



iscte

INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA



emprego  
digital



# Bases de dados

Introdução aos sistemas de gestão de bases de dados

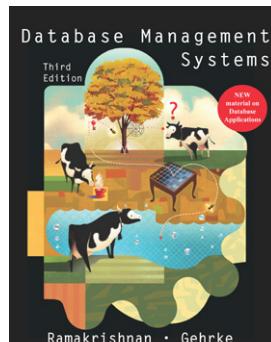
# Sumário e Referências

## Sumário

- Importância dos sistemas de informação
- Bases de dados e SGBDs
  - Razões de utilização
  - Vantagens e desvantagens
- Modelos de dados dos SGBDs
  - Independência dos dados
  - Evolução histórica
- Interrogações e gestão de transações
- Componentes de um SGBD
  - Vertentes dos SGBDs
  - Utilizadores típicos

## Referências

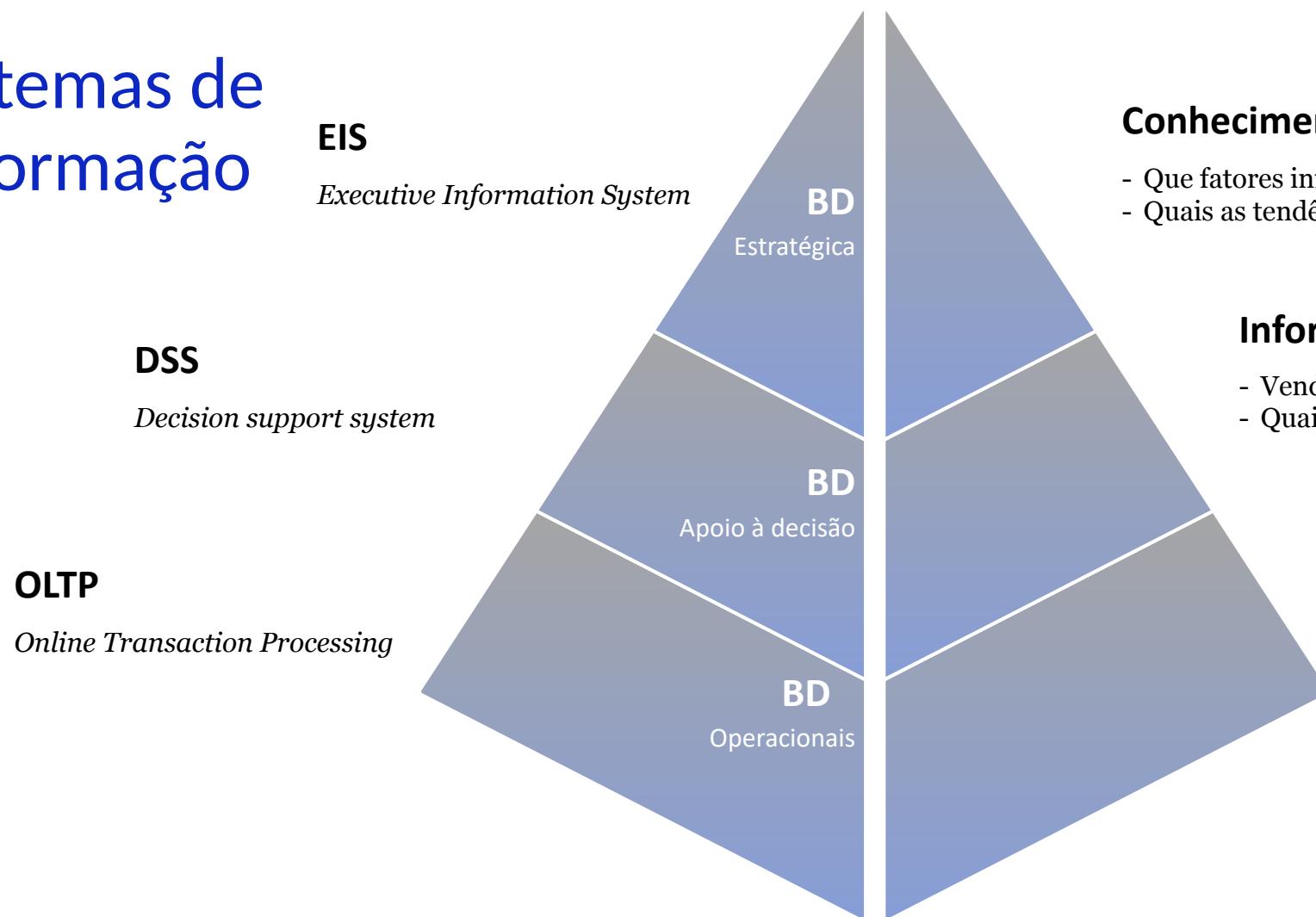
- R. Ramakrishnan ([capítulo 1](#))



