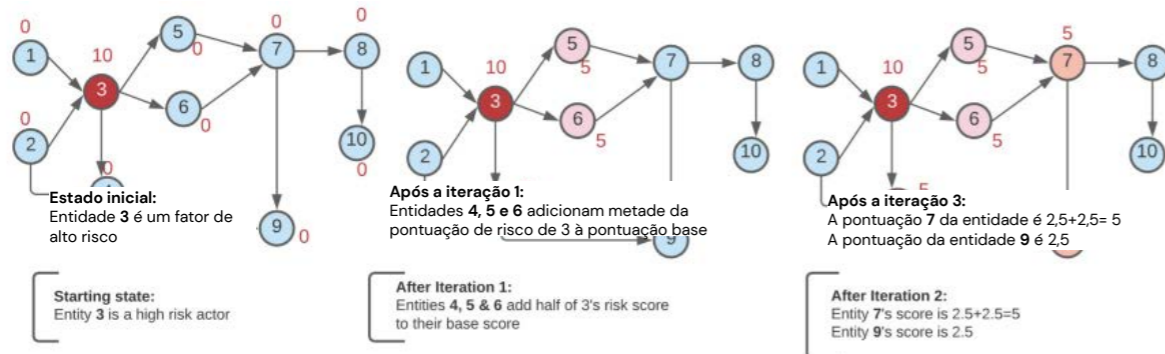


Figura 6

O diagrama abaixo à esquerda mostra o estado inicial da rede e duas iterações subsequentes. Digamos que começamos com um malfeitor (Nó 3) com uma pontuação de risco de 10. Queremos penalizar todas as pessoas que efetuam transações com esse nó (ou seja, Nós 4, 5 e 6) e receber fundos repassando, por exemplo, metade da pontuação de risco do malfeitor, que então é adicionado à sua pontuação base. Na próxima iteração, todos os nós que estão downstream dos nós 4, 5 e 6 terão suas pontuações ajustadas.

Nó n.º	Iteração n.º 0	Iteração n.º 1	Iteração n.º 2
1	0	0	0
2	0	0	0
3	10	10	10
4	0	5	5
5	0	5	5
6	0	5	5
7	0	0	5
8	0	0	0
9	0	0	2,5
10	0	0	0

Usando a **API do Pregel** do GraphFrame, podemos fazer esse cálculo e persistir com as pontuações modificadas de outros aplicativos downstream para consumir.

```
from graphframes.lib import Pregel

ranks = aml_entity_g.pregel \
    .setMaxIter(3) \
    .withVertexColumn(
        "risk_score",
        col("risk"),
        coalesce(Pregel.msg()+ col("risk"),
        col("risk_score"))
    ) \
    .sendMsgToDst(Pregel.src("risk_score")/2) \
    .aggMsgs(sum(Pregel.msg())) \
    .run()
```

## Correspondência de endereço

Um padrão que queremos abordar brevemente é a correspondência de endereços de texto com imagens reais do Street View. Muitas vezes, há a necessidade de um analista de AML validar a legitimidade dos endereços vinculados a entidades registradas. Este endereço é um edifício comercial, uma área residencial ou uma caixa postal simples? No entanto, analisar imagens é muitas vezes um processo tedioso, demorado e manual para obter, limpar e validar. Uma arquitetura de dados de lakehouse nos permite automatizar a maior parte desta tarefa usando tempos de execução Python e ML com PyTorch e modelos de código aberto pré-treinados. Veja abaixo um exemplo de um endereço válido para o olho humano. Para automatizar a validação, usaremos um modelo VGG pré-treinado para o qual existem centenas de objetos válidos que podemos usar para detectar uma residência.



Figura 7

Usando o código abaixo, que pode ser automatizado para ser executado diariamente, agora teremos um rótulo anexado a todas as nossas imagens — carregamos todas as referências de imagem e rótulos em uma tabela SQL para facilitar a consulta também. Observe no código abaixo como é simples consultar um conjunto de imagens para os objetos dentro delas — a capacidade de consultar esses dados não estruturados com o Delta Lake economiza um tempo enorme para os analistas e acelera o processo de validação para minutos, em vez de dias ou semanas.

```
from PIL import Image
from matplotlib import cm

img = Image.fromarray(img)
...

vgg = models.vgg16(pretrained=True)
prediction = vgg(img)
prediction = prediction.data.numpy().argmax()
img_and_labels[i] = labels[prediction]
```

À medida que começamos a resumir, notamos que algumas categorias interessantes aparecem. Como visto na divisão abaixo, existem alguns rótulos óbvios, como *pátio*, *trailer* e *motocicletas* que esperaríamos ver como itens detectados em um endereço residencial. Por outro lado, o modelo CV rotulou uma antena parabólica de objetos circundantes em uma imagem. (nota: uma vez que estamos restritos a um modelo de código aberto não treinado em um conjunto personalizado de imagens, o rótulo de antena parabólica não é preciso.) Após uma análise mais aprofundada da imagem, vemos que i) não há uma antena parabólica real aqui e, mais importante, ii) este endereço não é uma residência real (retratado em nossa comparação lado a lado na Figura 7). O formato Delta Lake nos permite armazenar uma referência aos nossos dados não estruturados, juntamente com um rótulo para consultas simples em nosso detalhamento de classificação abaixo.

### Address Validation

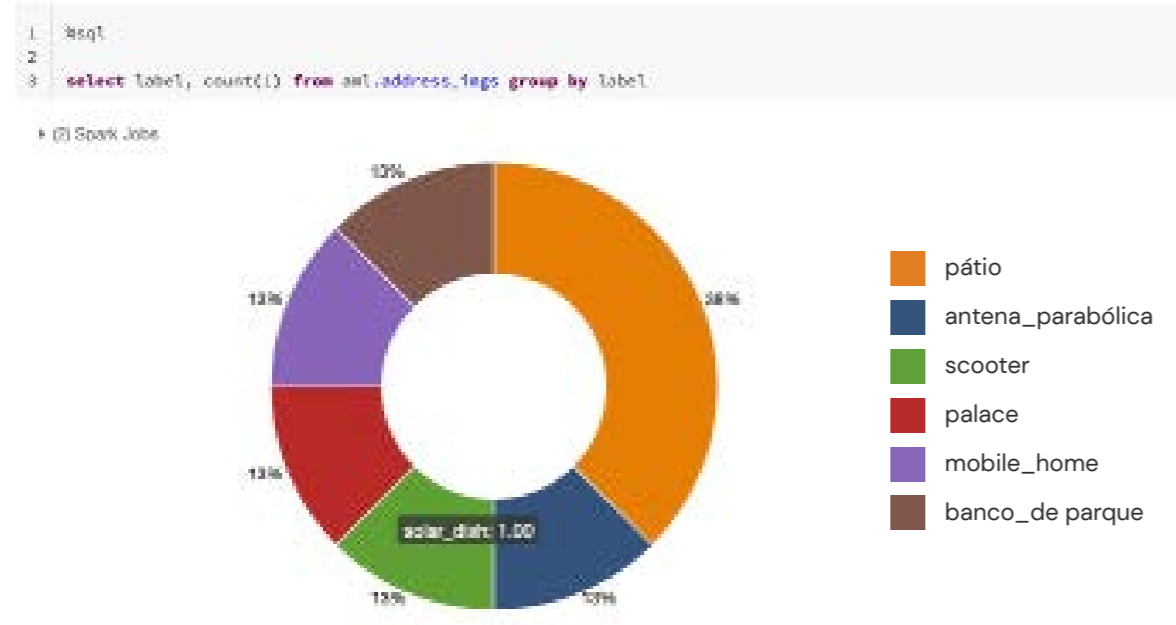


Figura 8



Image Name	Rendered Image	Main Object	Risk Level
img_0.jpg		Patio	Low
img_1.jpg		Solar Dish	High

Figura 9



## Resolução de entidade

A última categoria de desafios de AML em que nos concentraremos é a resolução de entidades. Muitas bibliotecas de código aberto resolvem esse problema, então, para algumas combinações básicas de entidades, optamos por destacar o **Splink**, que alcança a ligação em escala e oferece configurações para especificar colunas correspondentes e regras de bloqueio.

