



# Banco de Dados

---

# Por que estudar BD?

---

- Os Bancos de Dados fazem parte do nosso dia-a-dia:
  - operação bancária
  - reserva de hotel
  - matrícula em uma disciplina da universidade
  - cadastro na vídeo locadora

# Conceitos básicos

---

- **Dado**: fato do mundo real que está registrado
  - exemplos: endereço, data
- **Informação**: fato útil que pode ser extraído direta ou indiretamente a partir dos dados
  - exemplos: endereço de entrega, idade
- **Banco de Dados (BD)**: coleção de dados inter-relacionados e persistentes que representa um subconjunto dos fatos presentes em um domínio de aplicação(universo de discurso)

# Por que usar BD?

Considere o contexto de uma grande organização que NÃO utiliza BD

- exemplo: domínio da Universidade
  - várias divisões gerenciais (com suas aplicações)
  - grande volume de dados
  - aplicações manipulam dados comuns

## Acadêmica

Alunos  
Professores  
Disciplinas  
Turmas  
Salas

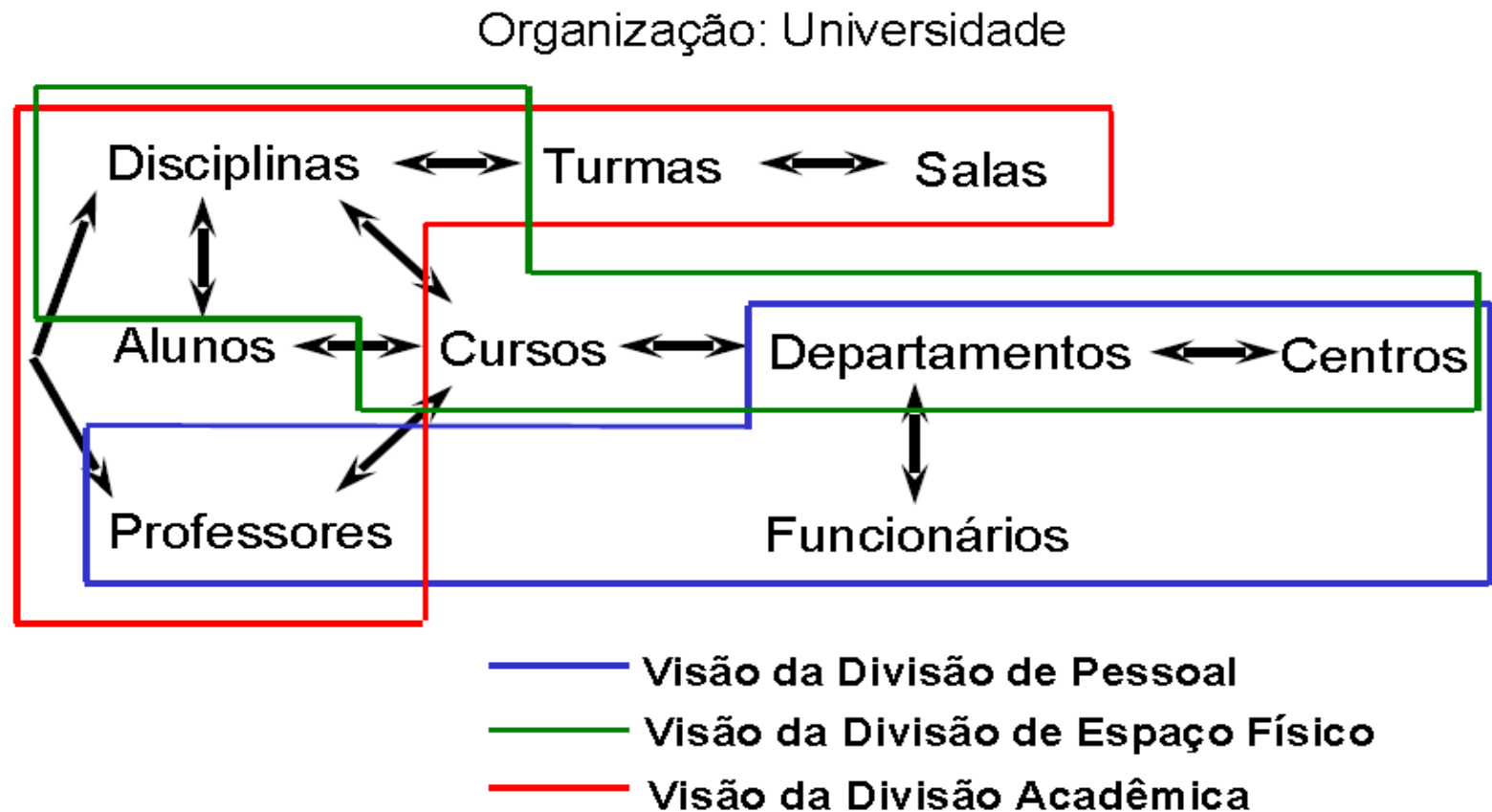
## Espaço Físico

Centros  
Departamentos  
Cursos  
Disciplinas

## Pessoal

Centros  
Departamentos  
Professores  
Funcionários

# Exemplo de um BD



# Banco de dados

---

- Banco de dados = instância de dado + meta-dados
  - ✓ Instância de dado
    - Dado propriamente
  - ✓ Meta-dados
    - *Dicionário de dados*
      - Esquema da base de dados
      - Acessado através de linguagens de definição de dados