# Importância dos Sistemas de Informação

- Na década de 1990, as organizações dão-se conta que a informação é um **“recurso organizacional”**
- O valor da informação depende da sua correcção e disponibilidade, apenas possível com uma gestão adequada
- As tecnologias de bases de dados são uma ferramenta essencial à gestão eficaz do recurso informação
- Diferença entre **dados** e **informação**
  - **Dados** representam factos do mundo real em bruto  
ex. respostas a inquérito
  - **Informação** resulta do processamento dos dados, permitindo interpretação e suporte a decisões  
ex. respostas ao inquérito depois de processadas

# Sistemas de Informação



## Conhecimento

- Que fatores influenciam vendas em Paris
- Quais as tendências significativas

## Informação

- Vendas mensais por produtos em Paris
- Quais são as tendências

## Dados

- Qual o preço do produto?

# Bases de dados vs. SGBDs

- **Bases de dados (BD)**

- Coleção de dados, organizados e interligados
- Modela o Negócio, uma organização ou um universo do discurso
- BD de grandes dimensões:
  - Exemplos
    - Em 2005: BD da Yahoo! Com 100 Tbytes  $\approx 10^{14}$  bytes
    - BD da MySpace com 1 Pbyte  $\approx 1000$  Tbytes  $\approx 10^{15}$  bytes

- **SGBD (Sistema de Gestão de Base de Dados)**

- DBMS (Database Management System)
- SW para gerir o armazenamento e utilização de BDs
- Exemplos de vendedores: *Oracle, IBM, Microsoft, Teradata, Sybase, ...*

- **Bases de Dados Relacionais** Classe dominante de SGBDs

# Razões de Uso de SGBDs

- Suportam conceitos úteis para a gestão de BDs
  - Linguagem para criar, alterar, e consultar/analisar dados
  - Restrições para manter a integridade dos dados (ex. idade > 0)
  - Utilizadores e privilégios de acesso aos dados
  - Transações: programas que transformam o estado da BD
- Oferecem **mecanismos automáticos** de gestão de BDs
  - Aplicação de restrições de integridade aquando de alterações nos dados
  - Controlo de acesso à BD por vários utilizadores em simultâneo
  - Recuperação de faltas (ex. falta de energia elétrica)
  - Otimização do acesso aos dados, para respostas rápidas
- Suportam **grandes quantidades de dados e de transações**
  - Mecanismos desenhados para permitir expansão de capacidade

# Sistemas Baseados em Ficheiros vs BD

**Não podem os dados ser guardados em Ficheiros?**

- Problemas típicos

- Redundância
- Inconsistência dos dados
- Maior esforço de introdução dos dados
- Falta de flexibilidade
- Partilha de dados limitada
- Manutenção difícil
- Dificuldade de Normalização  
*(Nomes, formatos, restrições de acesso)*

# SGBD Vantagens - I

- Independência dos dados

- Aplicações não estão expostas aos detalhes de como os dados estão representados e armazenados
- SGBD disponibiliza uma visão abstrata dos dados

- Acesso Eficiente aos Dados

- O SGBD utiliza uma variedade de técnicas sofisticadas para armazenar e recolher dados de uma forma eficiente

# SGBD Vantagens - II

- Integridade dos Dados e Segurança
  - O SGBD pode aplicar restrições de integridade durante o acesso aos dados
- Administração dos Dados
  - Profissionais experientes podem organizar a representação dos dados, por forma a minimizar a redundância e melhorar o armazenamento e recolha dos dados

# SGBD Vantagens - III

- Acesso Concorrente e Recuperação de Falhas
  - Acesso aos dados como se fosse acedido por um utilizador de cada vez
  - Minimiza os efeitos de falhas no sistema
- Redução do Tempo de Desenvolvimento de Aplicações
  - Disponibiliza funções de acesso comuns
  - Interface de alto nível para os dados
  - Mais robusto:
    - Tarefas executadas pelo DBMS não precisam de ser verificadas

