



João Defanti Neto  
joao.defanti@dataex.com.br  
+55 11 3446-8380  
www.dataex.com.br

FASTER  
MAKE SMARTER DECISIONS  
TABLE

# Conteúdo Programado

- Apresentação
- Conceito de ETL
- Etapas do Processo de ETL
- Por que o ETL é importante?
- Objetivos do ETL
- Diferença entre ETL e ELT
- Ferramentas de ETL
- Azure Data Factory
- Demos
- Considerações Finais e dúvidas

# Conceito

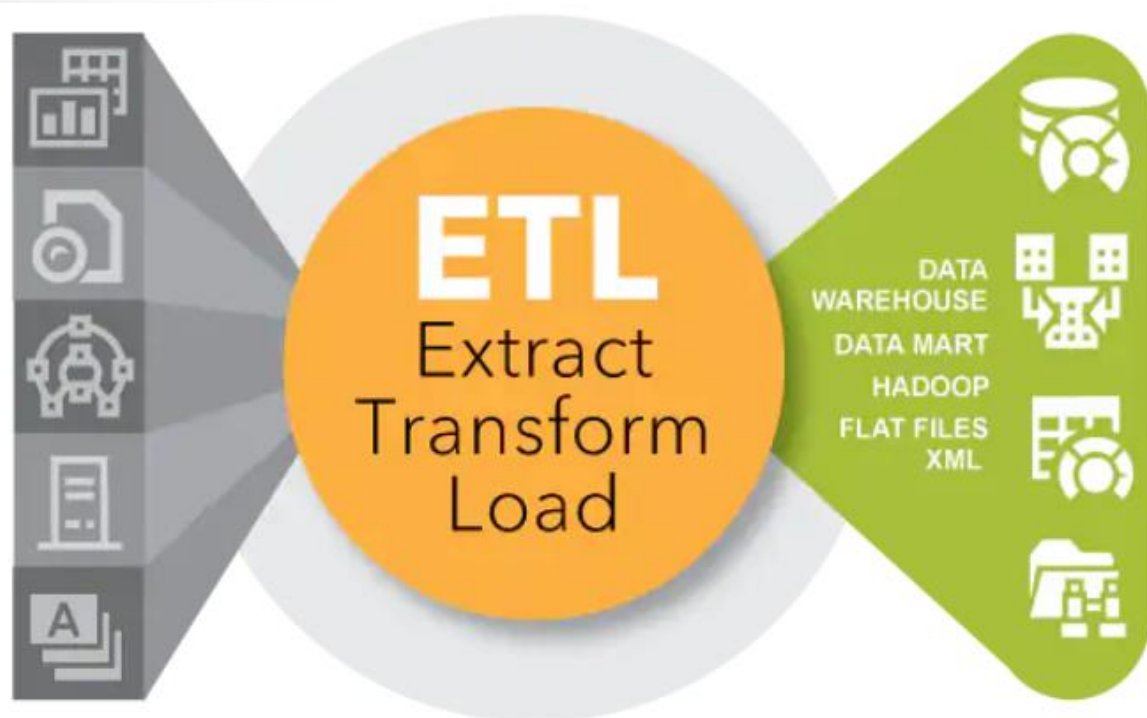
- ETL é uma sigla que se refere a um processo de integração de dados. ETL representa Extract, Transform, Load, que são as três fases desse processo.
- O ETL é utilizado em projetos de Business Intelligence (BI) e em ambientes onde a integração de dados é essencial para tomar decisões. Ele ajuda a consolidar dados de diferentes fontes, garantindo a consistência, qualidade e integridade dos dados no sistema de destino. Essa abordagem é fundamental em ambientes onde os dados são dispersos e precisam ser unificados para análises eficientes e relatórios precisos.

# Etapas do Processo

1. **Extract (Extração):** Nesta fase, os dados são coletados de diversas fontes. Isso pode incluir bancos de dados, arquivos, sistemas legados, APIs e outras origens de dados.
2. **Transform (Transformação):** Após a extração, os dados são transformados para atender aos requisitos de destino. Isso pode envolver a limpeza dos dados, a remoção de linhas duplicadas, a conversão de formatos, a aplicação de regras de negócios e outras manipulações necessárias.
3. **Load (Carga):** Na fase final, os dados transformados são carregados no sistema de destino, que pode ser um data warehouse, um banco de dados relacional, ou qualquer outro local onde os dados serão armazenados e utilizados para análises e relatórios.

