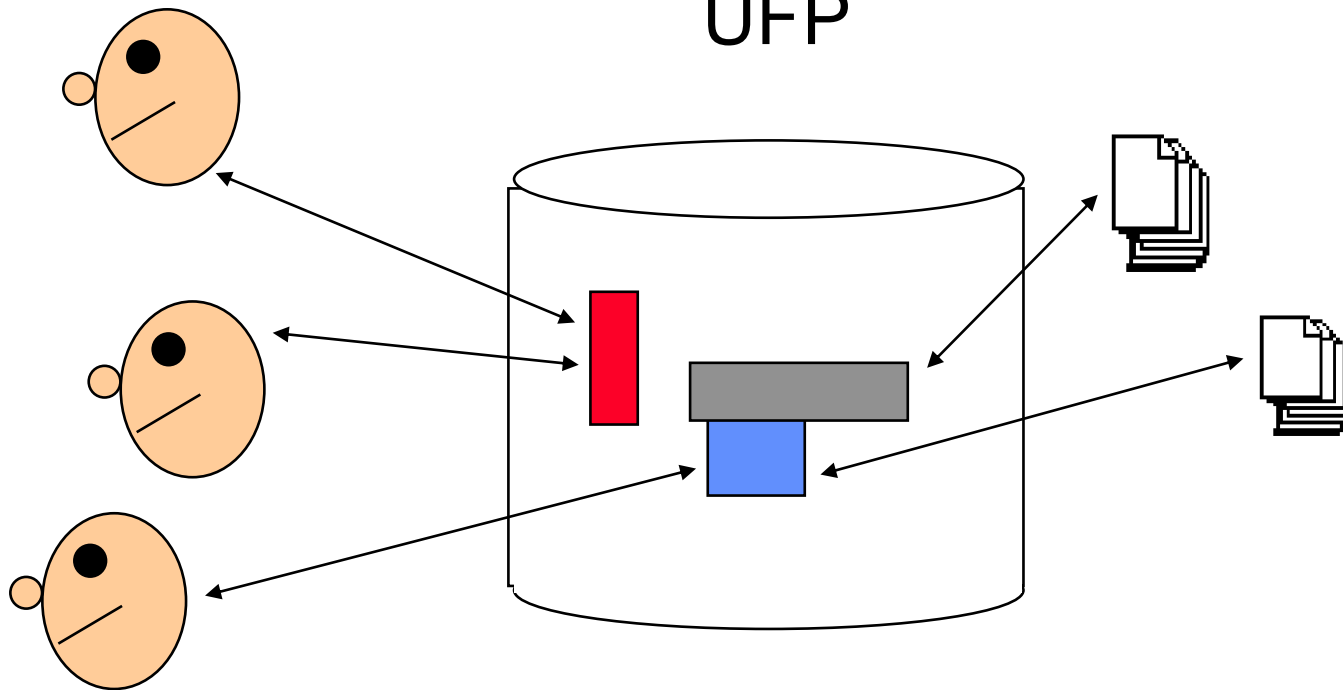
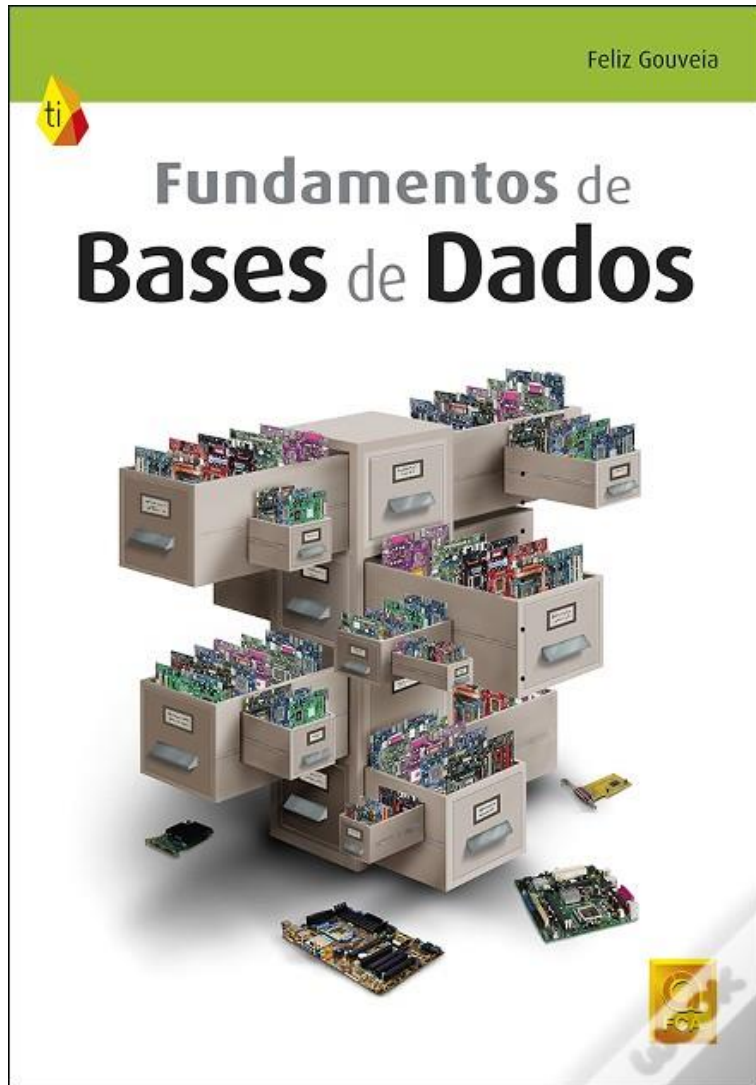