# SGBD Desvantagens

- **Aplicações complexas** de software
- **Desempenho inaceitável** para algumas aplicações:
  - Aplicações de **tempo-real**
- **Não** disponibiliza análise **flexível** de dados **texto**
- **Nem sempre** os benefícios dos SGBD são **necessários**

# Quando não usar um SGBD

- Custos iniciais demasiado elevados
  - Investimentos em hardware, software e formação
  - Necessidade de definição e aplicação de políticas de segurança
- Bases de dados simples e imutáveis (\*)
- Requisitos de tempo-real incontornáveis (\*)
  - SGBDs não garantem execução num intervalo de tempo fixo
- Custo inicial do software pode ser reduzido
  - Open source: MySQL, MariaDB, PostgreSQL, BerkeleyDB, ...
  - Versões grátis de produtos: Oracle Express, SQL Server Express, ...

(\*) uso de ficheiros – pode ser suficiente

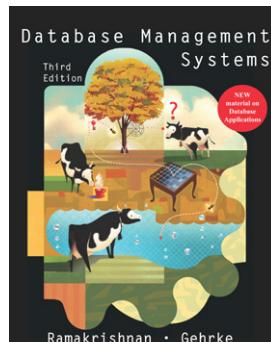
# Sumário e Referências

## Sumário

- Importância dos sistemas de informação
- Bases de dados e SGBDs
  - Razões de utilização
  - Vantagens e desvantagens
- **Modelos de dados dos SGBDs**
  - Independência dos dados
  - Evolução histórica
- Interrogações e gestão de transações
- Componentes de um SGBD
  - Vertentes dos SGBDs
  - Utilizadores típicos

## Referências

- R. Ramakrishnan ([capítulo 1](#))



# Modelo de Dados e Esquema

- Modelo de dados
  - Colecção de construtores para descrever os dados a nível abstracto
  - Esconde detalhes de armazenamento
  - Define os dados que serão armazenados pelo SGBD
- O modelo de dados está mais perto da estrutura de armazenamento do SGBD do que da visão do utilizador
- A descrição dos dados em termos de um modelo de dados é chamado de **esquema** (schema)

sid	name	Login	age	GPA
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53832	Madayan	madayan@music	11	1.8
53831	Guldu	guldu@music	12	2.0

