Instituto Superior de Engenharia

Politécnico de Coimbra

Integração de Dados

CTeSP Tecnologias e Programação de Sistemas de Informação (Cantanhede)

Professor: João Leal

joao.leal@isec.pt

UNIgreen The green European University

PRR Plano de Recuperação e Resiliência

REPÚBLICA PORTUGUESA

Financiado pela União Europeia NextGenerationEU

O Papel das Ferramentas no Processo ETL

- As Ferramentas de Integração de Dados são aplicações de software concebidas para automatizar e gerir o processo de Extração, Transformação e Carregamento (ETL), ou a sua variação mais recente, Extração, Carregamento e Transformação (ELT).

- Estas ferramentas são cruciais para a consolidação de dados de múltiplas fontes heterogéneas num destino unificado, como um Data Warehouse ou Data Lake.

O Papel das Ferramentas no Processo ETL

O seu papel principal é o de abstrair a complexidade da codificação manual de scripts de integração, oferecendo interfaces gráficas (GUI) para:

- **Conectividade:** Estabelecer ligações a uma vasta gama de fontes de dados (bases de dados, ficheiros, APIs, sistemas de ERP/CRM).

O Papel das Ferramentas no Processo ETL

- **Transformação:** Aplicar regras de negócio, limpeza de dados, padronização e agregação de forma visual ou através de componentes pré-construídos.

- **Orquestração:** Gerir o fluxo de trabalho, o agendamento, o tratamento de erros e a monitorização de todo o processo de integração.

Classificação das Ferramentas de Integração

- As ferramentas de integração de dados podem ser classificadas de várias formas, sendo a distinção entre Open Source e Comerciais a mais relevante para fins de custo e personalização.

Classificação das Ferramentas de Integração

Instituto Superior de Engenharia
Politécnico de Coimbra

Característica	Ferramentas Open Source (Ex: Airflow, NiFi, Pentaho Kettle)	Ferramentas Comerciais (Ex: Informatica, Talend Enterprise, SSIS)
Custo	Geralmente gratuitas, mas com custos de implementação, suporte e manutenção internos.	Requerem licenciamento pago (muitas vezes elevado), mas incluem suporte e manutenção do fornecedor.
Personalização	Altamente personalizáveis, exigindo conhecimento técnico aprofundado.	Menos flexíveis, focadas em soluções prontas a usar e configuráveis via GUI.
Suporte	Baseado na comunidade, fóruns e documentação.	Suporte profissional e garantido pelo fornecedor.
Escalabilidade	Depende da arquitetura e da experiência da equipa interna.	Geralmente escaláveis e otimizadas para ambientes empresariais de grande escala.
Curva de Aprendizagem	Pode ser mais acentuada devido à necessidade de configurar a infraestrutura.	Geralmente mais suave, com interfaces intuitivas e documentação completa.

Critérios de Seleção de uma Ferramenta ETL

- A escolha da ferramenta de integração de dados mais adequada depende de vários fatores relacionados com os requisitos de negócio, a infraestrutura tecnológica e o orçamento. Os principais critérios de seleção incluem :

- **Conectividade:** A capacidade da ferramenta de se ligar a todas as fontes e destinos de dados necessários (ex: bases de dados legadas, APIs modernas, serviços de Cloud).

Critérios de Seleção de uma Ferramenta ETL

- **Escalabilidade e Performance:** A capacidade de processar volumes crescentes de dados e de manter a performance em operações de alta frequência.

- **Facilidade de Uso (Curva de Aprendizagem):** A qualidade da interface gráfica (GUI) e a facilidade com que os programadores e analistas podem criar e gerir fluxos de trabalho.

Critérios de Seleção de uma Ferramenta ETL

- **Funcionalidades de Transformação:** A riqueza dos componentes de transformação de dados, incluindo limpeza, padronização e agregação.

- **Orquestração e Agendamento:** A capacidade de automatizar, agendar e monitorizar os processos de integração de forma robusta.

