



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

Fundamentos de Integração de Dados

- A **integração de dados** é o processo de reunir informações provenientes de diferentes fontes e apresentá-las de forma coerente e unificada.
- É essencial em ambientes empresariais onde existem múltiplos sistemas (ERP, CRM, bases de dados locais, aplicações web) que armazenam dados sobre os mesmos objetos (clientes, produtos, vendas).

Fundamentos de Integração de Dados

- **Problema clássico:** um cliente pode estar registado no sistema de faturação com o campo “NIF”, no sistema de marketing como “Número de Contribuinte” e na loja online apenas como “Tax ID”.
- **Objetivo da integração:** disponibilizar uma versão única e fiável da verdade (*single source of truth*).

Fundamentos de Integração de Dados

- A interoperabilidade é a capacidade de diferentes sistemas comunicarem e trocarem dados:
 - **Sintática** → quando usam o mesmo formato (ex.: CSV, XML).
 - **Semântica** → quando interpretam os dados da mesma forma (ex.: “Customer ID” = “ID_Cliente”).
- A integração e a interoperabilidade são **fundamentais para evitar redundância, inconsistência e perda de informação.**

Dados Estruturados vs. Não Estruturados

- Os dados podem ser classificados conforme a sua estrutura:
 - **Estruturados**
 - **Semi-estruturados**
 - **Não estruturados**
- Na integração, o desafio é normalizar e transformar estes diferentes tipos de dados para que possam ser usados em conjunto.

Dados Estruturados vs. Não Estruturados

- **Estruturados:** organizados em tabelas (linhas e colunas), facilmente processáveis por bases de dados relacionais.

Exemplos: registos de vendas, listas de alunos, inventários.

Formatos: SQL, CSV, Excel.

Dados Estruturados vs. Não Estruturados

- **Semi-estruturados:** não seguem um esquema fixo, mas têm organização interna.

Exemplos: XML, JSON, logs de sistemas.

São comuns na integração moderna, pois transportam dados entre aplicações.

Dados Estruturados vs. Não Estruturados

- **Não estruturados:** não possuem modelo fixo; exigem técnicas avançadas para análise.

Exemplos: imagens, vídeos, documentos em PDF, e-mails.

Arquiteturas de Integração (Batch, Streaming, APIs)

- Uma arquitetura de integração define o modelo e a infraestrutura para conectar diferentes aplicações e fontes de dados.
- A escolha da arquitetura depende dos requisitos de negócio, como o volume de dados, a latência (tempo real vs. agendado) e a complexidade dos sistemas.

Arquiteturas de Integração (Batch, Streaming, APIs)

- Existem várias formas de integrar dados:
 - Integração em Batch
 - Integração em Streaming (tempo real)
 - Integração via APIs/Serviços Web

Nota: Atualmente, muitas soluções são híbridas: combinam batch (para históricos) com streaming (para eventos em tempo real).

Arquiteturas de Integração (Batch, Streaming, APIs)

Batch (em Lote):

- **Descrição:** Os dados são recolhidos, processados e movidos em grandes lotes, de forma agendada (ex: todas as noites). É uma das abordagens mais tradicionais.
 - **Tecnologia Comum:** Processos ETL (Extract, Transform, Load).
 - **Casos de Uso:** Atualização noturna de um Data Warehouse, processamento de folhas de pagamento, geração de relatórios mensais.
 - **Vantagens:** Ideal para grandes volumes de dados, fiável e menos complexo de gerir para processos não críticos em tempo real.
 - **Desvantagens:** Alta latência (os dados não estão disponíveis em tempo real).
-

Arquiteturas de Integração (Batch, Streaming, APIs)

Streaming (em Tempo Real):

- **Descrição:** Os dados são processados e movidos continuamente, evento a evento ou em micro-lotes, assim que são gerados.
 - **Tecnologia Comum:** Plataformas de streaming como Apache Kafka, AWS Kinesis, Azure Event Hubs.
 - **Casos de Uso:** Detecção de fraude em transações financeiras, monitorização de sensores IoT (Internet of Things), análise de cliques em websites em tempo real.
 - **Vantagens:** Baixa latência, permite a tomada de decisão imediata.
 - **Desvantagens:** Mais complexo de implementar e gerir, pode ser mais caro.
-

Arquiteturas de Integração (Batch, Streaming, APIs)

Serviços Web / APIs:

- **Descrição:** A integração é feita através de chamadas a serviços web ou APIs (Application Programming Interfaces). Um sistema expõe funcionalidades e dados que outros sistemas podem consumir através de pedidos (requests).
- **Tecnologia Comum:** APIs REST (baseadas em HTTP), SOAP, gRPC, GraphQL.
- **Casos de Uso:** Uma aplicação móvel que consulta o stock de um produto, um site de reservas de voos que se conecta a várias companhias aéreas.
- **Vantagens:** Promove o desacoplamento entre sistemas, é flexível, escalável e reutilizável.
- **Desvantagens:** Requer uma boa governação de APIs (segurança, versionamento, documentação).