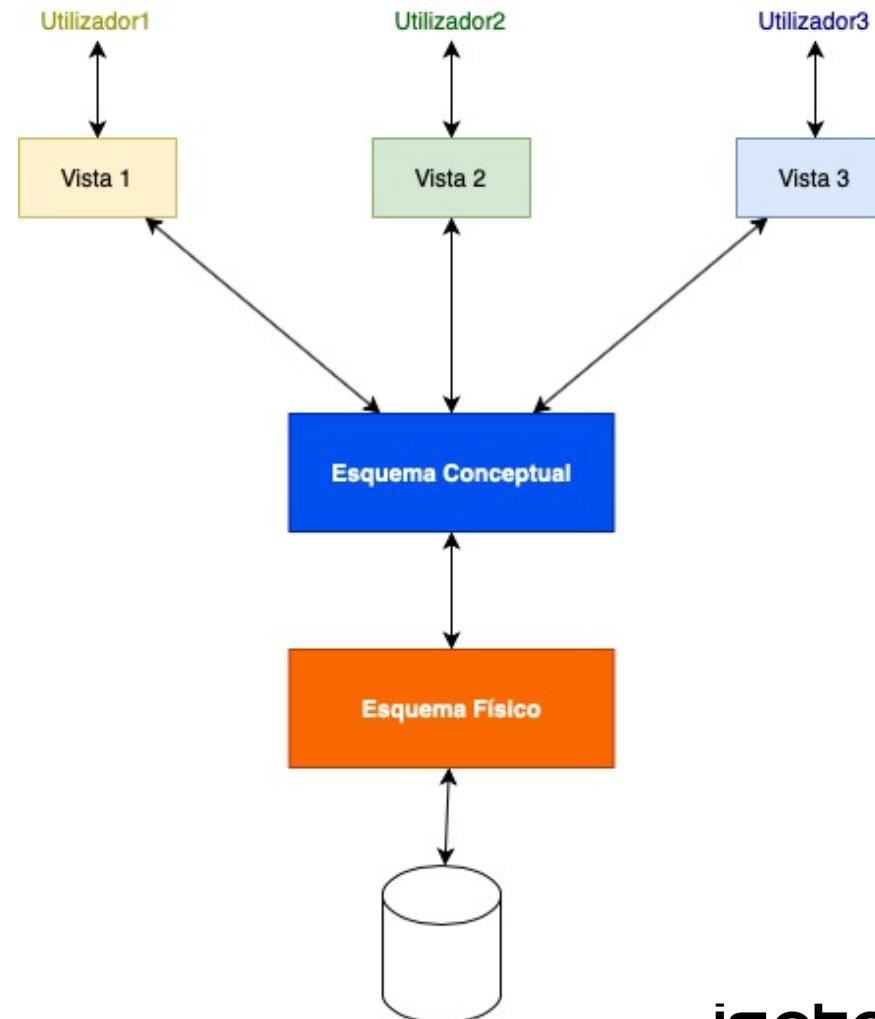
# Modelo Relacional

- O Modelo mais usado na atualidade
  - Conceito fundamental é a **Relação**
    - **Relação** = tabela com **colunas** (atributos, campos) e **linhas** (registos)
- **O esquema relational**
  - Descreve o nome das **relações** e suas **colunas**
  - *Exemplo:* Students ( sid: string, name: string, login: string, age: integer, gpa: real)
    - Cada linha na relação Students é um **registro** que descreve um aluno
    - Cada linha segue o **esquema da relação** Students
- Inclui também **regras de integridade**, que são condições que os registos de uma relação têm de satisfazer
  - *Exemplo:* Students.age >  $\emptyset$

# Níveis de Abstração num SGBD

Os dados num SGBD são descritos em 3 níveis de abstração

- **Esquema Externo**
  - Permite particularizar o acesso a cada tipo de utilizador
  - **Vistas**
    - Ex. Courseinfo ( cid: string, fname: string, enrollment: integer)
- **Esquema Conceptual** (ou lógico)
  - Dados em termos do modelo de dados do SGBD (ex. relacional)
- **Esquema Físico**
  - Dados armazenados em disco



# Independência dos Dados

- Uma das principais vantagens dos SGBDs
  - Possibilitada pelos níveis de abstração

## 1. Independência do esquema lógico

- Vistas sobre os dados podem ser preservadas mesmo que as tabelas subjacentes sejam alteradas
- Aplicações dos utilizadores podem manter-se funcionais
  - Sem necessidade de manutenção do código

## 2. Independência do esquema físico

- Tabelas podem continuar acessíveis mesmo que dados mudem de disco
  - Sem necessidade de manutenção do código
- Índices podem ser criados posteriormente e usados automaticamente
  - Para aplicações responderem mais rapidamente, sem manutenção do código