No contexto das entidades derivadas de nossas transações, é um exercício simples inserir nossas transações do Delta Lake no contexto do Splink.

```
settings = {
  "link_type": "dedupe_only",
  "blocking_rules": [
    "l.txn_amount = r.txn_amount",
  ],
  "comparison_columns": [
    {
      "col_name": "rptd_originator_address",
    },
    {
      "col_name": "rptd_originator_name",
    }
  ]
}

from splink import Splink
linker = Splink(settings, df2, spark)
df2_e = linker.get_scored_comparisons()
```

O Splink funciona atribuindo uma probabilidade de correspondência que pode ser usada para identificar transações nas quais os atributos da entidade são altamente semelhantes, levantando um alerta em potencial em relação a um endereço relatado, nome da entidade ou valor da transação. Dado o fato de que a resolução da entidade pode ser altamente manual para combinar informações da conta, ter bibliotecas de código aberto que automatizam essa tarefa e salvam as informações no Delta Lake pode tornar os investigadores muito mais produtivos para a resolução de casos. Embora haja várias opções disponíveis para correspondência de entidade, recomendamos usar o Locality-Sensitive Hashing (LSH) para identificar o algoritmo certo para o trabalho. Você pode saber mais sobre o LSH e seus benefícios nesta [publicação do blog](#).

Conforme relatado acima, encontramos rapidamente algumas inconsistências para o endereço bancário NY Mellon, com "Canada Square, Canary Wharf, Londres, Reino Unido" semelhante a "Canada Square, Canary Wharf, Londres, Reino Unido". Podemos armazenar nossos registros deduplicados de volta numa tabela Delta que pode ser usada para investigação de AML.

unique_id_l ▲	unique_id_r ▲	rptd_originator_address_l ▲	rptd_originator_address_r ▲
223254	223256	Canada Square, Canary Wharf, London, United Kingdom	Canada Square, Canary Wharf, London, UK

Figura 10

## Painel de lakehouse AML

O Databricks SQL no lakehouse está se aproximando dos data warehouses tradicionais em termos de gerenciamento simplificado de dados, desempenho com o novo mecanismo de consulta Photon e concorrência do usuário. Isso é importante, pois muitas organizações não têm o orçamento para o excessivamente caro software AML proprietário para corroborar os diversos casos de uso, como combater o financiamento do terrorismo (CFT), que ajuda a combater o crime financeiro. No mercado, existem soluções dedicadas que podem executar a análise gráfica acima, soluções dedicadas para lidar com BI em um warehouse e soluções dedicadas para ML. O design do lakehouse de AML unifica todos os três. As equipes de plataforma de dados da AML podem aproveitar o Delta Lake ao menor custo de armazenamento em nuvem, ao mesmo tempo em que integram facilmente tecnologias de código aberto para produzir relatórios selecionados com base em tecnologia de gráficos, visão computacional e engenharia de análise SQL. Na figura 11, vamos mostrar uma materialização do relatório de AML.

Os notebooks anexados produziram um objeto de transações, objeto de entidades, bem como resumos como previsões de estruturação, camadas de identidade sintéticas e classificações de endereços usando modelos pré-treinados. Na visualização do Databricks SQL abaixo, usamos nosso mecanismo Photon SQL para executar resumos sobre eles e visualização integrada para produzir um painel de relatórios em poucos minutos. Há ACLs completos em ambas as tabelas, bem como no próprio painel, para permitir que os usuários compartilhem com executivos e equipes de dados — um agendador para executar esse relatório periodicamente também está integrado. O painel é a culminação da engenharia de IA, BI e análise integrada à solução AML.

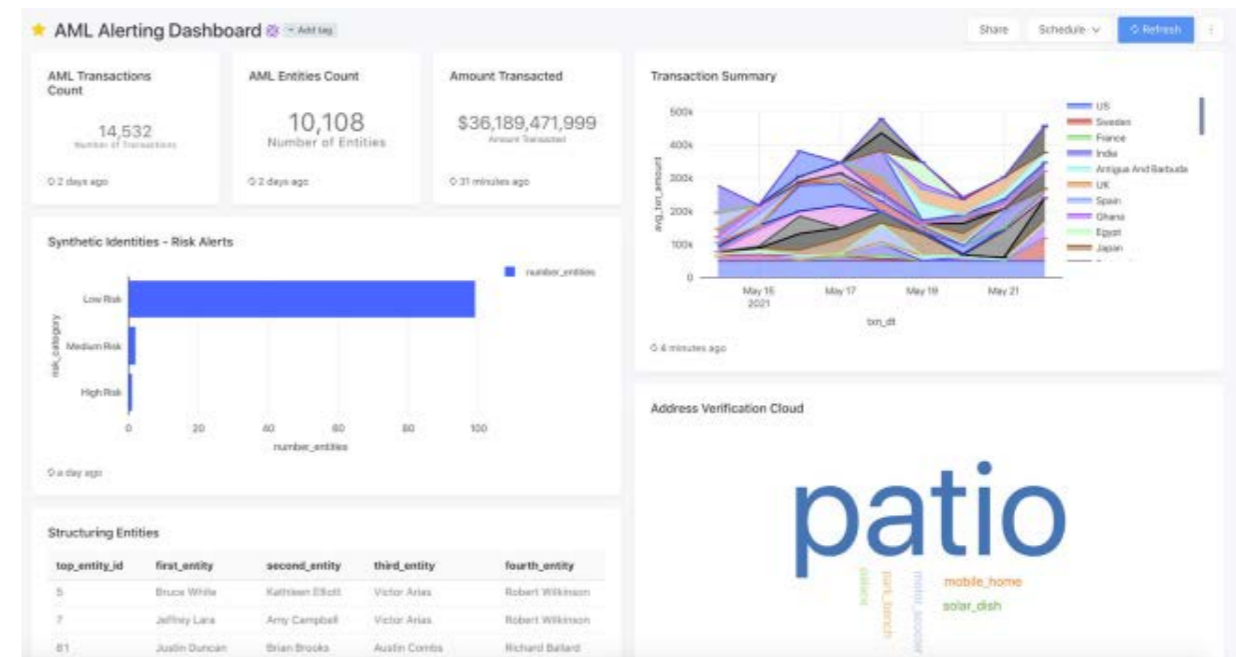


Figura 11

## A transformação do open banking

O aumento do open banking permite que as Instituições Financeiras (IFs) forneçam uma melhor experiência ao cliente através do compartilhamento de dados entre consumidores, IFs e provedores de serviços de terceiros por meio de APIs. Um exemplo disso é a Diretiva de Serviços de Pagamento (PSD2), que transformou os serviços financeiros na região da UE como parte da **regulação** do Open Banking na Europa. Como resultado, as IFs têm acesso a mais dados de vários bancos e provedores de serviços, incluindo dados de conta de clientes e transações. Essa tendência se expandiu no mundo de fraudes e crimes financeiros com a mais recente orientação do FinCEN na **seção 314(b)** da Lei Patriótica dos EUA; as IFs cobertas agora podem compartilhar informações com outras IFs e em agências nacionais e estrangeiras sobre indivíduos, entidades, organizações e assim por diante que são suspeitas de estarem envolvidas em possível lavagem de dinheiro.

Embora a provisão de compartilhamento de informações ajude na transparência e proteja os sistemas financeiros dos Estados Unidos contra lavagem de dinheiro e financiamento de terrorismo, a troca de informações deve ser feita usando protocolos com proteção adequada de dados e segurança. Para resolver o problema de proteger o compartilhamento de informações, a Databricks anunciou recentemente o **Delta Sharing**, um protocolo aberto e seguro para o compartilhamento de dados. Usando APIs de código aberto conhecidas, como pandas e Spark, produtores de dados e consumidores agora podem compartilhar dados usando protocolos seguros e abertos e manter uma auditoria completa de todas as transações de dados para manter a conformidade com as regulamentações da FinCEN.