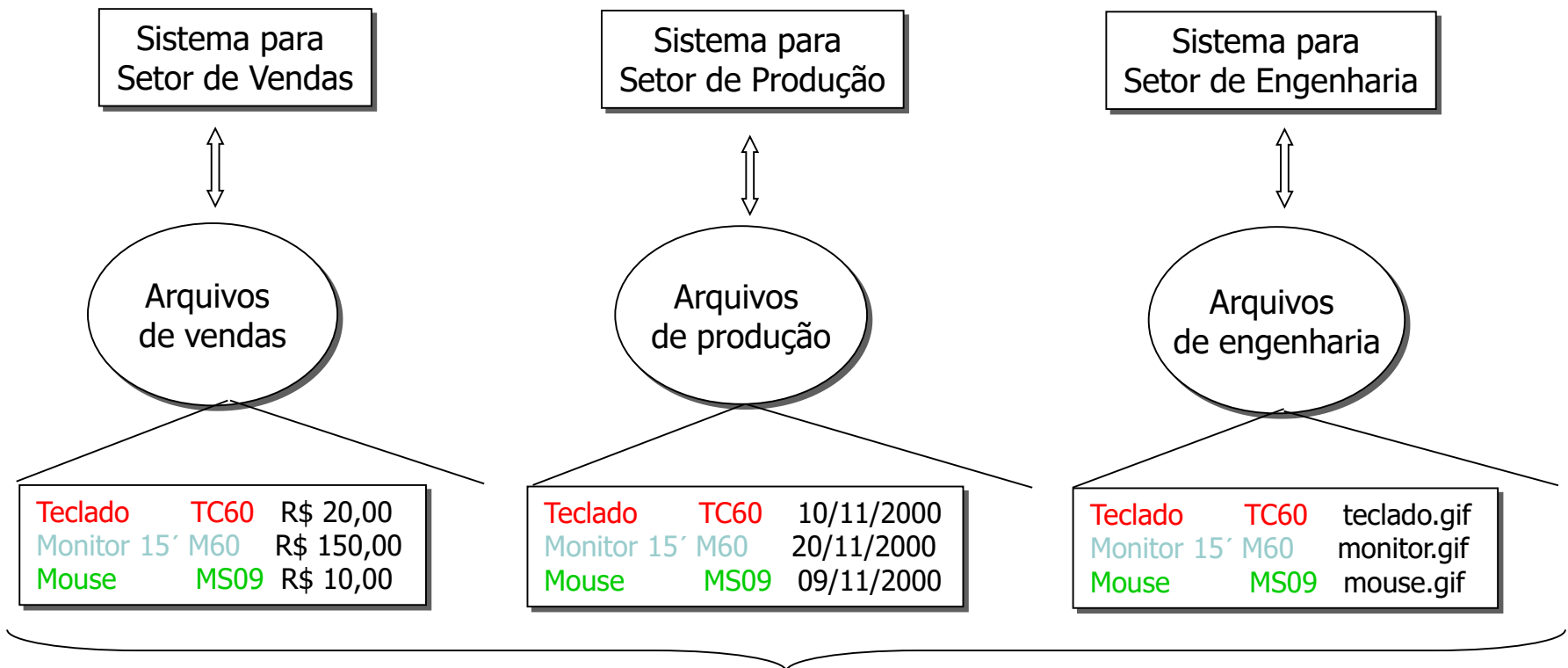
# Nem sempre foi assim...

---

- Sistemas de Arquivos (armazenados em pastas, no disco):
  - ✓ Funcionalidades oferecidas
    - Registros de tamanho fixo com campos de tipos diferentes
    - Possibilidade de memória virtual e persistência
    - Índices: *hash*, árvore-B
    - Bloqueio de arquivo e registro para concorrência
- Dados de diferentes aplicações **não estão integrados**
- Dados são projetados para atender **uma aplicação** específica

# Sistemas de arquivos

- Em uma fábrica com os dados em sistemas de arquivos:



Mesmos dados aparecem em todos os arquivos da fábrica



# Sistemas de arquivos

dados não integrados

---

- Mesmo objeto da realidade é representado várias vezes na base de dados
    - ✓ Exemplo - teclado, monitor e mouse
  - **Redundância não controlada** de dados
    - ✓ Não há gerência automática da redundância
    - ✓ Redundância leva a
      - *inconsistência dos dados*
      - *re-digitação de informações*
      - *dificuldade de extração de informações*
- Dados **pouco confiáveis** e de **baixa disponibilidade**

# Sistemas de arquivos

---

- Concorrência
  - ✓ Difícil implementação
  - ✓ Políticas de acesso concorrente consistente são independentes de domínio
- Tolerância a falhas
  - ✓ Falta de luz, erro de disco, interrupção de funcionamento, etc
  - ✓ Cópias? restauração do estado anterior? Consistência da base?
- Segurança
  - ✓ Acesso diferenciado por tipo de usuário