# Introdução aos SGBD

Feliz Gouveia  
fribeiro@ufp.edu.pt  
UFP



# Bibliografia

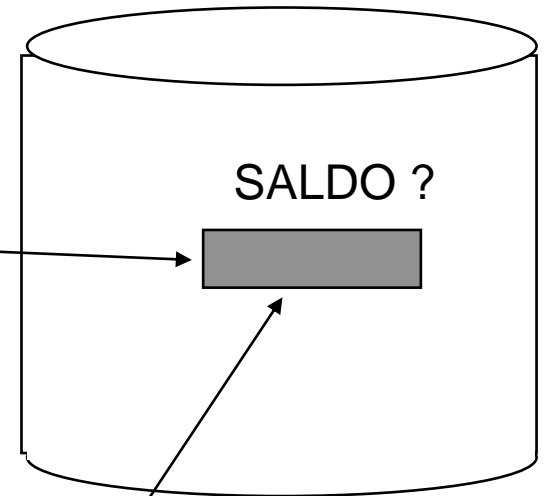


- Disponível na biblioteca, e também:
- A first course in database systems - Jeffrey D. Ullman, Jennifer Widom
- Structured Query Language - Luís Damas

# Introdução

- Em situações comuns, vários utilizadores necessitam partilhar informação:
  - Vendas, assinaturas, custos, produtos
- É necessário que as suas operações não tenham interferência
- Todos deverão ter uma percepção uniforme sobre os tipos de dados, valores, constantes,...

# Operações sem interferência



# Solução (?)

- Partilha de ficheiros não iria funcionar
- Regras de segurança e acessos teriam que ser implementadas de forma centralizada
- Critérios de disponibilidade, cópias de segurança teriam que ser definidas de forma centralizada
- Regras de negócio teriam que ser uniformes (por exemplo, formato do código de produto em todas as aplicações)