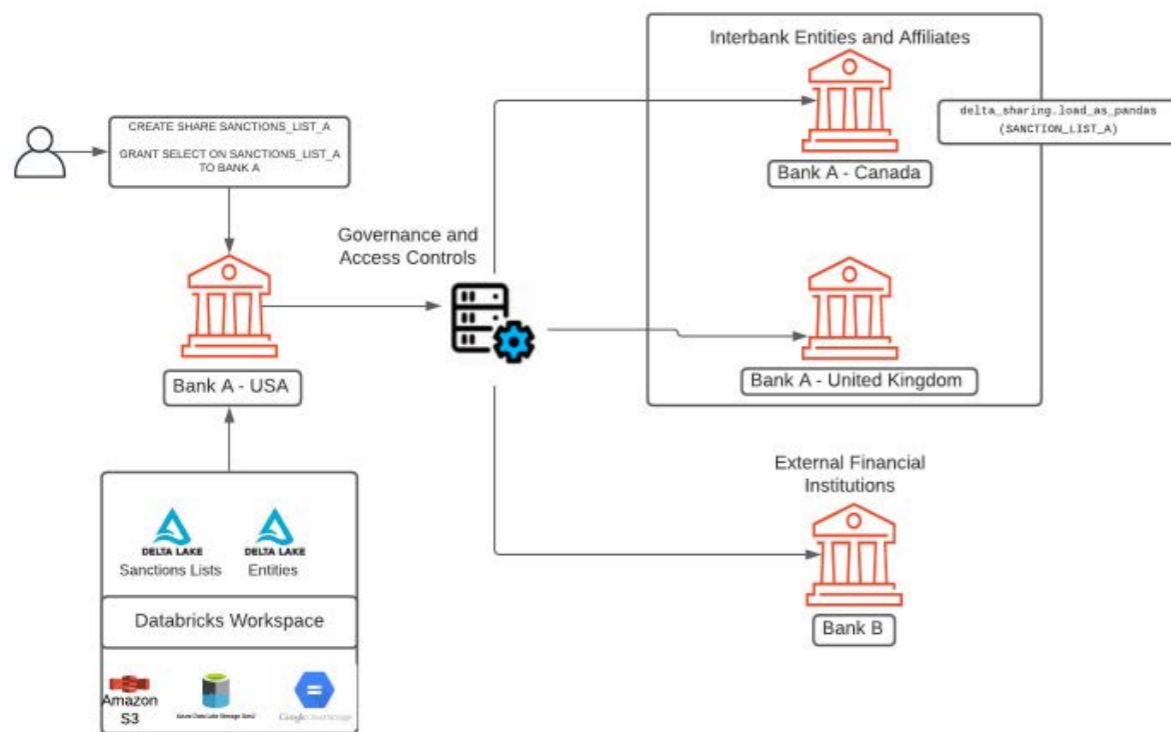
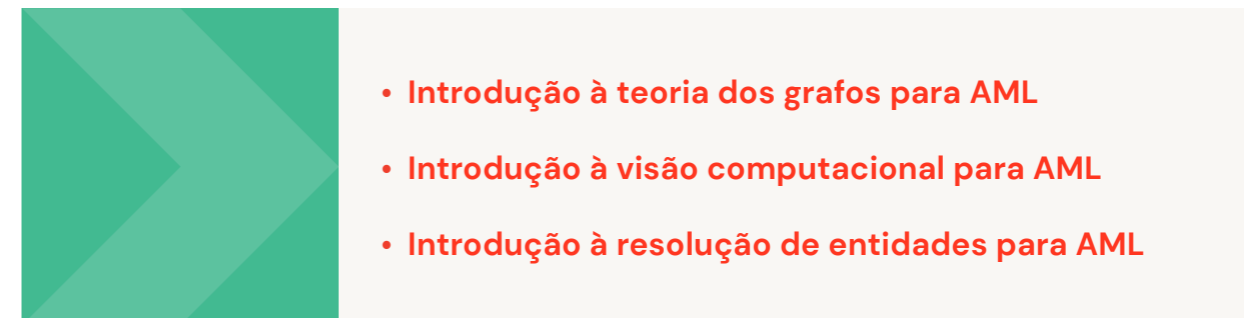


Figura 12

## Conclusão

A arquitetura lakehouse é a plataforma mais escalável e versátil para capacitar analistas em suas funções analíticas AML. Lakehouse oferece suporte a casos de uso que vão desde fuzzy match até funções analíticas de imagem e BI com painéis integrados, e todos esses recursos permitirão que as organizações reduzam o custo total de propriedade em comparação com soluções AML proprietárias. A equipe de serviços financeiros da Databricks está trabalhando em uma variedade de problemas de negócios no espaço de serviços financeiros e permitindo que profissionais de engenharia de dados e ciência de dados iniciem a jornada Databricks por meio de **Aceleradores de soluções** como AML.

## Comece a experimentar com estes notebooks Databricks gratuitos





## SEÇÃO 2.6 **Crie um modelo de IA em tempo real para Detectar comportamentos tóxicos em jogos**

por **DAN MORRIS** e **DUNCAN DAVIS**

16 de junho de 2021

Através de jogos multijogador massivos online (MMOs), jogos de arena de batalha multijogador online (MOBAs) e outras formas de jogos online, os jogadores interagem continuamente em tempo real para coordenar ou competir à medida que se movem em direção a um objetivo comum — vencer. Essa interatividade é essencial para a dinâmica do jogo, mas, ao mesmo tempo, é uma abertura privilegiada para o comportamento tóxico — um problema generalizado em toda a esfera de jogos online.



O comportamento tóxico se manifesta de muitas formas, como os graus variados de griefing, assédio virtual e assédio sexual ilustrados na matriz acima, diretamente da **Behaviour Interactive**, que lista os tipos de interações vistas no jogo multijogador "Dead by Night".

**Diagrama de toxicidade**

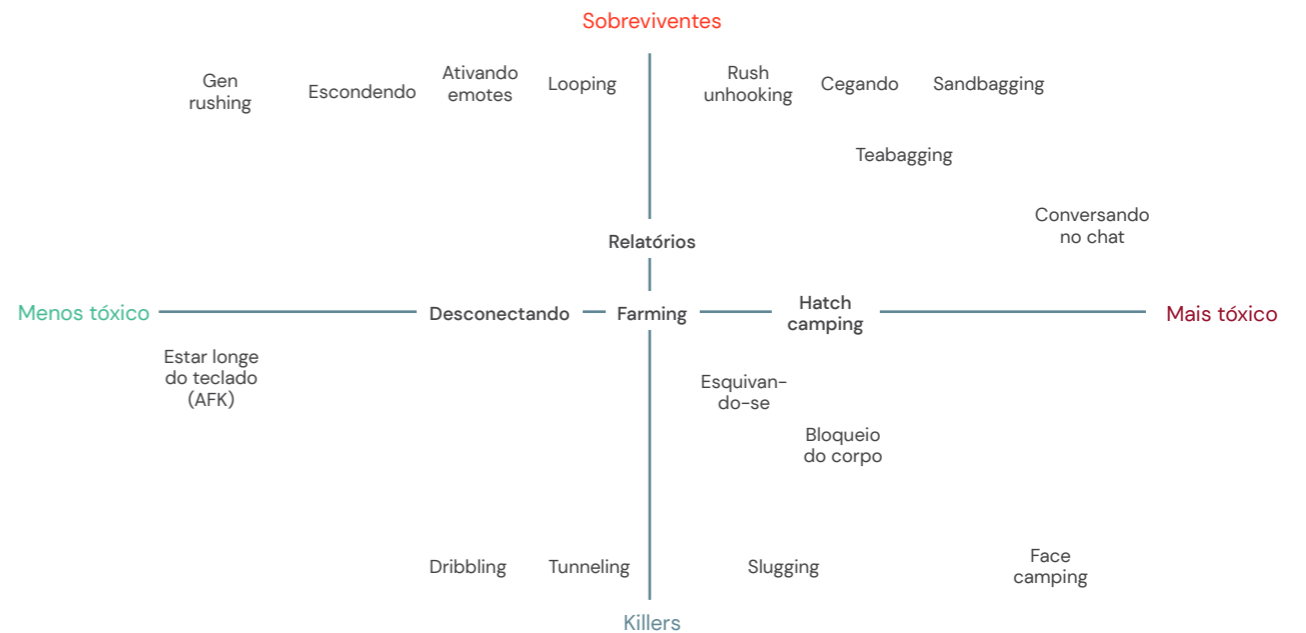


Figura 1  
Matriz de interações tóxicas que os jogadores experimentam

Além do **impacto pessoal** que o comportamento tóxico pode ter nos jogadores e na comunidade — um problema que não pode ser subdimensionado — também é prejudicial para muitos estúdios de jogos. Por exemplo, um estudo da **Universidade Estadual de Michigan** revelou que 80% dos jogadores sofreram toxicidade recentemente e, desses, 20% relataram deixar o jogo devido a essas interações. Da mesma forma, um estudo da **Universidade de Tilburgo** mostrou que ter um encontro tóxico ou disruptivo na primeira sessão do jogo levou jogadores a serem mais de três vezes mais propensos a sair do jogo sem retornar. Como a retenção de jogadores é uma prioridade para muitos estúdios, principalmente porque os jogos passam por uma transição de versões da mídia física para serviços de longa duração, é claro que a toxicidade deve ser combatida.