# Sistemas de arquivos

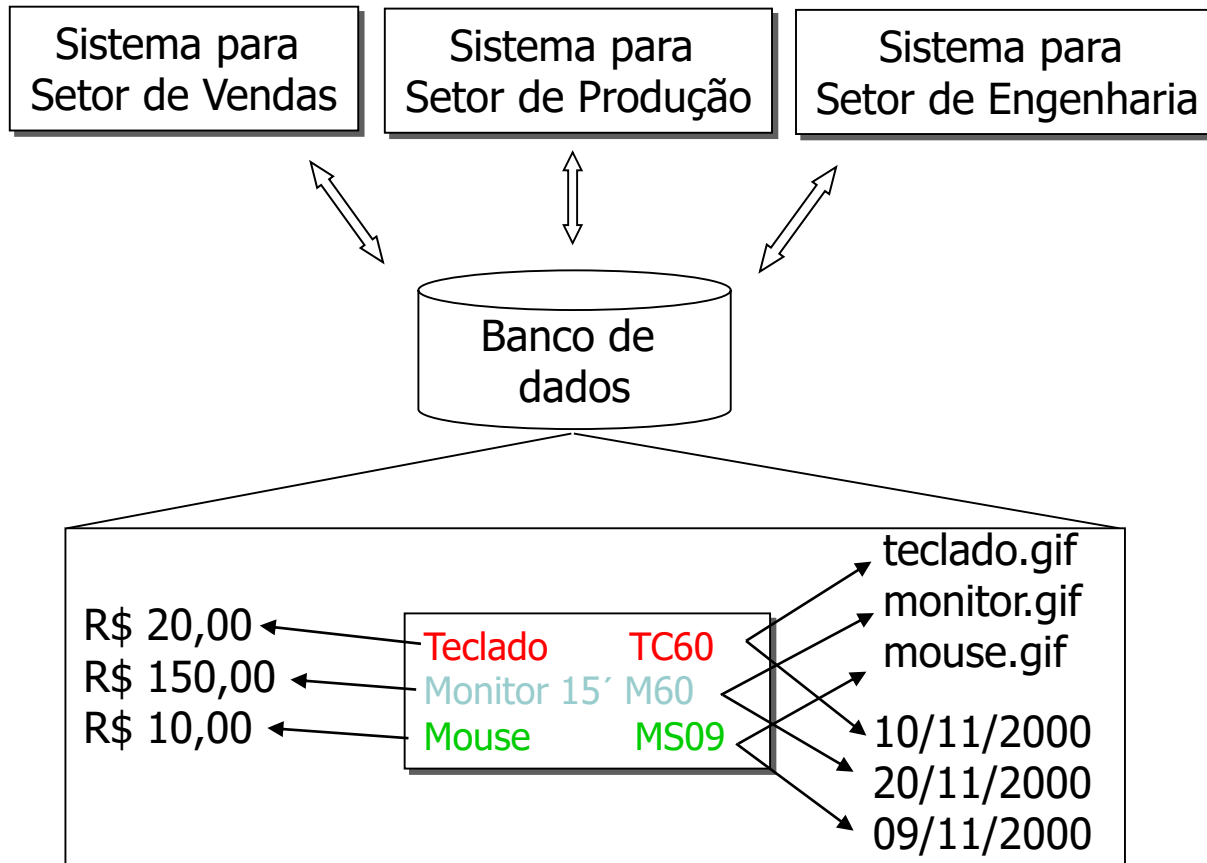
gerenciamento dos arquivos

---

- Outros problemas:
  - ✓ Número máximo de arquivos
  - ✓ Tamanho de memória
  - ✓ Limitações do tipo de arquivo, tipo de acesso
  - ✓ Preocupações técnicas junto com problemas do domínio
- Exemplo: efetuar aluguel de um DVD
  - ✓ Sem reservas? sem multas?
  - ✓ Como registrar um empréstimo?
    - abrir arquivos (*fechando outros ...*)
    - *carregar registros na memória (abre índice, usa ponteiro, estourou memória?, ....)*

# Banco de dados

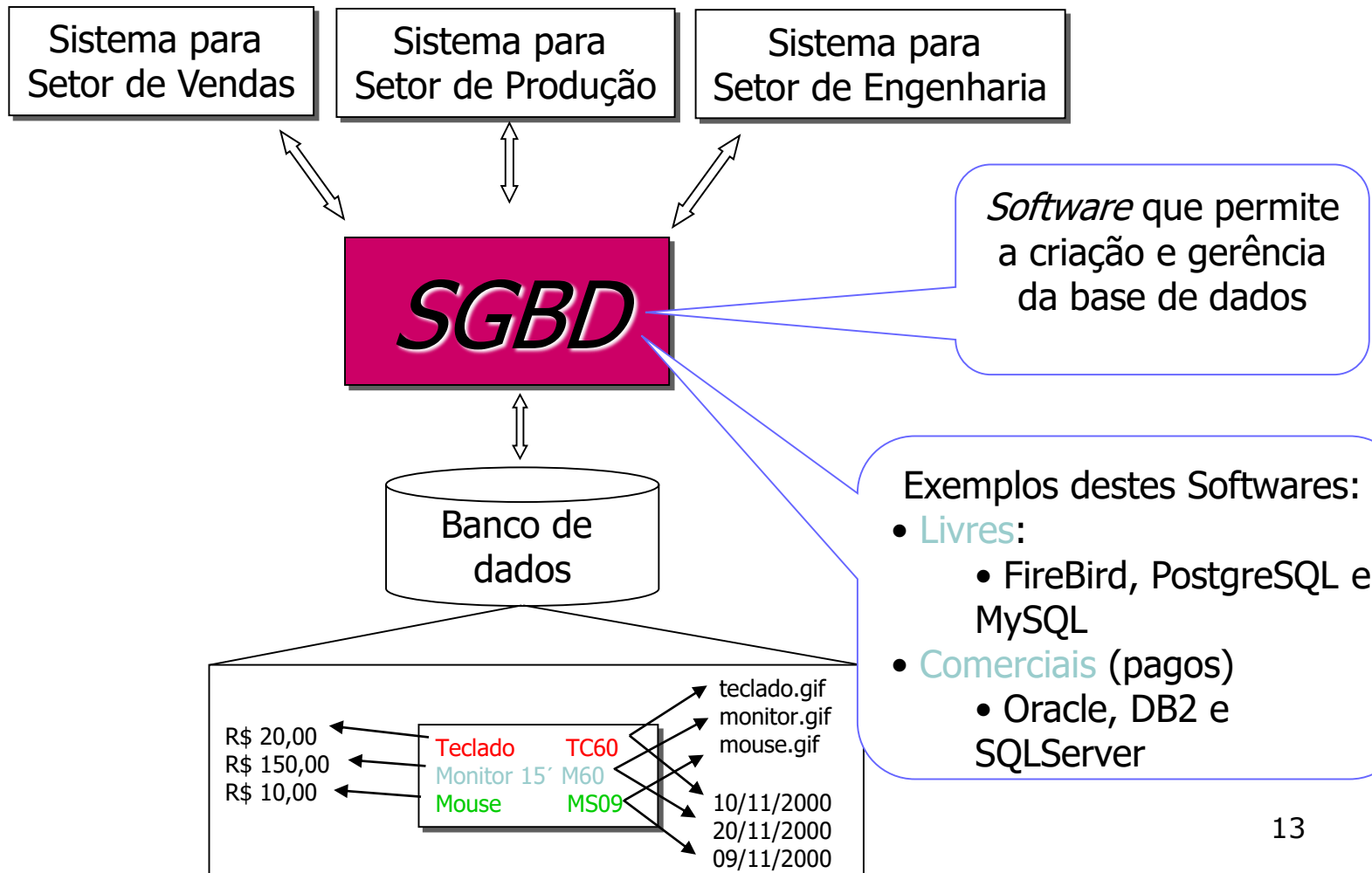
- Em uma fábrica com os dados em bancos de dados:



Dados aparecem  
uma única vez  
no banco

# Gerenciamento do banco de dados

## ■ BD de uma fábrica:



# Sistema Gerenciador de Bancos de Dados (SGBD)

---

- Um SGBD (Sistema Gerenciador de Banco de Dados) consiste em uma coleção de dados inter-relacionados **e em um conjunto de programas para acessá-los**
- SGBDs são projetados para gerenciar grandes grupos de informações

# SGBD

---

- O gerenciamento envolve
  - A definição de estruturas para o armazenamento da informação
  - O fornecimento de mecanismos para manipular as informações
- Quando vários usuários acessam os dados o SGBD precisa garantir a INTEGRIDADE dos dados, evitando resultados anômalos

# Objetivos de um SGBD

---

- Isolar os usuários dos detalhes mais internos do banco de dados (abstração de dados).
- Prover independência de dados às aplicações (estrutura física de armazenamento e à estratégia de acesso).

## **Vantagens:**

- rapidez na manipulação e no acesso à informação,
- redução do esforço humano (desenvolvimento e utilização),
- redução da redundância e da inconsistência de informações,
- redução de problemas de integridade,
- compartilhamento de dados,
- aplicação automática de restrições de segurança,
- controle integrado de informações distribuídas fisicamente.

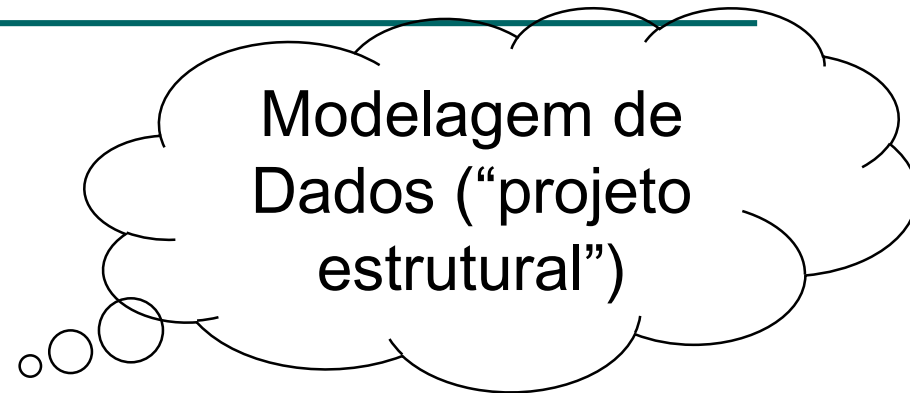
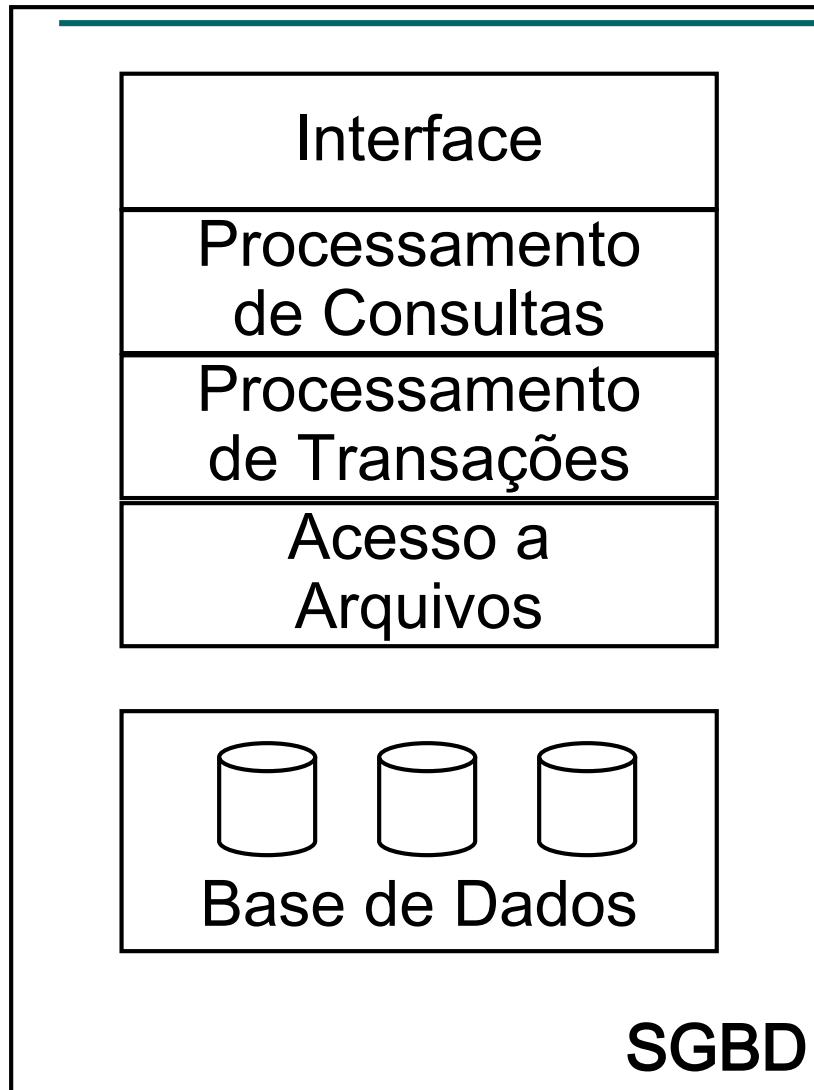