Critérios de Seleção de uma Ferramenta ETL

- **Custo Total de Propriedade (TCO):** O custo combinado de licenciamento, infraestrutura, manutenção e recursos humanos necessários.

- **Comunidade e Suporte:** A dimensão da comunidade de utilizadores (para ferramentas Open Source) ou a qualidade do suporte do fornecedor (para ferramentas comerciais).

Ferramentas ETL de Código Aberto (Open Source)

- As ferramentas de integração de dados de código aberto oferecem uma alternativa poderosa e flexível às soluções comerciais, sendo frequentemente a escolha preferencial para ambientes com restrições orçamentais ou que exigem um alto grau de personalização.

Apache NiFi

- O Apache NiFi (NiagaraFiles) é uma ferramenta de software de código aberto concebida para automatizar o fluxo de dados entre sistemas de software.

- É baseado nos princípios de programação de fluxo de dados (dataflow programming) e é ideal para gerir o movimento de dados em tempo real ou quase real.

Apache NiFi

- **Conceitos Chave:**
  - **FlowFile:** Representa um único pedaço de dados (o conteúdo) e os seus atributos (metadados). É a unidade de trabalho no NiFi.
  - **Processors:** Os blocos de construção que realizam o trabalho. Cada Processor executa uma função específica, como extrair dados de um URL, transformar o formato ou enviar para um destino.

Apache NiFi

- **Connections:** Filas que ligam os Processors, permitindo-lhes comunicar de forma assíncrona.

- **Flow Controller:** O motor que gere a execução dos Processors e o fluxo de dados.

- **Data Provenance:** O NiFi mantém um registo detalhado de todo o ciclo de vida de cada FlowFile, o que é crucial para auditoria e resolução de problemas.

Apache NiFi

- **Casos de Uso:**
  - Ingestão de dados de streaming e em tempo real (ex: logs, dados de sensores).
  - Movimentação de grandes volumes de dados entre sistemas empresariais.
  - Transformação e roteamento de dados complexos com base em regras dinâmicas.

Apache Airflow

- O Apache Airflow é uma plataforma para criar, agendar e monitorizar fluxos de trabalho (workflows) programaticamente.

- Embora não seja estritamente uma ferramenta ETL, é amplamente utilizado para orquestrar processos de integração de dados, garantindo que as tarefas sejam executadas na ordem correta e no momento certo.

Apache Airflow

- **Conceitos Chave:**
  - **DAGs (Directed Acyclic Graphs):** O coração do Airflow. Um DAG é uma coleção de todas as tarefas que se pretende executar, organizadas pelas suas dependências. O "Acyclic" significa que não pode haver ciclos (loops) no fluxo de trabalho.

Apache Airflow

- **Operators:** Classes que definem o que uma tarefa deve fazer (ex: BashOperator para executar comandos shell, PythonOperator para chamar funções Python, MySqlOperator para executar SQL).

- **Tasks:** Uma instância de um Operator num DAG.

- **Scheduler:** O componente que agendamento os DAGs e submete as tarefas para serem executadas.

Apache Airflow

- **Casos de Uso:**
  - Agendamento de processos ETL diários ou semanais.
  - Orquestração de pipelines de Machine Learning.
  - Gestão de dependências complexas entre diferentes tarefas de processamento de dados.

Nota: Pipelines: conjunto de passos automatizados que levam o software do código → à produção, de forma contínua e com o mínimo de intervenção humana)

Pentaho Data Integration (Kettle)

- O Pentaho Data Integration (PDI), também conhecido como Kettle, é uma suíte de código aberto que fornece capacidades ETL robustas e uma interface gráfica amigável para o desenvolvimento de transformações e jobs.