Agravando esse problema relacionado à rotatividade, algumas empresas enfrentam desafios relacionados à toxicidade no início do desenvolvimento, mesmo antes do lançamento. Por exemplo, o **Amazon's Crucible** foi lançado em testes sem mensagens de texto ou chat por voz devido, em parte, a não ter um sistema instalado para monitorar ou gerenciar gamers tóxicos e interações. Isso ilustra que a dimensão do espaço de jogos ultrapassou muito a capacidade da maioria das equipes de gerenciar esse comportamento por meio de relatórios ou intervenções em interações disruptivas. Dado isso, é essencial que os estúdios integrem funções analíticas aos jogos no início do ciclo de vida de desenvolvimento e, em seguida, projetem para o gerenciamento contínuo de interações tóxicas.

A toxicidade nos jogos é claramente um problema multifacetado que se tornou parte da cultura de jogos eletrônicos e não pode ser abordada universalmente de uma única maneira. Dito isso, abordar a toxicidade no chat no jogo pode ter um impacto enorme dada a frequência do comportamento tóxico e a capacidade de automatizar sua detecção usando o processamento de linguagem natural (NLP).

## Conheça a detecção de toxicidade no acelerador de soluções de jogos da Databricks

Usando **dados de comentários tóxicos** dos jogos Jigsaw e **Dota 2**, este Acelerador de soluções percorre as etapas necessárias para detectar comentários tóxicos em tempo real usando o NLP e seu **lakehouse** existente. Para o NLP, este Acelerador de soluções usa o **Spark NLP** do John Snow Labs, uma solução de código aberto de nível empresarial criada nativamente no Apache Spark.™

As etapas que você seguirá neste Acelerador de soluções são:

- Carregue os dados do Jigsaw e Dota 2 em tabelas usando o Delta Lake
- Classifique os comentários tóxicos usando a classificação multirrótulo (**Spark NLP**)

- Rastreie experimentos e registre modelos usando o MLflow
- Aplique inferência em dados em batch e streaming
- Examine o impacto da toxicidade nos dados de jogos

## Detectar toxicidade no bate-papo do jogo durante a produção

Com o Acelerador de soluções, agora você pode integrar mais facilmente a detecção de toxicidade em seus próprios jogos. Por exemplo, a arquitetura de referência abaixo mostra como pegar dados do chat e de jogos de uma variedade de fontes, como streams, arquivos, bancos de dados de voz ou operacionais, e aproveitar a Databricks para ingerir, armazenar e organizar dados em tabelas de recursos para pipelines de machine learning (ML), ML no jogo,

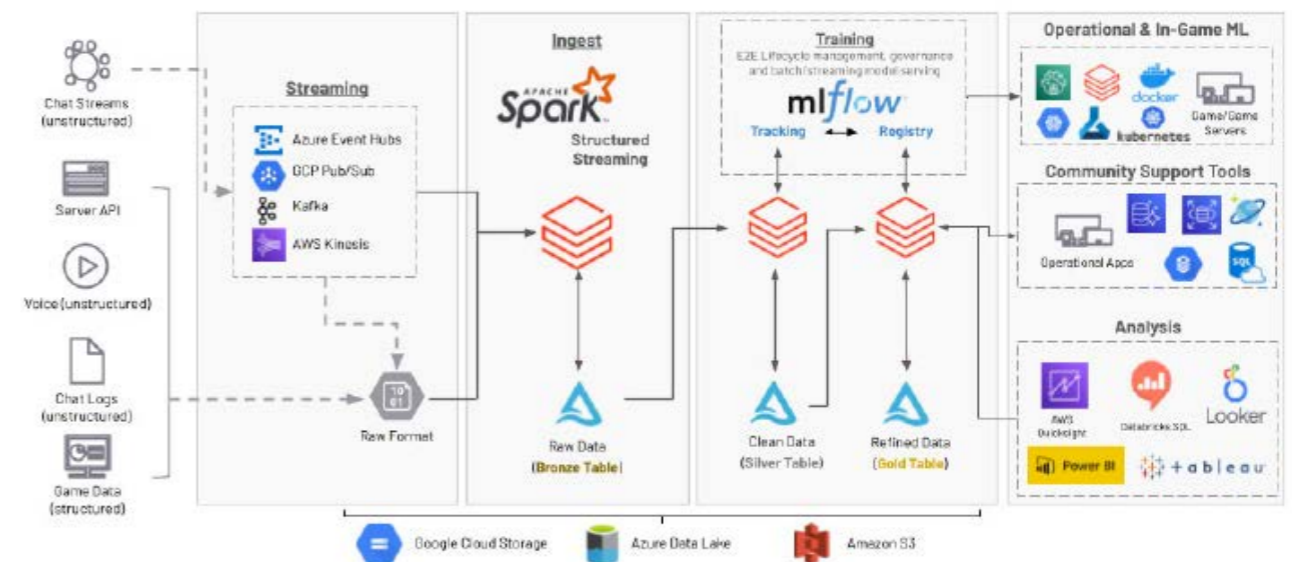


Figura 2  
Arquitetura de referência de detecção de toxicidade

Ter uma arquitetura escalável em tempo real para detectar toxicidade na comunidade permite simplificar os fluxos de trabalho para gerentes de relacionamento com a comunidade e a capacidade de filtrar milhões de interações em cargas de trabalho gerenciáveis. Da mesma forma, a possibilidade de alertar sobre eventos altamente tóxicos em tempo real, ou até mesmo automatizar uma resposta, como silenciar jogadores ou alertar um CRM sobre o incidente rapidamente, pode ter um impacto direto na retenção de jogadores. Além disso, ter uma plataforma capaz de processar grandes conjuntos de dados, de fontes diversas, pode ser usado para monitorar a percepção da marca por meio de relatórios e painéis.

## Introdução

O objetivo deste Acelerador de soluções é ajudar o gerenciamento contínuo de interações tóxicas em jogos online, permitindo a detecção em tempo real de comentários tóxicos no bate-papo no jogo. Comece hoje importando este Acelerador de soluções diretamente para o seu espaço de trabalho Databricks.

Depois de importados, você terá notebooks com dois pipelines prontos para serem movidos para a produção.

- Pipeline de ML usando classificação multirrotulo com treinamento em conjuntos de dados em inglês do mundo real do Google Jigsaw. O modelo classificará e rotulará as formas de toxicidade no texto.
- Pipeline de inferência de streaming em tempo real que utiliza o modelo de toxicidade. A origem do pipeline pode ser facilmente modificada para ingerir dados de chat de todas as fontes de dados comuns.

Com esses dois pipelines, você pode começar a entender e analisar a toxicidade com mínimo esforço. Este Acelerador de soluções também fornece uma base para construir, personalizar e melhorar o modelo com dados



**Comece a experimentar esses notebooks Databricks gratuitos.**

## SEÇÃO 2.7 **Impulsionar a Transformação na Northwestern Mutual (Insights Platform) ao Avançar para uma Arquitetura de Lakehouse Escalável e Aberta**

por **MADHU KOTIAN**

15 de julho de 2021

A transformação digital tem sido uma questão central na maioria das iniciativas corporativas de big data contemporâneas, especialmente em empresas com forte pegada legada. Um dos componentes principais na transformação digital são os dados e seu armazenamento de dados relacionado. Há mais de 160 anos, a Northwestern Mutual tem ajudado famílias e empresas a alcançar a segurança financeira. Com mais de US\$ 31 bilhões em receita, mais de 4,6 milhões de clientes e mais de 9.300 profissionais financeiros, não há muitas empresas que tenham esse volume de dados em uma variedade de fontes.