# Objetivos de um SGBD

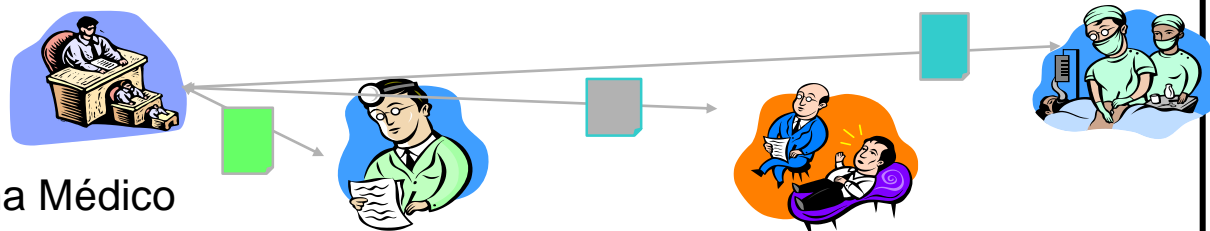
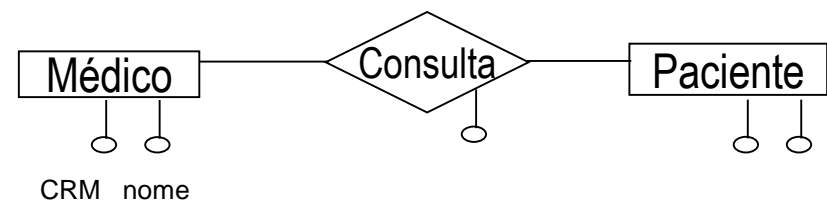
---

- O grande objetivo de um SGBD é prover aos usuários uma visão **ABSTRATA** dos dados
  - O sistema omite certos detalhes de como os dados são armazenados e mantidos
  - Mas oferece mecanismos eficientes para BUSCA e ARMAZENAMENTO

# Arquitetura Geral de um SGBD



# Abstração de Dados

Mundo Real	 <p>Sistema Médico</p>			
Modelo Conceitual (modelo abstrato dos dados)	<ul style="list-style-type: none"> <li>Independente do modelo de dados</li> <li>Independente do SGBD</li> </ul>			
Modelo Lógico (estrutura dos dados)	<ul style="list-style-type: none"> <li>Dependente do modelo de dados</li> <li>Independente do SGBD</li> </ul> <div>Médico (CRM, Nome)</div>	Relacional	Orientado a Objetos	Objeto-relacional
Modelo Físico	<ul style="list-style-type: none"> <li>Dependente do modelo de dados</li> <li>Dependente do SGBD</li> </ul>	<ul style="list-style-type: none"> <li>Organização física dos dados</li> <li>Estruturas de armazenamento de dados</li> <li>Índices de acesso</li> </ul>		



# Modelos de Datos

---

# Modelos de Dados

---

- Um **modelo de dados** é uma coleção de ferramentas conceituais para a **descrição** de dados, **relacionamentos**, semântica de dados e restrições de **consistência**

# Modelos de Dados

---

- Modelos de Dados (conceitual)
    - Entidade-Relacionamento (ER)
    - Orientado a Objetos (OO)
  - Modelos de Dados (lógicos)
    - Redes
    - Hierárquico
    - Relacional
    - Objeto-relacional
    - Orientado a Objetos
- } Modelos mais antigos