Pentaho Data Integration (Kettle)

- **Componentes principais:**
  - **Spoon:** A ferramenta de design gráfico onde os utilizadores criam Jobs e Transformations.
  - **Transformations:** Onde o trabalho real de ETL ocorre. Uma Transformation consiste em passos (steps) ligados por hops, que definem o fluxo de dados. Os steps incluem extração (ex: leitura de CSV), transformação (ex: limpeza, agregação) e carregamento (ex: escrita numa base de dados).

Pentaho Data Integration (Kettle)

- **Jobs:** Utilizados para orquestrar Transformations e outros Jobs, gerindo o fluxo de controlo, o tratamento de erros e a lógica de agendamento.

Pentaho Data Integration (Kettle)

- **Vantagens:**
  - Interface gráfica intuitiva (Spoon).
  - Ampla conectividade com diversas fontes de dados.
  - Ideal para processos ETL tradicionais baseados em lotes (batch).

Outras Ferramentas Open Source

- **Logstash:** Parte da Elastic Stack (ELK), é focado na ingestão, transformação e carregamento de logs e dados de eventos em tempo real.

- **Meltano:** Uma ferramenta de ELT de código aberto que se foca em extrair e carregar dados, delegando a transformação para o Data Warehouse (ELT).

Outras Ferramentas Open Source

- **dbt (data build tool):** Focado exclusivamente na fase de Transformação (T no ELT), permitindo que os engenheiros de dados escrevam transformações SQL como código com versões e possível teste.

Ferramentas ETL Comerciais e de Cloud

- As ferramentas comerciais e baseadas em cloud dominam o mercado empresarial, oferecendo soluções robustas, escaláveis e com suporte profissional garantido.

Talend Data Integration

- A Talend é uma das principais fornecedoras de soluções de integração de dados, oferecendo tanto uma versão Open Studio (gratuita e de código aberto) quanto soluções empresariais pagas.

- A sua plataforma é conhecida pela sua arquitetura unificada e pela capacidade de lidar com diversos cenários de integração, desde ETL tradicional até Big Data e integração em tempo real.

Talend Data Integration

- **Interface Gráfica Intuitiva:** Utiliza uma abordagem baseada em componentes (drag-and-drop) que facilita a criação de fluxos de trabalho complexos.

- **Ampla Conectividade:** Possui milhares de conectores pré-construídos para bases de dados, aplicações, serviços de cloud e fontes de Big Data.

Talend Data Integration

- **Administração de Dados:** As versões empresariais integram funcionalidades de qualidade de dados, gestão de metadados e administração de dados.

- **Suporte a Big Data:** Capacidade de gerar código nativo para execução em ambientes Big Data (ex: Apache Spark).

Microsoft SQL Server Integration Services (SSIS)

- O SQL Server Integration Services (SSIS) é a ferramenta ETL da Microsoft, incluída no pacote SQL Server.

- É uma plataforma robusta e madura, ideal para organizações que já utilizam o ecossistema Microsoft (SQL Server, Azure)

Microsoft SQL Server Integration Services (SSIS)

- **Ecossistema Microsoft:** É a escolha natural para a integração de dados em ambientes que dependem fortemente do SQL Server como fonte ou destino.

- **Transformações Complexas:** Oferece um ambiente rico para transformações complexas, incluindo limpeza de dados, fusão de dados e processamento de dados hierárquicos (XML, JSON).

Microsoft SQL Server Integration Services (SSIS)

- **Integração com .NET:** Permite a incorporação de código .NET personalizado para transformações que exigem lógica de programação específica.

Ferramentas de Cloud (Exemplos)

Com a migração para a cloud, surgiram ferramentas de integração nativas que tiram partido da escalabilidade e dos modelos de pagamento por utilização das plataformas cloud.

Ferramentas de Cloud (Exemplos)

Ferramenta	Plataforma Cloud	Foco Principal
AWS Glue	Amazon Web Services (AWS)	Serviço ETL serverless para Big Data e Data Lakes. Integra-se nativamente com S3 e Redshift.
Azure Data Factory (ADF)	Microsoft Azure	Serviço de orquestração e integração de dados baseado na cloud. Suporta tanto fluxos ETL como ELT.
Google Cloud Dataflow	Google Cloud Platform (GCP)	Serviço unificado para processamento de dados stream e batch, baseado no Apache Beam.