A ingestão de dados é um desafio nos dias de hoje, quando as organizações lidam com milhões de pontos de dados provenientes de diferentes formatos, prazos e direções em um volume sem precedentes. Queremos preparar os dados para análise a fim de dar-lhes sentido. Hoje, estou entusiasmado em compartilhar nossa nova abordagem para transformar e modernizar nosso processo de ingestão de dados, processo de agendamento e jornada com armazenamento de dados. Uma coisa que aprendemos é que uma abordagem eficaz é multifacetada, e é por isso que, além das medidas técnicas, vou mostrar nosso plano para embarcar você com a nossa equipe.

### **Desafios enfrentados**

Antes de embarcarmos em nossa transformação, trabalhamos com nossos parceiros de negócios para realmente entender nossas restrições técnicas e nos ajudar a moldar a descrição de problema para nosso caso de negócios.

O ponto crítico do negócio que identificamos foi a falta de dados integrados, com dados de clientes e negócios provenientes de diferentes equipes internas e externas e fontes de dados. Percebemos o valor dos dados em tempo real, mas tínhamos acesso limitado a dados de produção/tempo real que poderiam nos permitir tomar decisões de negócios em tempo hábil. Aprendemos também que os armazenamentos de dados criados pela equipe de negócios resultaram em silos de dados, o que, por sua vez, causou problemas de latência de dados, aumento do custo de gerenciamento de dados e restrições de segurança indesejadas.

Além disso, houve desafios técnicos em relação ao nosso estado atual. Com o aumento da demanda e a necessidade de dados adicionais, experimentamos restrições com a escalabilidade da infra-estrutura, a latência de dados, o custo de gerenciamento de silos de dados, as limitações de tamanho e volume de dados e questões de segurança de dados. Com esses desafios crescendo, sabíamos que tínhamos muito a enfrentar e precisávamos encontrar os parceiros certos para nos ajudar em nossa jornada de transformação.

## Análise da solução

Precisávamos nos tornar orientados por dados para ser competitivos e atender melhor aos nossos clientes e otimizar os processos internos. Exploramos várias opções e realizamos várias POCs para selecionar uma recomendação final. Os itens a seguir foram obrigatórios para nossa estratégia de avanço:

- Uma solução completa para nossa ingestão de dados, gerenciamento de dados e necessidades analíticas
- Uma plataforma de dados moderna que pode efetivamente apoiar nossos desenvolvedores e analistas de negócios para realizar suas análises usando SQL
- Um mecanismo de dados que pode oferecer suporte a transações ACID além do S3 e habilite a segurança baseada em funções
- Um sistema que pode proteger eficazmente nossas informações de PII/PHI
- Uma plataforma que pode ser dimensionada automaticamente com base no processamento de dados e na demanda analítica

Nossa infraestrutura legada foi baseada no MSBI Stack. Usamos SSIS para ingestão, SQL Server para nosso armazenamento de dados, Azure Analysis Service para modelo tabular e Power BI para criação de painéis e relatórios. Embora a plataforma tenha atendido às necessidades do negócio inicialmente, tivemos desafios no dimensionamento com o aumento do volume de dados e da demanda de processamento de dados e restringimos nossas expectativas analíticas de dados. Com necessidades adicionais de dados, nossos problemas de latência de dados devido a atrasos de carga e um armazenamento de dados para necessidades específicas de negócios causaram silos de dados e expansão de dados.

A segurança tornou-se um desafio devido à disseminação de dados em vários armazenamentos de dados. Tivemos aproximadamente 300 jobs de ETL que levaram mais de 7 horas de nossos jobs diários. O tempo de lançamento no mercado para qualquer mudança ou novo desenvolvimento foi de aproximadamente 4 a 6 semanas (dependendo da complexidade).



Figura 1  
Arquitetura legada

Depois de avaliar várias soluções no mercado, decidimos avançar com a Databricks para nos ajudar a fornecer uma solução integrada de gerenciamento de dados em uma arquitetura aberta de lakehouse.



O fato de a Databricks ter sido desenvolvida sobre o Apache Spark™ nos permitiu usar o Python para construir nossa estrutura personalizada para ingestão de dados e gerenciamento de metadados.

Ele nos proporcionou a flexibilidade para realizar análises ad hoc e outras descobertas de dados usando o notebook. O Databricks Delta Lake (a camada de armazenamento construída sobre nosso data lake) nos forneceu a flexibilidade de implementar várias funções de gerenciamento de banco de dados (transações ACID, governança de metadados, viagens de tempo etc.), incluindo a implementação dos controles de segurança necessários. A Databricks tirou a dor de cabeça do gerenciamento/dimensionamento de cluster e reagiu efetivamente à demanda reprimida de nossos engenheiros e usuários de negócios.

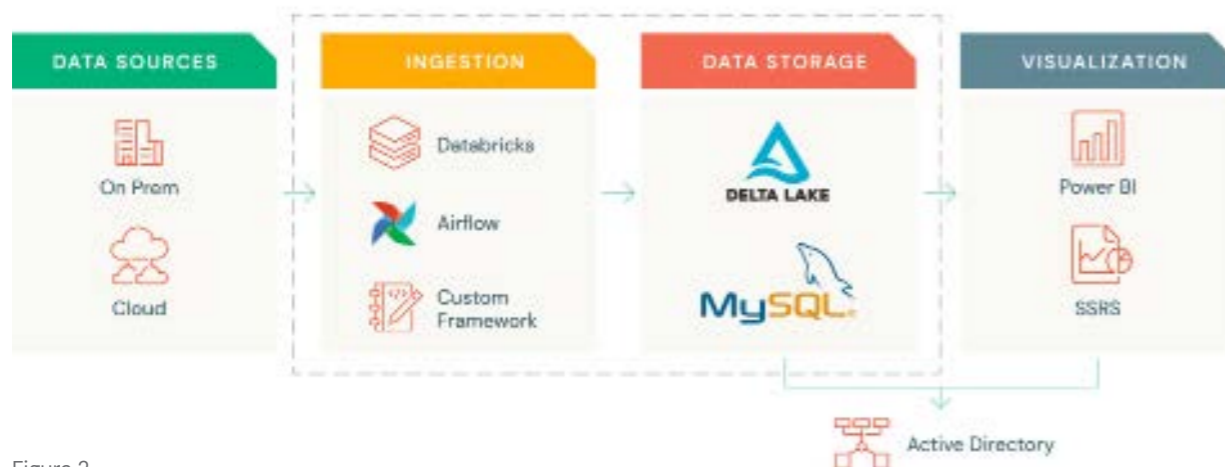


Figura 2  
Arquitetura com Databricks

Começamos com um pequeno grupo de engenheiros e os atribuímos a uma equipe virtual de nossa equipe Scrum existente. O objetivo deles era executar diferentes POCs, consolidar a solução recomendada, desenvolver melhores práticas e fazer a transição de volta para sua respectiva equipe para ajudar na integração. Aproveitar os membros existentes da equipe nos ajudou porque eles tinham conhecimento do sistema legado existente, entendiam o fluxo de ingestão/as regras de negócios atuais e estavam bem versados em pelo menos um conhecimento de programação (engenharia de dados + conhecimento de engenharia de software). Essa equipe primeiro se treinou em Python, entendeu detalhes complexos de Spark e Delta e fez uma parceria estreita com a equipe da Databricks para validar a solução/abordagem. Enquanto a equipe estava trabalhando na formação do estado futuro, o resto de nossos desenvolvedores trabalharam na entrega das prioridades do negócio.

Como a maioria dos desenvolvedores eram engenheiros da MSBI Stack, nosso plano de ação era entregar uma plataforma de dados sem atritos para nossos desenvolvedores, usuários de negócios e consultores de campo.