# Exemplo das Informações em um Banco de Dados

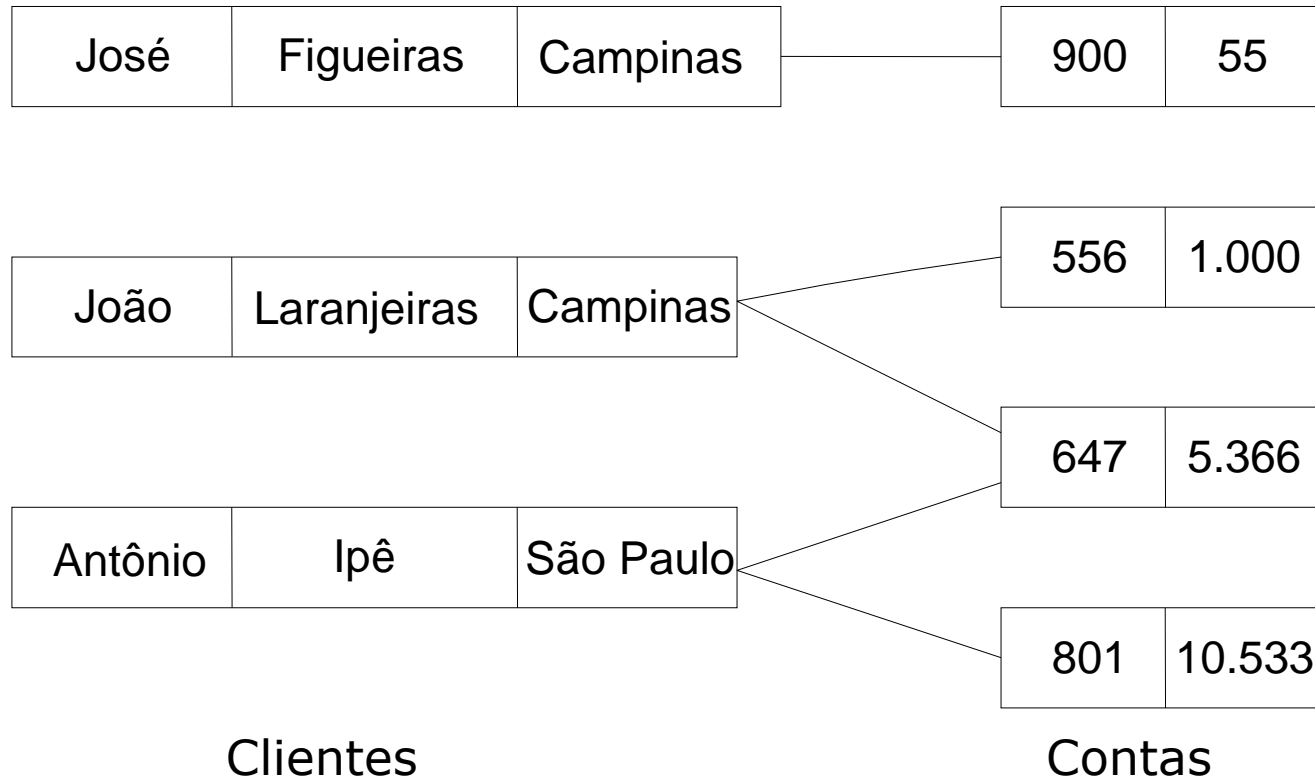
---

nome	rua	cidade	conta	saldo
José	Figueiras	Campinas	900	55
João	Laranjeiras	Campinas	556	1.000
João	Laranjeiras	Campinas	647	5.366
Antônio	Ipê	São Paulo	647	5.366
Antônio	Ipê	São Paulo	801	10.533

# O Modelo de Redes

---

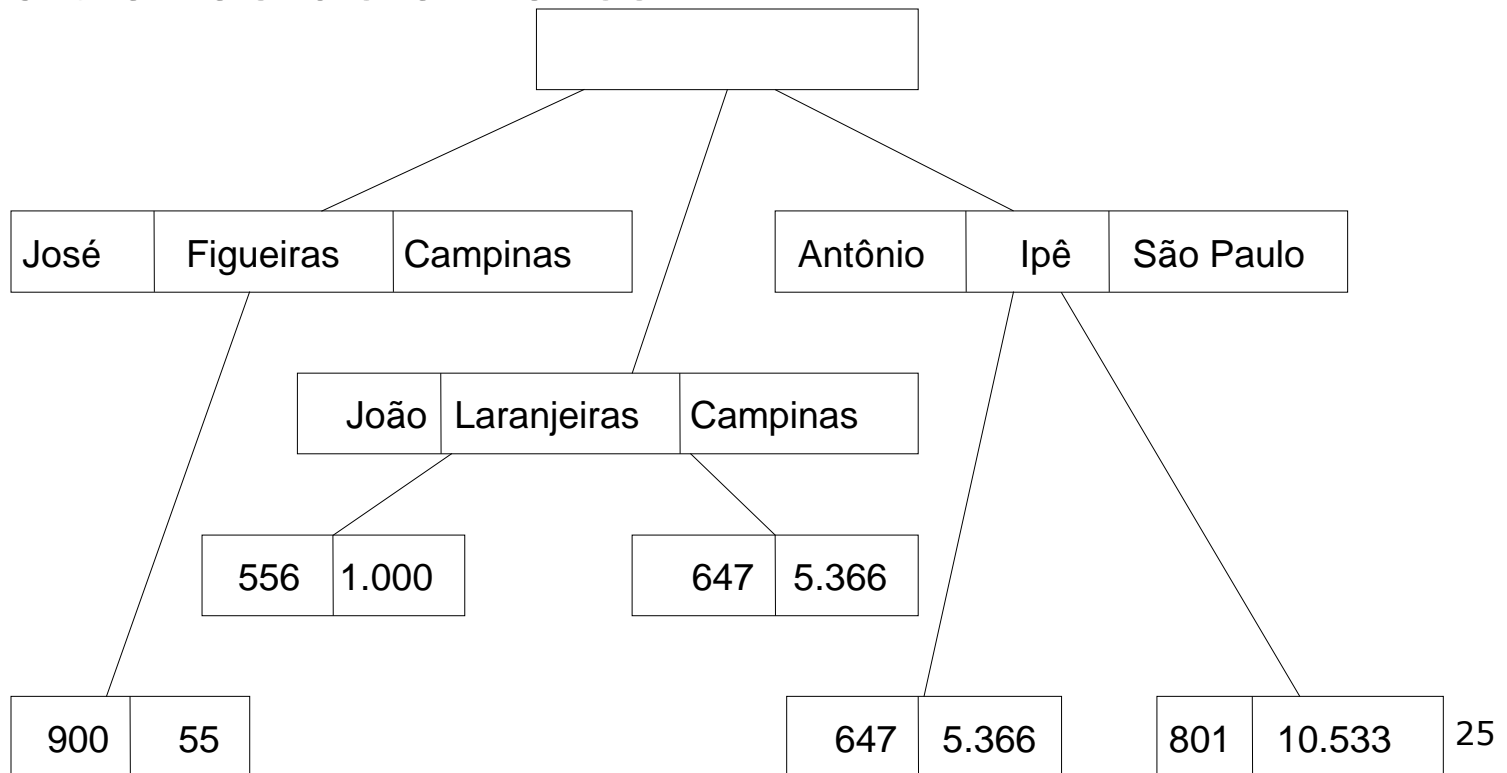
Os dados são representados por coleções de registros e os relacionamentos por elos





# O Modelo Hierárquico

- Os dados e relacionamentos são representados por registros e ligações, respectivamente.
- Os registros são organizados como coleções arbitrárias de árvores.