# Por que o ETL é importante?

As empresas têm confiado no processo ETL por muitos anos para obter uma visão consolidada dos dados que impulsiona melhor decisões de negócios. Hoje, esse método de integração de dados de vários sistemas e fontes ainda é um componente central da caixa de ferramentas de integração de dados de uma organização.



- Quando usado com um data warehouse corporativo (dados em repouso), o ETL fornece um contexto histórico profundo para o negócio.
- Ao fornecer uma visão consolidada, o ETL facilita que os usuários de negócios analisem e informem dados relevantes para suas iniciativas.
- O ETL pode melhorar a produtividade dos profissionais de dados porque codifica e reutiliza processos que movem dados sem exigir habilidades técnicas para escrever códigos ou scripts.
- O ETL evoluiu ao longo do tempo para suportar requisitos emergentes de integração para coisas como streaming de dados. As organizações precisam tanto do ETL quanto do ELT para reunir dados, manter a precisão e fornecer a auditoria normalmente necessária para armazenamento de dados, relatórios e análises.

# Objetivos do ETL

O uso de ETL tem vários objetivos e benefícios em ambientes onde a integração de dados é essencial. Aqui estão alguns dos principais objetivos:

- **Consolidação de Dados:** O ETL permite a coleta e consolidação de dados provenientes de diversas fontes. Isso é particularmente útil em organizações onde os dados estão distribuídos em diferentes sistemas, bancos de dados ou formatos.
- **Integração de Dados:** Ao extrair dados de várias fontes e transformá-los para um formato comum, o ETL facilita a integração de informações. Isso proporciona uma visão unificada e coesa dos dados, permitindo análises mais abrangentes.
- **Qualidade dos Dados:** Durante a fase de transformação, o ETL possibilita a limpeza e a melhoria da qualidade dos dados. Isso inclui a detecção e correção de erros, a remoção de registros duplicados e a padronização de formatos.
- **Padronização e Normalização:** O ETL ajuda na padronização e normalização dos dados, garantindo que eles estejam em conformidade com as regras e padrões definidos pela organização. Isso facilita a consistência e a compreensão dos dados.

# Objetivos do ETL

- **Preparação para Análise:** Os dados transformados e carregados pelo processo ETL são geralmente otimizados para análises. Isso inclui a estruturação dos dados de maneira apropriada para consultas eficientes e relatórios significativos.
- **Aprimoramento do Desempenho:** A carga de dados em um formato otimizado pode melhorar o desempenho das consultas e relatórios. O ETL pode incluir estratégias para otimizar o armazenamento e a recuperação de dados, contribuindo para um ambiente de análise mais eficiente.
- **Suporte a Tomada de Decisão:** Ao criar um ambiente de dados consistente e confiável, o ETL fornece uma base sólida para tomada de decisões. Isso é crucial em ambientes de negócios onde decisões estratégicas dependem de análises precisas e atualizadas.
- **Auditoria e Rastreabilidade:** O ETL geralmente inclui recursos de auditoria que permitem rastrear as transformações e carregamentos de dados. Isso é importante para garantir a rastreabilidade e a conformidade com regulamentações.

Em resumo, o uso de ETL é fundamental para garantir que os dados estejam prontos para análises e relatórios, proporcionando às organizações uma base sólida para a tomada de decisões estratégicas.

# Diferenças entre ETL e ELT

As abordagens ETL (Extract, Transform, Load) e ELT (Extract, Load, Transform) referem-se a diferentes sequências de execução no processo de integração de dados.

As principais diferenças entre ETL e ELT são:

- **Sequência de fases:** No ETL a transformação ocorre antes do carregamento no destino final. Significa que os dados são extraídos da origem, passam pelas transformações e por fim são carregados no destino. No ELT, a transformação ocorre após as etapas de extração e carga.
- **Local da transformação:** No ETL, as transformações ocorrem em uma camada intermediária antes que os dados sejam carregados no destino. No ELT, as transformações já ocorrem no local de armazenamento dos dados final.