# Breve histórico

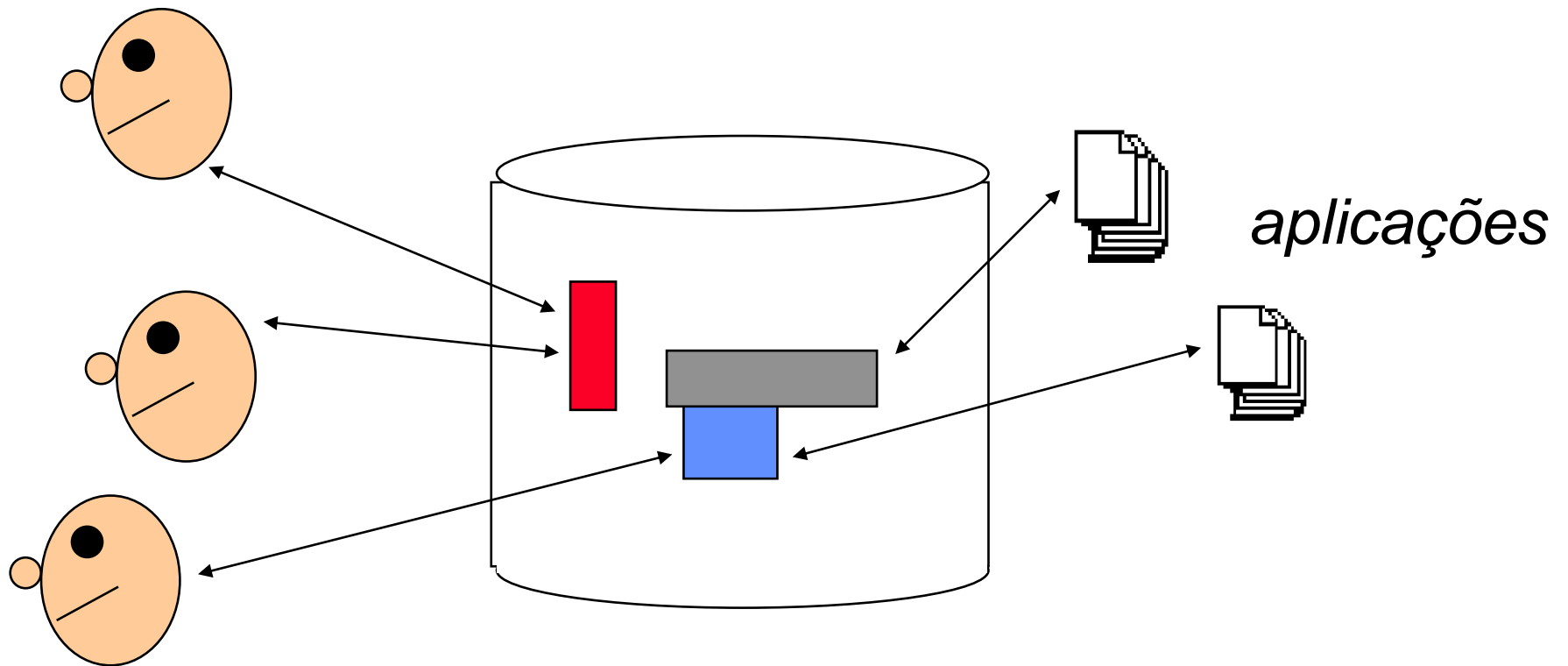
- 1959: CODASYL, *Conference on Data Systems Languages*, estabelece dois comités para desenvolver uma linguagem aplicativa:
  - Comité “curto prazo” desenvolve COBOL
  - Comité “longo prazo” cria o CODASYL List Processing Task Group, mais tarde CODASYL Database Task Group (CDTG)
- 1960: Charles Bachman cria o primeiro SGBD
- 1966: NASA Apollo: IBM, Rockwell e Caterpillar criam base de dados (futuro IMS)

## Breve histórico (2)

- 1969: CDTG publica a especificação do Modelo em Rede
- 1970: Ted Codd (da IBM) publica as bases teóricas do modelo relacional de dados
- 1976: Chen Peter Pin-Shan publica o modelo Entidade-Relacionamento (E-R)
- 1977: Larry Ellison e Bob Milner fundam a Oracle

# Um SGBD

SGBD: esconde dos utilizadores os detalhes de *implementação física* dos dados





# Sistema de Gestão de Bases de Dados

- é o programa que gere os acessos à BD
- permite definir os dados
- permite manipular os dados (ler, modificar, apagar, inserir)
- implementa segurança e integridade
- implementa concorrência e recuperação de dados