# O Modelo Relacional

Tabela Cliente (dados)

cód-cliente		nome	rua	cidade
015	José	Figueiras	Campinas	
021	João	Laranjeiras	Campinas	
037	Antônio	Ipê	São Paulo	

Tabela Conta (dados)

nro-conta	saldo
900	55
556	1.000
647	5.366
801	10.533

Tabela Cliente-Conta  
(relacionamento)

cód-cliente	nro-conta	
015	900	
021	556	
021	647	
037	647	
037	801	26

# Diferença entre os Modelos

---

- O modelo relacional não usa ponteiros ou ligações
- O modelo relacional relaciona registros a partir de valores do registro



# Funções de um SGBD

---

# Instâncias e Esquemas

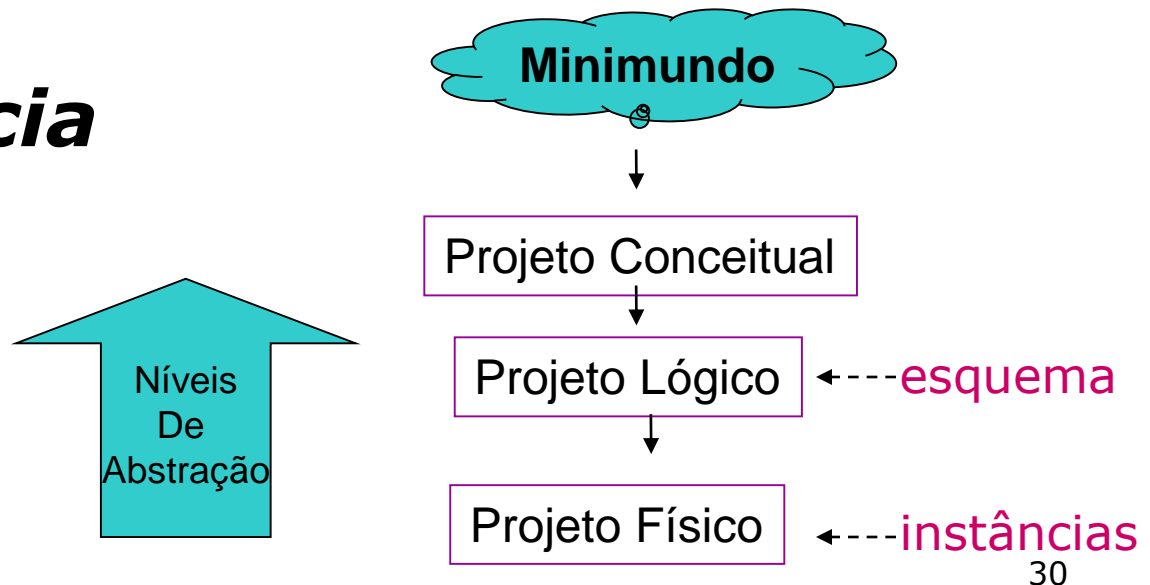
---

- Os bancos de dados mudam a medida que informações são inseridas ou apagadas
  - A coleção de informações armazenadas é chamada de **INSTÂNCIA** do banco de dados (mudam com frequência)
  - O projeto geral do banco de dados é chamado **ESQUEMA** do banco de dados (não mudam com frequência)

# Independência dos Dados

- O uso de bancos de dados permite modificar o ESQUEMA dos dados em um nível sem afetar a definição do esquema em um nível mais alto. Isto é chamado de

***independência dos dados***



# Independência dos Dados

---

- Existem 2 tipos de Independência
  - ***Independência física de dados:*** habilidade de modificar o **esquema físico** sem a necessidade de reescrever os programa aplicativos
    - Estas modificações são necessárias para melhorar o desempenho
  - ***Independência lógica de dados:*** habilidade de modificar o **esquema conceitual** sem a necessidade de reescrever os programas aplicativos
    - Estas modificações são necessárias quando a estrutura lógica é alterada.
      - Exemplo: adição de um novo atributo

# Independência dos Dados

---

- A independência lógica dos dados é mais difícil de ser alcançada do que a independência física, pois os programas são bastante dependentes da estrutura lógica dos dados que eles acessam



# Linguagem de Definição de Dados (DDL)

---

- Um esquema de banco de dados é especificado por um conjunto de definições expressas por uma linguagem especial chamada ***linguagem de definição de dados (Data Definition Language)***
- *O resultado da compilação de comandos de uma DDL é o conjunto de tabelas que serão armazenadas no dicionário (ou diretório) de dados*

# Linguagem de Definição de Dados (DDL)

- Um dicionário de dados contém metadados, i.e., dados sobre os dados
- Este dicionário (diretório) é consultado antes que os dados sejam lidos ou modificados no sistema de banco de dados

# Linguagem de Manipulação de Dados (DML)

---

- Manipulação de dados significa:
  - A busca da informação armazenada no BD
  - A inserção de novas informações no BD
  - A eliminação de informações do BD
  - A modificação dos dados armazenados no BD
- No nível físico precisamos definir algoritmos que permitam acesso eficiente aos dados

# Linguagem de Manipulação de Dados (DML)

---

- A linguagem de manipulação dos dados permite ao usuário manipular os dados da seguinte forma:
  - Procedural: o usuário informa qual dado deseja acessar e como obtê-lo
  - Não-procedural: o usuário informa qual dado deseja acessar SEM especificar como obtê-lo

# Linguagem de Manipulação de Dados (DML)

---

- Linguagens não-procedurais são usualmente mais fáceis de aprender e usar do que DMLs procedurais
- Se o usuário NÃO especificar COMO obter os dados, as linguagens não-procedurais poderão gerar um código não tão eficiente.