Ferramentas ELT Modernas

O paradigma ELT (Extract, Load, Transform) inverte a ordem, carregando os dados brutos primeiro para o Data Warehouse (geralmente baseado em cloud como Snowflake, BigQuery ou Redshift) e executando lá a transformação, aproveitando o poder de processamento do Data Warehouse.

Ferramentas ELT Modernas

- **Fivetran & Stitch:** São exemplos de ferramentas ELT que se especializam em automatizar a fase de Extração e Carregamento (E e L), oferecendo centenas de conectores pré-construídos para SaaS (*Software as a Service*) e bases de dados.

- A transformação (T) é frequentemente realizada utilizando SQL dentro do Data Warehouse ou com ferramentas como o dbt

Integração de Dados com Linguagens de Programação

- Embora as ferramentas ETL/ELT ofereçam soluções robustas e de baixo código, a utilização de linguagens de programação como Python e SQL continua a ser fundamental para a integração de dados.

- Especialmente em cenários que exigem alta personalização, flexibilidade ou orçamentos limitados para licenciamento de software.

Python e Pandas

- O Python, com as suas bibliotecas ricas, é a linguagem de facto para a engenharia e ciência de dados.

- A biblioteca Pandas é particularmente central para a fase de Transformação (T) e Limpeza de Dados

Vantagens do Python/Pandas

- **Extração Flexível (E):** Bibliotecas como requests permitem extrair dados de APIs web (REST, SOAP), enquanto pandas pode ler uma vasta gama de formatos de ficheiro (CSV, Excel, JSON, Parquet).

- **Transformação Poderosa (T):** O DataFrame do Pandas oferece métodos eficientes para manipulação, limpeza, agregação e fusão de dados.

Vantagens do Python/Pandas

- **Limpeza de Dados:** Funções nativas para tratamento de valores em falta (fillna, dropna), remoção de duplicados (drop_duplicates) e padronização de formatos.

- **Comunidade e Ecossistema:** A vasta comunidade e o ecossistema de bibliotecas (ex: sqlalchemy para bases de dados, pyspark para Big Data) tornam-no extremamente versátil.

SQL: Utilização de Stored Procedures e Views para Transformação

O SQL (Structured Query Language) é a linguagem essencial para a fase de Transformação em ambientes de bases de dados relacionais e, sobretudo, no paradigma ELT.

- **Views:** Uma View é uma tabela virtual cujo conteúdo é definido por uma consulta SQL. São frequentemente usadas para simplificar consultas complexas, aplicar regras de negócio e criar uma camada de abstração sobre os dados brutos, servindo como a camada de transformação (T) no Data Warehouse.

SQL: Utilização de Stored Procedures e Views para Transformação

- **Stored Procedures (SP):** Uma Stored Procedure é um conjunto de instruções SQL pré-compiladas e armazenadas no servidor de base de dados. As SPs são utilizadas para encapsular a lógica de transformação e carregamento, permitindo a execução de operações complexas de forma eficiente e repetível. São ideais para tarefas como a aplicação de regras de qualidade de dados, fusão de tabelas e carregamento incremental.

Conexão a APIs: Utilização de bibliotecas Python para extração de dados

- A extração de dados de APIs (Application Programming Interfaces) é um requisito comum na integração de dados modernos.

- APIs RESTful, em particular, são a forma mais comum de aceder a dados de serviços web (ex: redes sociais, sistemas de terceiros, dados financeiros).

Conexão a APIs: Utilização de bibliotecas Python para extração de dados

Instituto Superior de Engenharia
Politécnico de Coimbra

- A biblioteca requests em Python simplifica a interação com APIs, permitindo enviar pedidos HTTP (GET, POST, etc.) e receber dados (geralmente em formato JSON ou XML).

