

# Complementos de Bases de Dados

José Mendes 107188

2023



universidade  
de aveiro

---

# 1 Evolução dos Sistemas de Base de Dados

**Sistemas de Dados** - Cada vez mais as aplicações de hoje em dia são Data-Intensive, em vez de Compute-Intensive.

Para Data-Intensive, o poder bruto da CPU deixa de ser um fator limitante quando comparado com a **quantidade, complexidade e velocidade de atualização** dos dados.

De forma a otimizar a sua performance, um sistema de dados tipicamente oferece as seguintes funcionalidades:

1. **Bases de Dados** - armazenam os dados para utilização futura;
2. **Caches** - guardam os resultados de operações dispendiosas, de forma a tornar a leitura mais rápida;
3. **Search Indexes** - permitem aos utilizadores procurarem por palavras-chave ou filtrar os dados;
4. **Message Queues** - permitem a comunicação assíncrona entre processos;
5. **Stream Processing** - permite o processamento de dados em tempo real;
6. **Batch Processing** - permite o processamento de dados acumulados, periodicamente;

**Exemplo:** Um exemplo de **stream processing** ocorre na banca. Sempre que é realizada uma transação, os dados da mesma são imediatamente processados de forma a que o saldo esteja sempre atualizado.

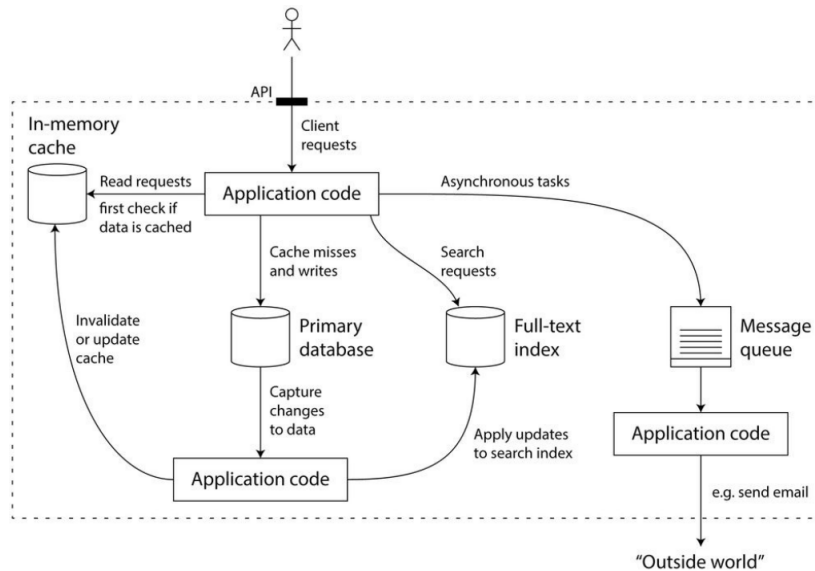
O **batch processing** é visível na faturação dos serviços pós-pagos pelas operadoras de telecomunicações. No final de cada mês, é feita uma consulta às suas bases de dados de forma a identificar todos os consumos do cliente, que são somados e depois gerada a fatura.

No **stream** os dados são processados antes de armazenados, enquanto que no **batch** são processados depois de armazenados.

Cada vez mais as aplicações requerem um **maior wide-range de requisitos**. Muitas das vezes, uma única ferramenta já não consegue satisfazer todas as necessidades de **data processing** e **storage**.

Em vez disso, o trabalho é partido em tasks que possam ser realizadas de forma eficiente por uma única ferramenta. As ferramentas individuais utilizadas são depois juntas utilizando código de aplicação.

**Exemplo:** Podemos ter uma aplicação que utiliza uma Catching Layer (**memcached**), um Full-Text Search (**Elasticsearch**) e uma Base de Dados principal separada (**MySQL**).



## 1.1 Desafios que os Sistemas de Dados enfrentam

Como garantir que todos os dados se mantêm corretos e consistentes, mesmo quando, internamente, ocorreu algum erro? (ex: persistência de dados)

Como fornecer boa performance para os clientes, mesmo quando partes do sistema estão degradadas?

Como escalar o sistema para ser capaz de aguentar uma load intensiva de trabalho?

Qual a aparência de uma boa API para o serviço?

## 1.2 Alguns Requisitos

**Fiabilidade** - O Sistema deve continuar a funcionar corretamente em caso de adversidades (ex: falhas de hardware, software ou mesmo humanas).

**Escalabilidade** - O Sistema deve ser capaz de responder ao crescimento seja do volume de dados, do tráfego, ou mesmo da complexidade.

**Manutenibilidade** - Deve ser possível que o Sistema sofra alterações ao longo do tempo por várias pessoas diferentes de forma produtiva.

---

### 1.3 Bases de Dados

São definidas como um conjunto de dados relacionados entre si e a sua organização. Dividem-se em vários tipos, sendo atualmente os mais comuns: **Relacionais**, seguidas por **Documentais**, **Motores de busca**, **Chave-Valor**, entre outras. O controlo às bases de dados é realizado por **Sistemas de Gestão de Base de Dados (SGBD** ou DBMS em inglês). Estes fornecem funções que permitem a manipulação de grandes quantidades de informação.