# Diferenças entre ETL e ELT

- **Flexibilidade e Escalabilidade:** O ETL pode ser mais adequado para casos em que a transformação de dados é intensiva e envolve processos complexos fora do ambiente de destino. Já o ELT tende a ser mais flexível em relação a mudanças nos requisitos de transformação e pode se beneficiar da escalabilidade do sistema de destino para lidar com grandes volumes de dados.
- **Tempo de Processamento:** Enquanto o ETL pode exigir mais tempo para a fase de transformação, especialmente se houver grandes volumes de dados a serem processados antes do carregamento, o ELT pode proporcionar um carregamento mais rápido dos dados, pois a transformação ocorre após o carregamento.

A escolha entre ETL e ELT depende das necessidades específicas do projeto, dos requisitos de desempenho, da infraestrutura disponível e de outros fatores. Em muitos casos, a escolha entre essas abordagens é influenciada pelas características dos sistemas e das ferramentas utilizadas na implementação do processo de integração de dados.

# Ferramentas de ETL

Existem várias ferramentas de ETL no mercado, oferecendo diferentes conjuntos de recursos e funcionalidades. Algumas das mais conhecidas são:

- Azure Data Factory
- Azure Synapse Analytics Pipelines
- Microsoft SQL Server Integration Services (SSIS)
- Apache NiFi
- Talend
- AWS Glue
- Google Cloud DataFlows
- Pentaho Data Integration

Trataremos hoje com o Azure Data Factory e suas funcionalidades.

# Azure Data Factory - Overview

- O Azure Data Factory (ADF) é uma solução de integração de dados totalmente gerenciada e sem servidor para ingestão, preparação e transformação de todos os seus dados em escala. Ele permite que as organizações o utilizem para uma ampla variedade de casos de uso: engenharia de dados, migração de seus pacotes SSIS locais para o Azure, integração de dados operacionais, análises, ingestão de dados em data warehouses e muito mais.
- Algumas vantagens do Data Factory:
  - **Ampla variedade de conectores para armazenamento de dados:** Permite que as organizações ingiram dados de uma ampla variedade de fontes de dados. Quer a fonte de dados seja local, multinuvem ou fornecida por fornecedores de software como serviço (SaaS).
  - **Acesso a dados locais:** Permite que as organizações se conectem a essas fontes de dados locais usando um Integration Runtime.
  - **Integração segura de dados:** Oferece suporte à integração segura de dados, conectando-se a endpoints de rede privados que são suportados por vários armazenamentos de dados do Azure.
  - **Suporte CI/CD:** Permite que qualquer desenvolvedor o use como parte de um processo de integração e entrega contínua (CI/CD). CI/CD com Azure Data Factory permite que um desenvolvedor mova ativos do Data Factory (pipelines, dataflows, linked services e muito mais) de um ambiente (desenvolvimento, teste, produção) para outro. Pronto para uso, o Azure Data Factory fornece integração nativa com Azure DevOps e GitHub.

# Azure Data Factory – Integration Runtime

- No Data Factory, uma task define a ação a ser realizada. Um linked service define um armazenamento de dados de destino ou um serviço de computação.
- Um Integration Runtime fornece a ponte entre a task e os linked services. Ele é referenciado pelo linked service ou pela task e fornece o ambiente de computação no qual a atividade é executada ou de onde é expedida.
- Desse modo, a atividade pode ser executada na região mais próxima possível do serviço de computação ou armazenamento de dados de destino, da maneira que proporciona o mais alto desempenho e atendendo às necessidades de segurança e de conformidade.