- Construímos uma estrutura de ingestão que cobria todas as nossas necessidades de carga de dados e transformação. Tinha controles de segurança integrados, que mantinham todos os metadados e segredos de nossos sistemas fonte. O processo de ingestão aceitou um arquivo JSON que incluía a fonte, o alvo e a transformação necessária. Permitiu tanto uma transformação simples como complexa.
- Para agendamento, acabamos usando o Airflow, mas dada a complexidade do DAG, construímos nossa própria estrutura personalizada em cima do Airflow, que aceitava um arquivo YAML que incluía informações de job e suas interdependências relacionadas.
- Para gerenciar alterações no nível do esquema usando o Delta, criamos nossa própria estrutura personalizada, que automatizou diferentes operações de tipo de banco de dados (DDL) sem exigir que os desenvolvedores tenham acesso de emergência ao armazenamento de dados. Isso também nos ajudou a implementar diferentes controles de auditoria no armazenamento de dados.



Em paralelo, a equipe também trabalhou com nossa equipe de segurança para garantir que entendemos e atendemos a todos os critérios de segurança de dados (criptografia em trânsito, criptografia em repouso e criptografia em nível de coluna para proteger as informações de PII).

Uma vez que essas estruturas foram configuradas, a equipe de coorte implementou um fluxo de ponta a ponta (fonte até alvo com toda a transformação) e gerou um novo conjunto de relatórios/painéis sobre Power BI apontando para o Delta Lake. O objetivo era testar o desempenho do nosso processo de ponta a ponta, validar os dados e obter qualquer feedback dos nossos usuários de campo. Melhoramos progressivamente o produto com base no feedback e nos resultados do nosso teste de desempenho/validação.

Simultaneamente, criamos guias de treinamento e instruções para integrar nossos desenvolvedores. Logo depois, decidimos mover os membros da equipe de coorte para suas respectivas equipes e, ao mesmo tempo, manter alguns para continuar apoiando a infraestrutura de plataforma (DevOps). Cada equipe do Scrum foi responsável por gerenciar e entregar seu respectivo conjunto de recursos à empresa. Uma vez que os membros da equipe voltaram para suas respectivas equipes, eles embarcaram na tarefa de ajustar a velocidade da equipe para incluir o backlog para o esforço de migração. Os leads da equipe receberam orientações específicas e metas adequadas para cumprir os objetivos de migração para diferentes Incrementos de Sprint/Programa. Os membros da equipe que estavam no grupo de coorte agora eram os especialistas residentes e ajudaram sua equipe a integrar à nova plataforma. Eles estavam disponíveis para quaisquer perguntas ou assistência ad hoc.

À medida que construímos progressivamente nossa nova plataforma, mantivemos a antiga plataforma para validação e verificação.

## O início do sucesso

A transformação global levou cerca de um ano e meio, o que é um feito e tanto, dado que tivemos que construir todas as estruturas, gerenciar prioridades de negócios, gerenciar expectativas de segurança, reorganizar nossa equipe e migrar a plataforma. O tempo de carregamento geral diminuiu notavelmente de 7 horas para apenas 2 horas. Nosso tempo de lançamento no mercado foi de cerca de 1 a 2 semanas, diminuindo significativamente de 4 a 6 semanas. Essa foi uma grande melhoria e sei que se estenderá ao nosso negócio de várias maneiras.

Nossa jornada ainda não terminou. Enquanto continuamos a aprimorar nossa plataforma, nossa próxima missão será expandir o padrão do lakehouse. Estamos trabalhando na migração de nossa plataforma para o E2 e na implantação do Databricks SQL. Estamos trabalhando em nossa estratégia para fornecer uma plataforma de autoatendimento aos nossos usuários comerciais para realizar suas análises ad hoc e também permitir que eles tragam seus próprios dados com a capacidade de realizar análises com nossos dados integrados. O que aprendemos é que nos beneficiamos muito ao utilizar uma plataforma que era aberta, unificada e escalável. À medida que nossas necessidades e capacidades crescem, sabemos que temos um parceiro robusto na Databricks.

---

Saiba mais sobre [A jornada da Northwestern Mutual's para o Lakehouse](#)

### SOBRE MADHU KOTIAN

Madhu Kotian é vice-presidente de engenharia (dados de produtos de investimento, CRM, aplicativos e relatórios) na Northwestern Mutual. Ele tem mais de 25 anos de experiência na área de Tecnologia da Informação, com experiência em engenharia de dados, gerenciamento de pessoas, gerenciamento de programas, arquitetura, design, desenvolvimento e manutenção usando práticas ágeis. Ele também é especialista em metodologias de data warehouse e implementação de integração e análise de dados.

## SEÇÃO 2.8

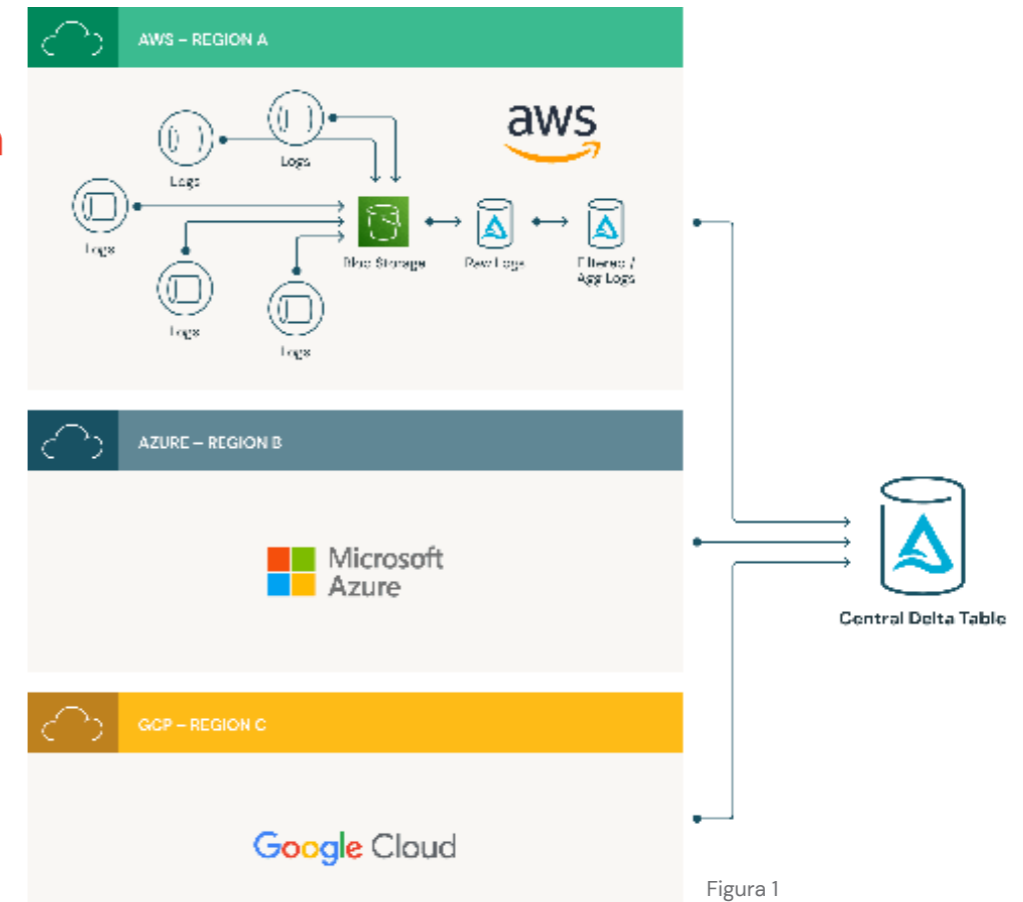
## Como a equipe de dados da Databricks criou um lakehouse em três nuvens e mais de 50 regiões

de JASON POHL e SURAJ ACHARYA

14 de julho de 2021

A infraestrutura de registro interno na Databricks evoluiu ao longo dos anos, e aprendemos algumas lições para manter um pipeline de registro altamente disponível em várias nuvens e geografias. Este blog lhe dará algumas informações sobre como coletamos e administramos métricas em tempo real usando nossa plataforma lakehouse e como aproveitamos várias nuvens para ajudar na recuperação de paralisações de nuvens públicas.