# Linguagem de Manipulação de Dados (DML)

---

- Uma consulta (QUERY) é um comando de busca de uma informação no BD
- A parte da DML que busca informações é chamada **LINGUAGEM DE CONSULTA**

# Usuários do Banco de Dados

---

- Programadores de Aplicativos:
  - São os usuários que escrevem os programas de aplicação através da DML
    - Exemplos de um sistema bancário são programas que geram cheques, fazem débitos e créditos em contas, transferem fundos entre contas
- Usuários de alto nível
  - Interagem com o sistema sem escrever programas
  - Formulam consultas em uma linguagem de consulta, e cada consulta é submetida a um processador de consulta, cuja função é gerar um comando da DML

# Usuários do Banco de Dados

---

- Usuários especializados (especialistas)
  - Escrevem aplicativos especializados como sistemas especialistas
- Usuários ingênuos
  - Interação com o sistema invocando os programas aplicativos
    - Exemplo: um cliente do banco invocaria um programa para efetuar a transferência de 50 reais da conta A para a conta B



# Usuários do Banco de Dados

---

- **Administrador do banco de dados:**  
tem o controle central dos dados e dos programas de acesso aos dados

# Usuários do Banco de Dados

---

- **Funções do Administrador do banco de dados:**
  - Definição do esquema
  - Definição de estruturas de armazenamento e métodos de acesso
  - Modificação de esquema e de organização física
  - Concessão de autorização para acesso aos dados
  - Especificação de restrições de integridade

# Estrutura Geral do sistema

---

- Gerenciador de arquivos
- Gerenciador do banco de dados
- Processador de consultas
- Pré-compilador da DML
- Compilador da DDL
  - Arquivos de dados
  - Dicionário de dados
  - Índices

# Transações

---

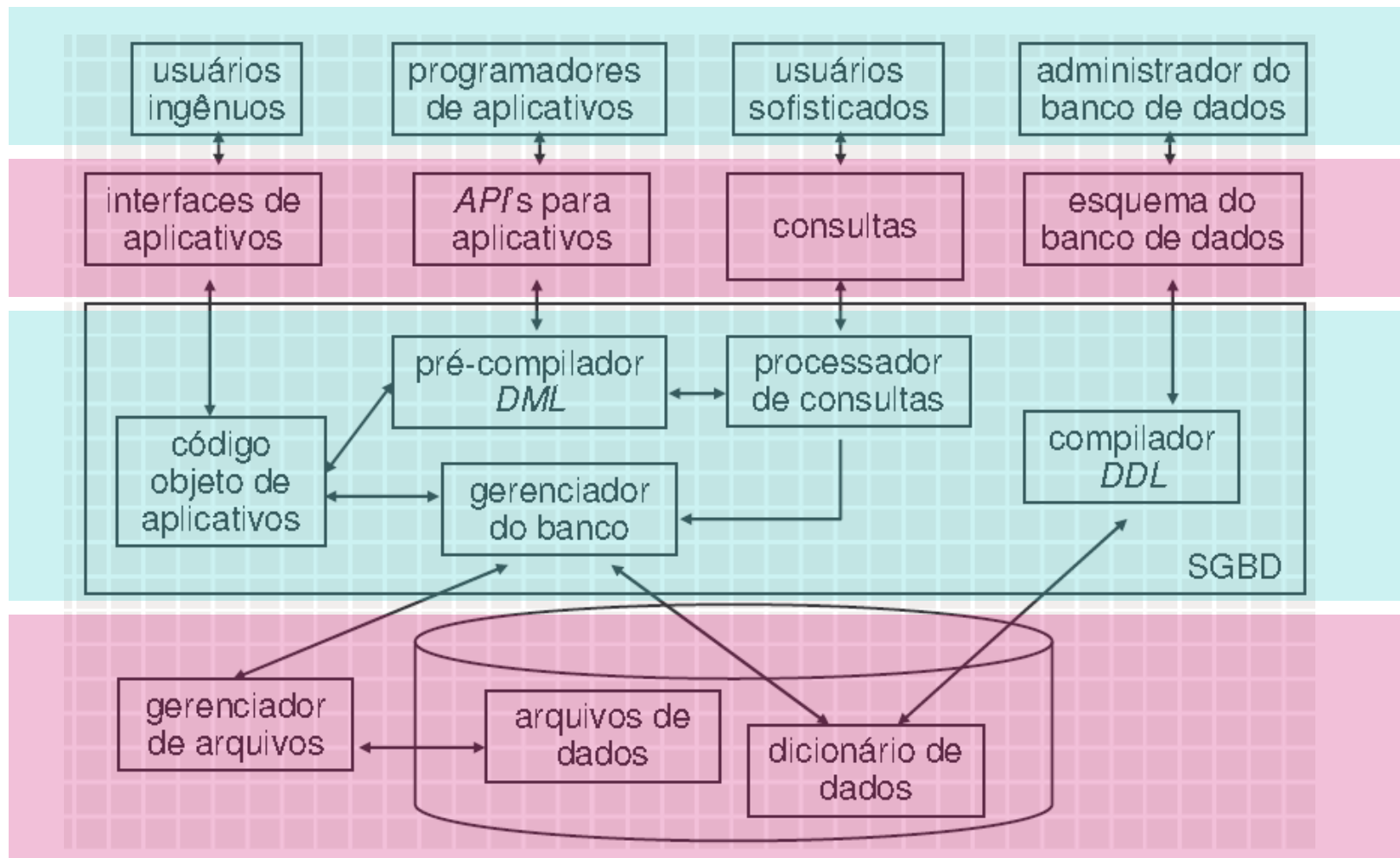
- Utilizadas para controlar a integridade dos dados no Banco de dados
  - Acessos simultâneos vários usuários
  - Falhas no sistema

# Otimizador de Consultas

---

- Escolhe a forma mais eficiente para execução de uma consulta

# Arquitetura Geral de um SGBD



# Bibliografia

---

- Elmasri & Navathe – Fundamentos de Bancos de Dados
- Carlos Alberto Heuser – Projeto de Banco de Dados
- Korth e Silberchatz – Sistema de Bancos de Dados