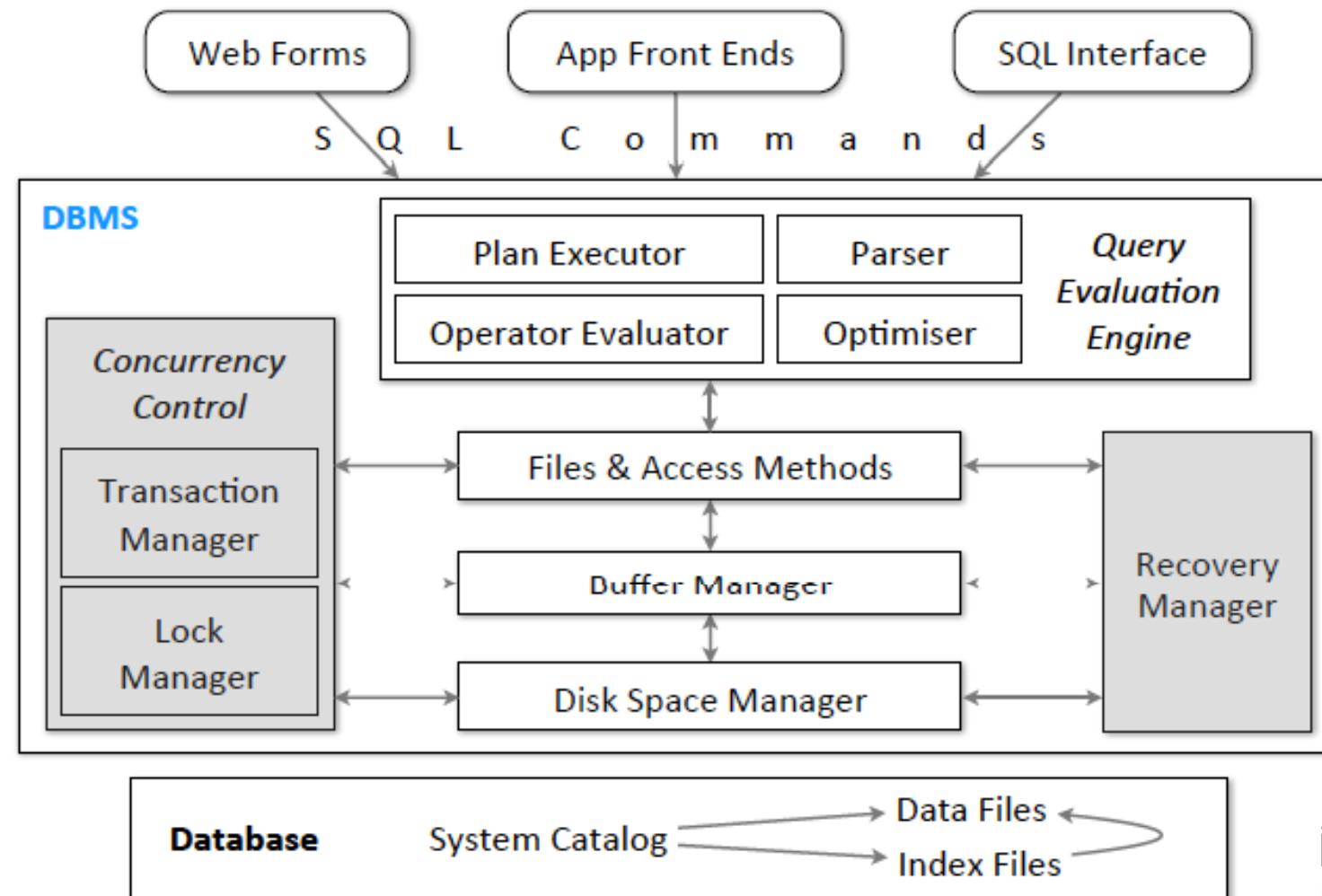
# Modelo Semântico dos Dados

- Modelo relacional suportado diretamente pelos SGBDs relacionais
  - SGBDs têm comandos para criar tabelas e gerir registos de dados
  - Ex. CREATE TABLE student (...), INSERT INTO student (...)
- Contudo, modelo relacional nem sempre é o mais adequado
  - A primeira abordagem de modelação deve ser mais abstrata
- Modelo semântico dos dados
  - Modelo mais rico com construtores mais ricos para descrever a realidade
  - Modelo mais perto da visão do utilizador
  - No final é traduzido para um modelo de dados
- Modelo Entidade–Associação (EA) Entity–Relationship (ER) model
  - Descreve de uma forma gráfica as entidades e as relações entre elas (mas, existem outros modelos)
  - Ponto de partida da modelação de dados em BD

# Evolução da história dos SGBDs

- 1<sup>a</sup> geração: anos 1960
  - Modelos de dados **hierárquico** e **em rede**
  - Algumas aplicações ainda a funcionar
  - Ex. SABRE sobre o SGBD IMS da IBM, para reserva de bilhetes de avião
- 2<sup>a</sup> geração: anos 1970
  - Edgar Codd propõem o modelo relacional
  - Inicialmente recusado pela IBM para proteger o IMS
  - Pressão dos clientes levou a IBM a desenvolver o System R
  - Mas a Oracle (então SDL) adiantou-se e lançou o Oracle v2 em 1979
  - Modelo relacional começa a massificar-se nos anos 1980
- 3<sup>a</sup> geração: anos 1990
  - Modelo relacional por objetos e adição de camadas aplicacionais ao SGBD
  - SGBDs entram na era da Internet e permitem acessos via Web