Quando a Databricks foi fundada, ela só suportava uma única nuvem pública. Agora, o serviço cresceu para oferecer suporte às três principais nuvens públicas (AWS, Azure, GCP) em mais de 50 regiões ao redor do mundo. Todos os dias, a Databricks move milhões de máquinas virtuais em nome de nossos clientes. Nossa equipe de plataforma de dados de menos de 10 engenheiros é responsável por construir e manter a infraestrutura de telemetria de registro, que processa meio petabyte de dados todos os dias. A orquestração, o monitoramento e o uso são capturados por meio de logs de serviço processados por nossa infraestrutura para fornecer métricas precisas e oportunas. No final, esses dados são armazenados em nosso próprio Delta Lake de tamanho petabyte. Nossa equipe do Data Platform usa a Databricks para executar o processamento em nuvem para que possamos federar os dados quando apropriado, mitigar a recuperação de uma interrupção em nuvem regional e minimizar a interrupção para nossa infraestrutura em tempo real.



### Arquitetura de pipeline

Cada região de nuvem contém sua própria infraestrutura e pipelines de dados para capturar, coletar e manter dados de log em um Delta Lake regional. Os dados de telemetria do produto são capturados em todo o produto e em nossos pipelines pelo mesmo processo replicado em todas as regiões da nuvem. Um daemon de log captura os dados de telemetria e, em seguida, grava esses logs em um bucket de armazenamento em nuvem regional (S3, WASBS, GCS). A partir daí, um pipeline agendado ingerirá os arquivos de log usando o Auto Loader (AWS | Azure | GCP), e gravará os dados em uma tabela regional Delta, os filtrará e os gravará em uma tabela centralizada Delta em uma região de nuvem única.

## Antes do Delta Lake

Antes do Delta Lake, escrevamos os dados de origem em sua própria tabela no lake centralizado e, em seguida, criávamos uma visão que era uma união em todas essas tabelas. Essa visualização precisava ser calculada no tempo de execução e se tornou mais ineficiente à medida que adicionamos mais regiões:

```
CREATE OR REPLACE VIEW all_logs AS
SELECT * FROM (
  SELECT * FROM region_1.log_table
  UNION ALL
  SELECT * FROM region_2.log_table
  UNION ALL
  SELECT * FROM region_3.log_table
  ...
);
```

## Depois do Delta Lake

Hoje, temos apenas uma única tabela Delta que aceita gravações simultâneas de mais de 50 regiões diferentes. Ao mesmo tempo em que lida com consultas nos dados. Isso torna a consulta da tabela central tão fácil quanto:

```
SELECT * FROM central.all_logs;
```

A transacionalidade é tratada pelo Delta Lake. Descontinuamos as tabelas regionais individuais em nosso Delta Lake central e aposentamos a visão UNION ALL. O código a seguir é uma representação simplificada da sintaxe que é executada para carregar os dados aprovados para a saída dos Delta Lakes regionais para o Delta Lake central:

```
spark.readStream.format("delta")
  .load(regional_source_path)
  .where("egress_approved = true")
  .writeStream
  .format("delta")
  .outputMode("append")
  .option("checkpointLocation", checkpoint_path)
  .start(central_target_path)
```

## Recuperação de desastre

Um dos benefícios de operar um serviço entre nuvens é que estamos bem posicionados para determinados cenários de recuperação após desastres. Embora raro, não é inédito que o serviço de computação de uma determinada região de nuvem experimente uma paralisação. Quando isso acontece, o armazenamento em nuvem é acessível, mas a capacidade de executar novas máquinas virtuais é prejudicada. Como desenvolvemos nosso código de pipeline de dados para aceitar configuração para os caminhos de origem e destino, isso nos permite implementar e executar pipelines de dados rapidamente em uma região diferente de onde os dados estão sendo armazenados. A nuvem em que o cluster é criado é irrelevante para qual nuvem os dados são lidos ou gravados.

Existem alguns conjuntos de dados que protegemos contra a falha do serviço de armazenamento, replicando continuamente os dados entre os provedores de nuvem. Isso pode ser facilmente feito aproveitando a funcionalidade Delta deep clone, como [descrito neste blog](#). Cada vez que o comando clone é executado em uma tabela, ele atualiza o clone apenas com as mudanças incrementais desde a última vez que foi executado. Esta é uma maneira eficiente de replicar dados entre regiões e até nuvens.

## Minimizando a interrupção de pipelines de dados ao vivo

Nossos pipelines de dados são a força vital do nosso serviço gerenciado e partem de um negócio global que não dorme. Não podemos nos dar ao luxo de pausar os dutos por um longo período de tempo para manutenção, upgrades ou reabastecimento de dados. Recentemente, precisávamos bifurcar nossos pipelines para filtrar um subconjunto dos dados normalmente escritos em nossa mesa principal para serem escritos em uma nuvem pública diferente. Conseguimos fazer isso sem interromper os negócios como de costume.

Seguindo essas etapas, fomos capazes de implantar mudanças em nossa arquitetura em nosso sistema ao vivo sem causar interrupções.

Primeiro, executamos um **deep clone** da tabela principal para um novo local na outra nuvem. Isso copia os dados e o login da transação de forma a garantir consistência.

Em segundo lugar, lançamos a nova configuração em nossos pipelines para que a maioria dos dados continue sendo gravada na tabela principal central, e o subconjunto de dados grave na nova tabela clonada em nuvem diferente. Essa mudança pode ser feita facilmente apenas implantando uma nova configuração, e as tabelas recebem atualizações apenas para as novas mudanças que devem receber.

Em seguida, executamos novamente o mesmo comando de deep clone. O Delta Lake somente capturará e copiará as alterações incrementais da tabela principal original para a nova tabela clonada. Isso essencialmente preenche a nova tabela com todas as alterações nos dados entre as etapas 1 e 2.

Finalmente, o subconjunto de dados pode ser excluído da tabela principal, e a maioria dos dados pode ser excluída da tabela clonada.

Agora, ambas as tabelas representam os dados que devem conter, com histórico transacional completo, e isso foi feito ao vivo sem interromper a atualização dos pipelines.

## Resumo

A Databricks desconsidera os detalhes dos serviços individuais de nuvem, seja para executar a infraestrutura com nosso gerenciador de clusters, ingerir dados com o Auto Loader ou realizar gravações transacionais no armazenamento em nuvem com o Delta Lake. Isso nos dá uma vantagem, pois podemos usar uma única base de código para fazer a ponte entre o armazenamento nas nuvens públicas e o compute, tanto para a federação de dados quanto para a recuperação de desastres. Essa funcionalidade entre nuvens nos dá a flexibilidade de mover o compute e o armazenamento onde quer que ele seja necessário para nós e nossos clientes.

SEÇÃO

# 03

## Histórias de clientes

Atlassian

ABN AMRO

J.B. Hunt

SEÇÃO 3.1

# Atlassian

A Atlassian é uma fornecedora líder de software de colaboração, desenvolvimento e acompanhamento de problemas para equipes. Com mais de 150.000 clientes globais (incluindo 85 da Fortune 100), a Atlassian está avançando no poder da colaboração com produtos como Jira, Confluence, Bitbucket, Trello e muito mais.

## CASO DE USO

A Atlassian usa a Plataforma Databricks Lakehouse para democratizar os dados em toda a empresa e reduzir os custos operacionais. Atualmente, a Atlassian tem vários casos de uso focados em colocar a experiência do cliente em primeiro plano.

### **Atendimento ao cliente e experiência de serviço**

Com a maioria de seus clientes sendo baseados em servidor (usando produtos como Jira e Confluence), a Atlassian se propôs a levar esses clientes para a nuvem para aproveitar insights mais profundos que enriquecem a experiência de suporte ao cliente.

### **Personalização de marketing**

Os mesmos insights também podem ser usados para oferecer e-mails de marketing personalizados para impulsionar o engajamento com novos recursos e produtos.

### **Detecção de fraude e antiabuso**

Eles podem prever o abuso de licenças e o comportamento fraudulento por meio da detecção de anomalias e análise preditiva.