# Azure Data Factory – Linked Services

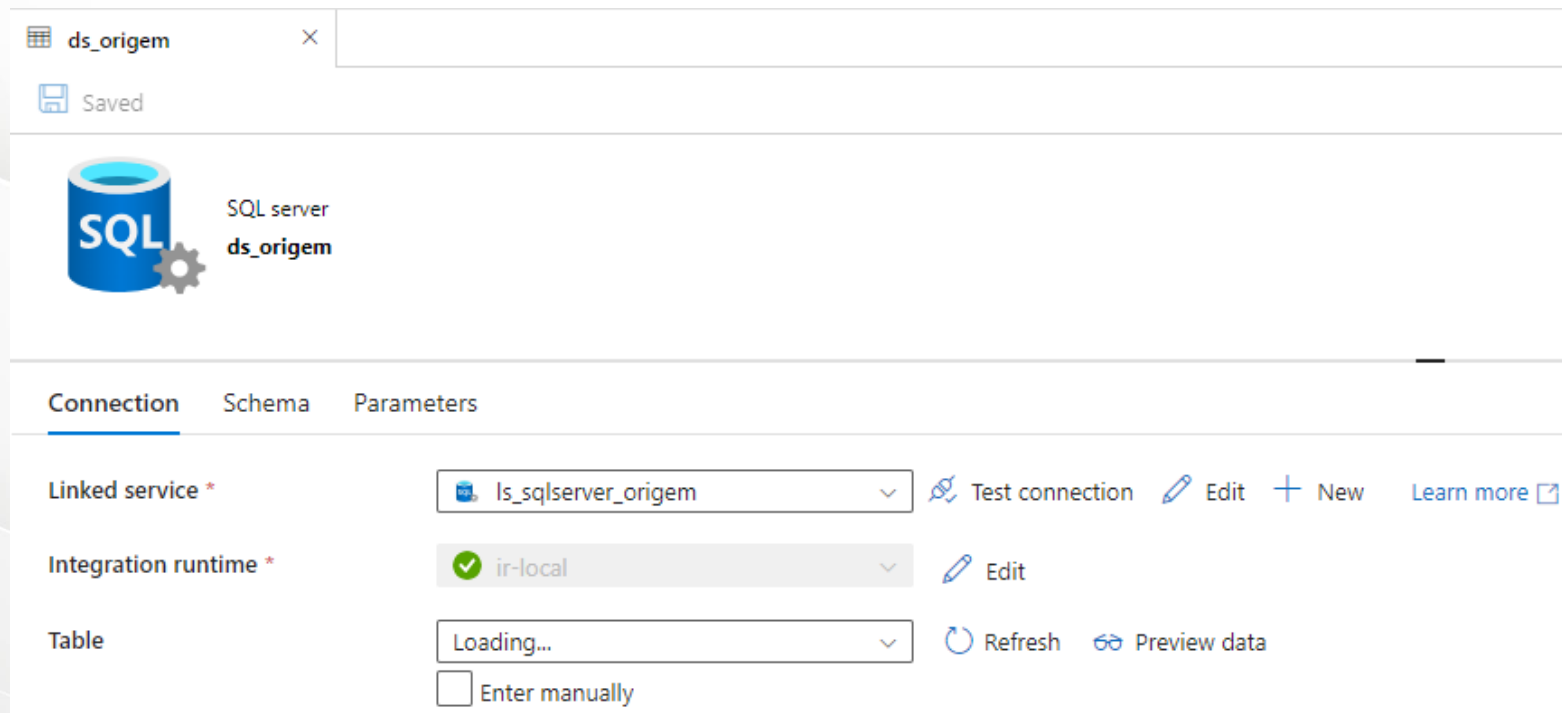
- Os linked services são como cadeias de conexão, que definem as informações de conexão necessárias para que o Data Factory se conecte aos recursos externos.
- Linked services são usados para duas finalidades no Data Factory:
  1. Para representar um armazenamento de dados que inclui, mas não está limitado a, um banco de dados do SQL Server, um banco de dados Oracle, um compartilhamento de arquivo ou uma conta de armazenamento de blobs do Azure.
  2. Para representar um recurso de computação que pode hospedar a execução de uma atividade.