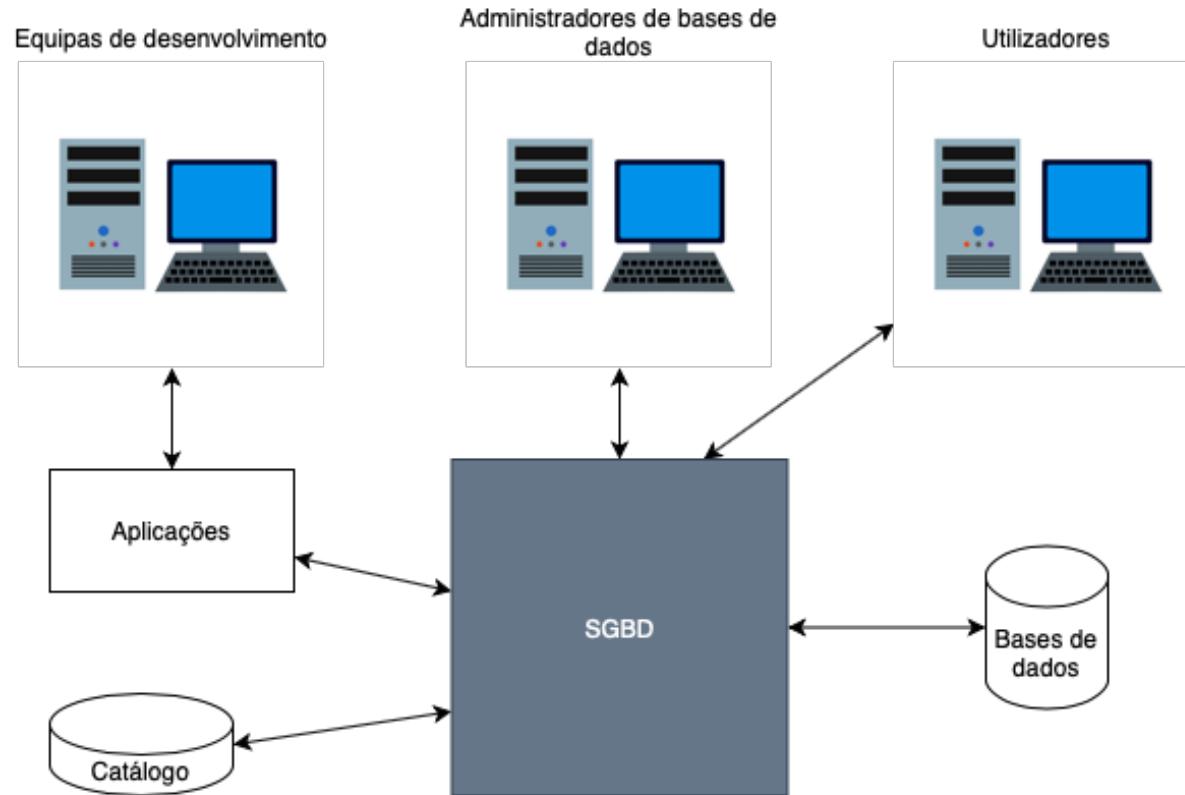
# Componentes de um SGBD



# Componentes de um SGBD

- Gestor de transações (transaction manager)
  - Cria, acompanha, e termina transações (ex. transferência de dinheiro)
  - **Orquestra serviços que detêm dados necessários à transação**
  - Pede locks para acesso aos dados, de acordo com um protocolo
- Gestor de locks (lock manager)
  - Controla o acesso concorrente aos dados
  - Concede, assim que disponíveis, locks para leitura ou escrita de dados
  - Locks podem ser exclusivos a uma transação ou partilhados por várias
- Gestor de recuperação (recovery manager)
  - Regista ações feitas pelas transações num ficheiro de log
  - **Repõe a base de dados num estado coerente em caso de crash**
  - Tem de sobreviver a variedade de faltas no software e no hardware

# Desenvolvimento e Uso de SGBDs



**Catálogo:** Contém a definição da BD. Ex: Num SGBD relacional contém definições das tabelas, vistas, regras de integridade etc.

# Usos dos SGBDs

- **OLAP (OnLine Analytical Processing)**
  - Bases de dados analíticas
  - Poucos utilizadores de cada vez
  - Transações longas e combinando muitos
  - Data warehousing e data-mining
  - Ex. Estudos de tendências e à decisão
- **OLTP (OnLine Transaction Processing)**
  - Bases de dados operacionais
  - Muitos utilizadores em simultâneo
  - Transações curtas com manipulação de poucos dados
  - Ex. vendas de produtos e serviços ao consumidor

# Utilizadores dos SGBDs

- Empresas que desenvolvem os SGBDs (DB implementors)
- Programadores (DB application programmers)
  - Cram pacotes de software que permitem aos utilizadores aceder aos dados
  - Lidam com o esquema externo (vistas sobre os dados)
- Utilizadores de aplicações (end users)
  - Indústrias, serviços, fora ou dentro da Web
  - Fazem entrar dados na organização e analisam dados
- Administradores de bases de dados (DB Administrator - DBAs)
  - Manipulam esquemas físico e lógico (ex. para afinação)
  - Cram utilizadores e definem privilégios de acesso aos dados
  - Garantem disponibilidade e recuperação em caso de faltas



O futuro profissional  
começa aqui