//  
**Na Atlassian, precisamos garantir que as equipes possam colaborar bem em todas as funções para alcançar metas em constante evolução. Uma arquitetura de lakehouse simplificada nos permitiria ingerir grandes volumes de dados do usuário e executar as funções analíticas necessárias para prever melhor as necessidades do cliente e melhorar a experiência de nossos clientes. Uma única plataforma de análise em nuvem fácil de usar nos permite melhorar rapidamente e construir novas ferramentas de colaboração baseadas em insights acionáveis.**

**Rohan Dhupelia**

Gerente sênior de plataforma de dados,  
Atlassian

**SOLUÇÃO E BENEFÍCIOS**

A Atlassian está usando a Plataforma Databricks Lakehouse para permitir a democratização de dados em escala, tanto interna quanto externamente. Eles passaram de um paradigma de armazenamento de dados para a padronização na Databricks, permitindo que a empresa se torne mais orientada para os dados em toda a organização. Mais de 3.000 usuários internos em áreas que vão desde RH e marketing até finanças e P&D – mais da metade da organização – estão acessando mensalmente insights da plataforma através de tecnologias abertas como Databricks SQL. A Atlassian também está usando a plataforma para oferecer experiências de suporte e serviço mais personalizadas para seus clientes.

- O Delta Lake sustenta um único lakehouse para PBs de dados acessados por mais de 3.000 usuários em RH, marketing, finanças, vendas, suporte e P&D
- Cargas de trabalho de BI viabilizadas pelo Databricks SQL proporcionam relatórios de painel para mais usuários
- O MLflow otimiza o MLOps para entrega mais rápida
- A unificação da plataforma de dados facilita a governança, e os clusters autogerenciados permitem a autonomia

Com uma arquitetura em escala de nuvens, maior produtividade através da colaboração entre as equipes e a capacidade de acessar todos os dados de seus clientes para análise e ML, o impacto sobre a Atlassian é projetado para ser imenso. A empresa já:

- Reduziu o custo das operações de TI (especificamente custos de compute) em 60% através da movimentação de mais de 50.000 jobs Spark de EMR para a Databricks com esforço mínimo e mudança de código baixo
- Tempo de entrega reduzido em 30% com ciclos de desenvolvimento mais curtos
- Redução de 70% nas dependências da equipe de dados com mais autoatendimento habilitado em toda a organização

**Saiba mais**

## SEÇÃO 3.2

# ABN AMRO

Como um banco estabelecido, o ABN AMRO queria modernizar seus negócios, mas foi prejudicado pela infraestrutura e por warehouses de dados legados que complicavam o acesso aos dados através de várias fontes e criavam processos de dados e fluxos de trabalho ineficientes. Hoje, o Azure Databricks habilita o ABN AMRO a democratizar dados e IA para uma equipe de mais de 500 engenheiros, cientistas e analistas capacitados que trabalham em colaboração para melhorar as operações comerciais e introduzir novos recursos de entrada no mercado em toda a empresa.

## CASO DE USO

O ABN AMRO utiliza a plataforma Databricks Lakehouse para fornecer transformação de serviços financeiros em escala global, proporcionando automatização e visão de todas as operações.

### Finanças personalizadas

O ABN AMRO aproveita dados em tempo real e insights de clientes para fornecer produtos e serviços adaptados às necessidades dos clientes. Por exemplo, eles usam o aprendizado de máquina para potencializar mensagens direcionadas em suas campanhas de marketing automatizadas para ajudar a impulsionar o engajamento e a conversão.

### Gestão de riscos

Usando a tomada de decisão baseada em dados, eles se concentram em mitigar o risco tanto para a empresa quanto para seus clientes. Por exemplo, eles geram relatórios e painéis que os líderes e tomadores de decisão internos usam para entender melhor o risco e evitar que ele afete os negócios do ABN AMRO.

### Detecção de fraudes

Com o objetivo de prevenir atividades maliciosas, eles estão usando funções analíticas preditivas para identificar fraudes antes que elas afetem seus clientes. Entre as atividades que eles estão tentando enfrentar estão lavagem de dinheiro e aplicativos de cartão de crédito falsos.



**A Databricks mudou a forma como fazemos negócios. Isso nos colocou em melhor posição para ter sucesso em nossos dados e transformação de IA como empresa, viabilizando profissionais de dados com capacidades avançadas de dados de forma controlada e escalável.**

**Stefan Groot**

Chefe de Engenharia Analítica,  
ABN AMRO

## SOLUÇÃO E BENEFÍCIOS

Hoje, o Azure Databricks capacita o ABN AMRO a democratizar os dados e a IA para uma equipe de mais de 500 engenheiros, cientistas e analistas que trabalham de forma colaborativa na melhoria das operações comerciais e na introdução de novas capacidades de entrada no mercado em toda a empresa.

- O Delta Lake permite pipelines de dados rápidos e confiáveis para alimentar dados precisos e completos para análise downstream
- A integração com o Power BI permite fácil análise SQL e alimenta insights para mais de 500 usuários de negócios por meio de relatórios e painéis
- O MLflow acelera a implantação de novos modelos que melhoram a experiência do cliente — com novos casos de uso entregues em menos de dois meses

**Tempo de lançamento no mercado**

10 vezes mais rápido — casos de uso implantados em dois meses

**Mais de 100**

casos de uso a serem entregues ao longo do próximo ano

**mais de 500**

negócios capacitados e usuários de TI



**Saiba mais**

SEÇÃO 3.2

## J.B. HUNT



O que a Databricks realmente nos forneceu é uma base para o mercado de frete digital mais inovador, nos permitindo aproveitar a IA para proporcionar a melhor experiência de transportadora possível.

**Joe Spinelle**

Diretor de Engenharia e Tecnologia,  
J.B. Hunt



Saiba mais



Em sua missão de construir a rede de transporte digital mais eficiente da América do Norte, J.B. Hunt queria racionalizar a logística de transporte de mercadorias e fornecer a melhor experiência — mas a arquitetura herdada, a falta de recursos de IA e a incapacidade de lidar com grandes dados de forma segura causaram entraves significativos. No entanto, após a implementação da Plataforma Databricks Lakehouse e Immuta, J.B. Hunt agora é capaz de fornecer soluções operacionais que vão desde a melhoria da eficiência da cadeia de suprimentos até o aumento da produtividade, resultando em economias significativas de infraestrutura de TI e ganhos de receita.

### CASO DE USO

J.B. Hunt usa a Databricks para entregar análises de transportadoras de frete líderes do setor por meio de sua plataforma Carrier 360, reduzindo os custos e aumentando a produtividade e a segurança do motorista. Os casos de uso incluem logística de frete, customer 360, personalização e muito mais.

### SOLUÇÃO E BENEFÍCIOS

A J.B. Hunt usa a Plataforma Databricks Lakehouse para construir o mercado de frete mais seguro e eficiente da América do Norte — otimizando a logística, as experiências das transportadoras e reduzindo os custos.

- O Delta Lake federa e democratiza dados para otimizações de rota em tempo real e recomendações de driver por meio da plataforma Carrier 360
- Notebooks aumentam a produtividade da equipe de dados para oferecer mais casos de uso mais rápido
- O MLflow acelera a implementação de novos modelos que melhoram a experiência do driver

**US\$ 2,7 milhões**

em economia de infraestrutura de TI, aumento da rentabilidade

**5%**

de aumento impulsionado pela melhoria logística

**recomendações  
99,8% mais  
rápidas**

para uma experiência melhor de transportadora

# Sobre a Databricks

A Databricks é a empresa de dados e IA. Mais de 5.000 organizações em todo o mundo — incluindo Comcast, Condé Nast, H&M e mais de 40% das empresas Fortune 500 — contam com a Plataforma Databricks Lakehouse para unificar seus dados, funções analíticas e IA. A Databricks está sediada em São Francisco, com escritórios em todo o mundo. Fundada pelos criadores originais do Apache Spark,™ Delta Lake e MLflow, a Databricks tem como missão ajudar as equipes de dados a resolver os problemas mais difíceis do mundo. Para saber mais, siga a Databricks no [Twitter](#), [LinkedIn](#) e [Facebook](#).

INICIE SUA AVALIAÇÃO

GRATUITA

Entre em contato conosco para obter uma demonstração personalizada [databricks.com/contact](https://databricks.com/contact)