The screenshot displays the 'Edit linked service' configuration page in Azure Data Factory. The title is 'Edit linked service' with a 'SQL server' icon and a 'Learn more' link. The configuration fields are as follows:

- Name \***: Is\_sqlserver\_origem
- Description**: (Empty text box)
- Connect via integration runtime \***: ir-local (with a green checkmark icon)
- Connection string** and **Azure Key Vault** tabs are visible.
- Server name \***: DESKTOP-HVTDV7U
- Database name \***: db-origem
- Authentication type**: SQL authentication
- User name \***: usr\_db\_origem
- Password** and **Azure Key Vault** tabs are visible.
- AKV linked service \***: kv\_treinamentofatec
- Secret name \***: pwd-db-origem
- Edit** checkbox is checked.
- Secret version**: Latest version
- Edit** checkbox is unchecked.

# Azure Data Factory – Datasets

- Os datasets representam as estruturas de dados nos repositórios de dados, que simplesmente apontam para ou fazem referência aos dados que você deseja usar em suas atividades como entradas ou saídas.
- Exemplo de configuração de um dataset:

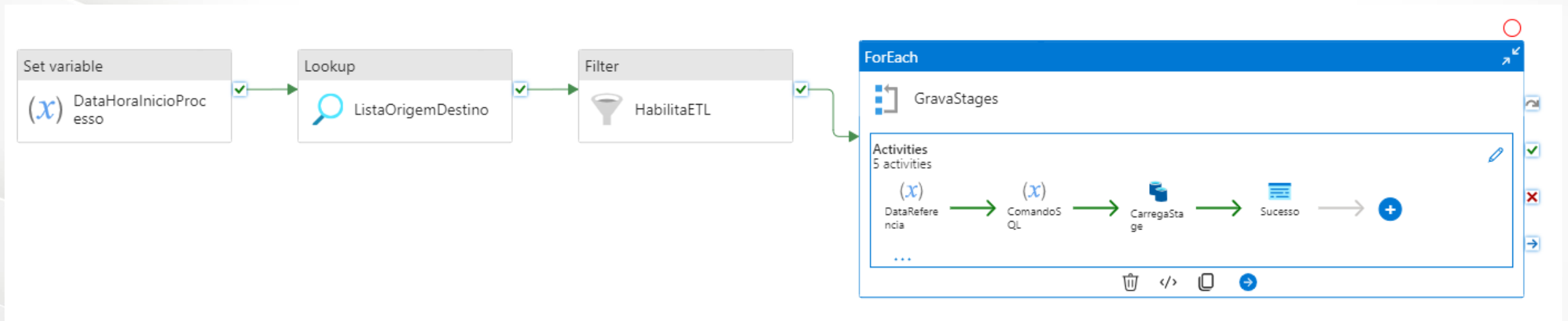


The screenshot shows the configuration page for a dataset named 'ds\_origem' in Azure Data Factory. The interface includes a top bar with the dataset name and a 'Saved' status. Below this, there is a visual representation of the dataset as a blue cylinder with 'SQL' and a gear icon, labeled 'SQL server ds\_origem'. The main configuration area has three tabs: 'Connection', 'Schema', and 'Parameters'. The 'Connection' tab is active, showing the following settings:

- Linked service \***: A dropdown menu showing 'ls\_sqlserver\_origem'. To the right are buttons for 'Test connection', 'Edit', '+ New', and 'Learn more'.
- Integration runtime \***: A dropdown menu showing 'ir-local' with a green checkmark. To the right is an 'Edit' button.
- Table**: A dropdown menu showing 'Loading...'. To the right are 'Refresh' and 'Preview data' buttons. Below this is a checkbox labeled 'Enter manually'.

# Azure Data Factory – Pipelines

- Um data factory pode ter um ou mais pipelines. Um pipeline é um agrupamento lógico de atividades que realiza uma unidade de trabalho. Juntas, as atividades em um pipeline executam uma tarefa.
- A vantagem disso é que o pipeline permite que você gerencie atividades como um conjunto, em vez de gerenciar cada uma individualmente. As atividades em um pipeline podem ser encadeadas para operarem de modo sequencial ou elas podem operar de forma independente em paralelo.
- Exemplo de pipeline:



**OBRIGADO!**

**FASTER  
MAKE SMARTER DECISIONS  
WHOLE**