Conexão a APIs: Utilização de bibliotecas Python para extração de dados

- Exemplo de Extração de API:
    1. **Importar Bibliotecas:** import requests, import json.
    2. **Definir Endpoint:** URL da API a ser acedida.
    3. **Fazer Pedido:**

```
response = requests.get(endpoint, headers=auth_headers)
```

    4. **Tratar Resposta:** Verificar o código de estado HTTP (`response.status_code == 200`).
    5. **Extrair Dados:** `data = response.json()`.
    6. **Transformar (com Pandas):** Carregar os dados JSON para um DataFrame Pandas para manipulação e limpeza.

Conexão a APIs: Utilização de bibliotecas Python para extração de dados

- A combinação de Python, Pandas e requests permite construir pipelines de integração de dados altamente flexíveis e personalizadas, especialmente quando as ferramentas ETL/ELT padrão não oferecem conectores específicos ou a lógica de transformação é demasiado complexa para ser implementada graficamente.

Exemplos

- *Usando pandas para simular um pipeline simples:*

```python
import pandas as pd

# Extract
clientes = pd.read_csv("clientes_limpos.csv")

# Transform
clientes["nome"] = clientes["nome"].str.upper()

# Load
clientes.to_csv("clientes_final.csv", index=False)

print("Pipeline executado!")
```

Exemplos

- *Extração de Dados de uma API usando Python*

Simular a extração de dados de uma API pública (por exemplo, dados de câmbio) e a sua transformação inicial com Pandas.

Objetivo: Extrair a taxa de câmbio atual de USD para EUR de uma API simulada e carregar para um DataFrame.

Exemplos


```python
import requests
import pandas as pd
from datetime import datetime

# URL da API simulada de taxas de câmbio
API_URL = "https://api.exchangerate-api.com/v4/latest/USD"
```

(...)

Exemplos

(...)

```python
def extract_transform_exchange_rate(api_url):
    """
    Extrai dados de uma API, transforma a taxa de câmbio e retorna num DataFrame.
    """
    try:
        # 1. Extração (E)
        response = requests.get(api_url)
        response.raise_for_status() # Lança exceção para códigos de status HTTP 4xx/5xx
        data = response.json()

        # 2. Transformação (T)
        base_currency = data.get('base')
        eur_rate = data.get('rates', {}).get('EUR')
        timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')

        if not eur_rate:
            print("Erro: Taxa de EUR não encontrada na resposta da API.")
            return pd.DataFrame()

        # Criação do DataFrame
        df = pd.DataFrame({
            'DataHora_Extracao': [timestamp],
            'Moeda_Base': [base_currency],
            'Moeda_Alvo': ['EUR'],
            'Taxa_Cambio': [eur_rate]
        })
```

(...)

Exemplos

(...)

```
        # Transformação adicional: Arredondar a taxa para 4 casas decimais
        df['Taxa_Cambio'] = df['Taxa_Cambio'].round(4)

        return df

    except requests.exceptions.RequestException as e:
        print(f"Erro na extração da API: {e}")
        return pd.DataFrame()
    except Exception as e:
        print(f"Ocorreu um erro: {e}")
        return pd.DataFrame()
```

(...)

Exemplos

(...)

```
# Execução do pipeline
df_taxa = extract_transform_exchange_rate(API_URL)

if not df_taxa.empty:
    print("\n--- Resultado da Extração e Transformação (DataFrame) ---")
    print(df_taxa)

    # 3. Carregamento (L) - Conceitual
    # Na prática, este DataFrame seria carregado para um Data Warehouse ou Base de Dados
    # Ex: df_taxa.to_sql('taxas_cambio', engine, if_exists='append', index=False)
    print("\n[L] Carregamento: O DataFrame seria carregado para o sistema de destino.")
else:
    print("\nNão foi possível obter os dados.")
```