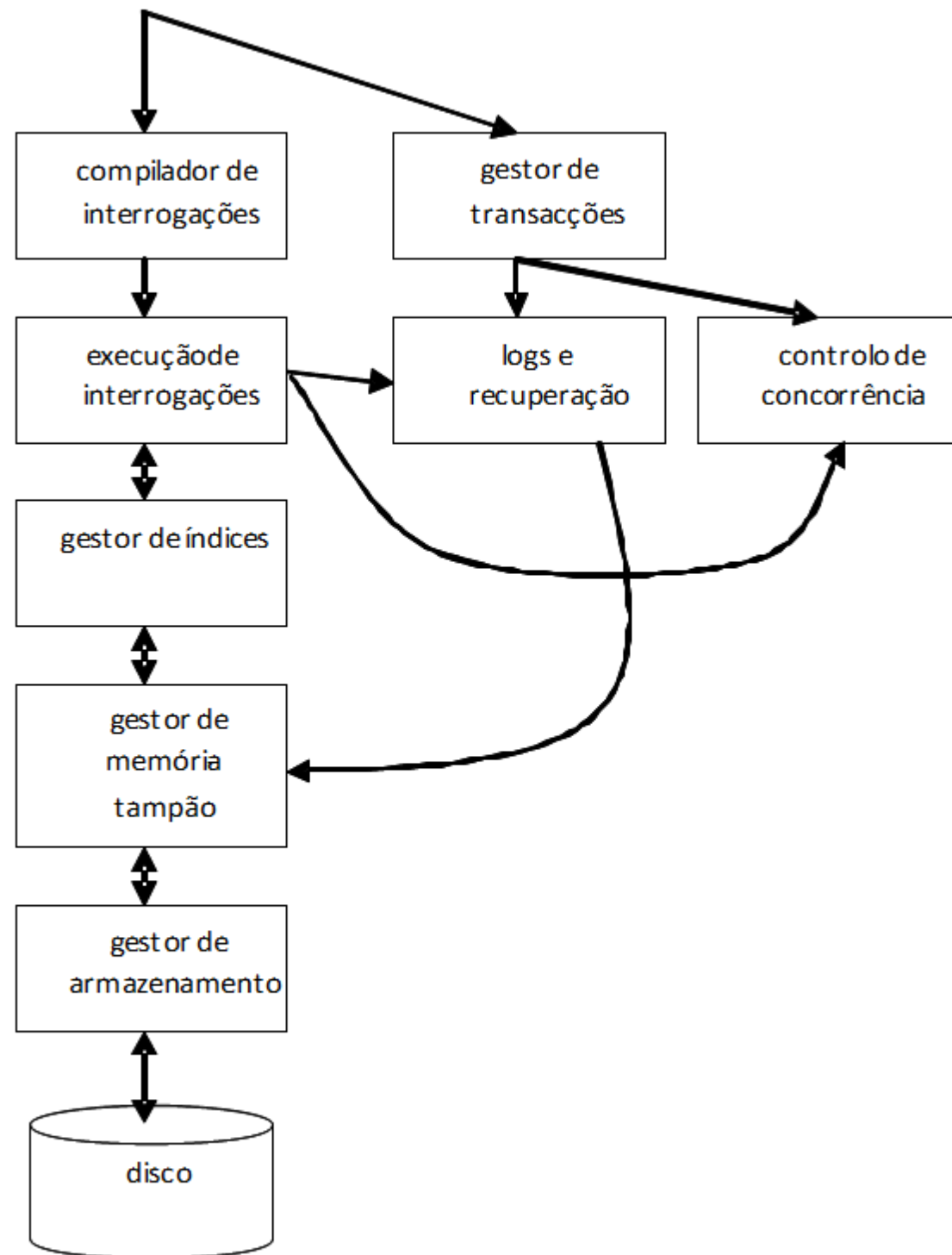
# SGBD (cont.)

- implementa um Dicionário de Dados: contém informação sobre a informação
- assegura um desempenho aceitável
- dito de outra forma, assegura a interface com o utilizador
- é a ferramenta principal de uma BD

# Abstracções principais

- Independência dos dados: podem-se ignorar os detalhes da representação interna, ou de acesso eficaz
- Atomicidade das transacções: permite ignorar os efeitos de outras aplicações que acedem concorrenciaismente aos dados
- Independência da localização física

utilizadores, programas



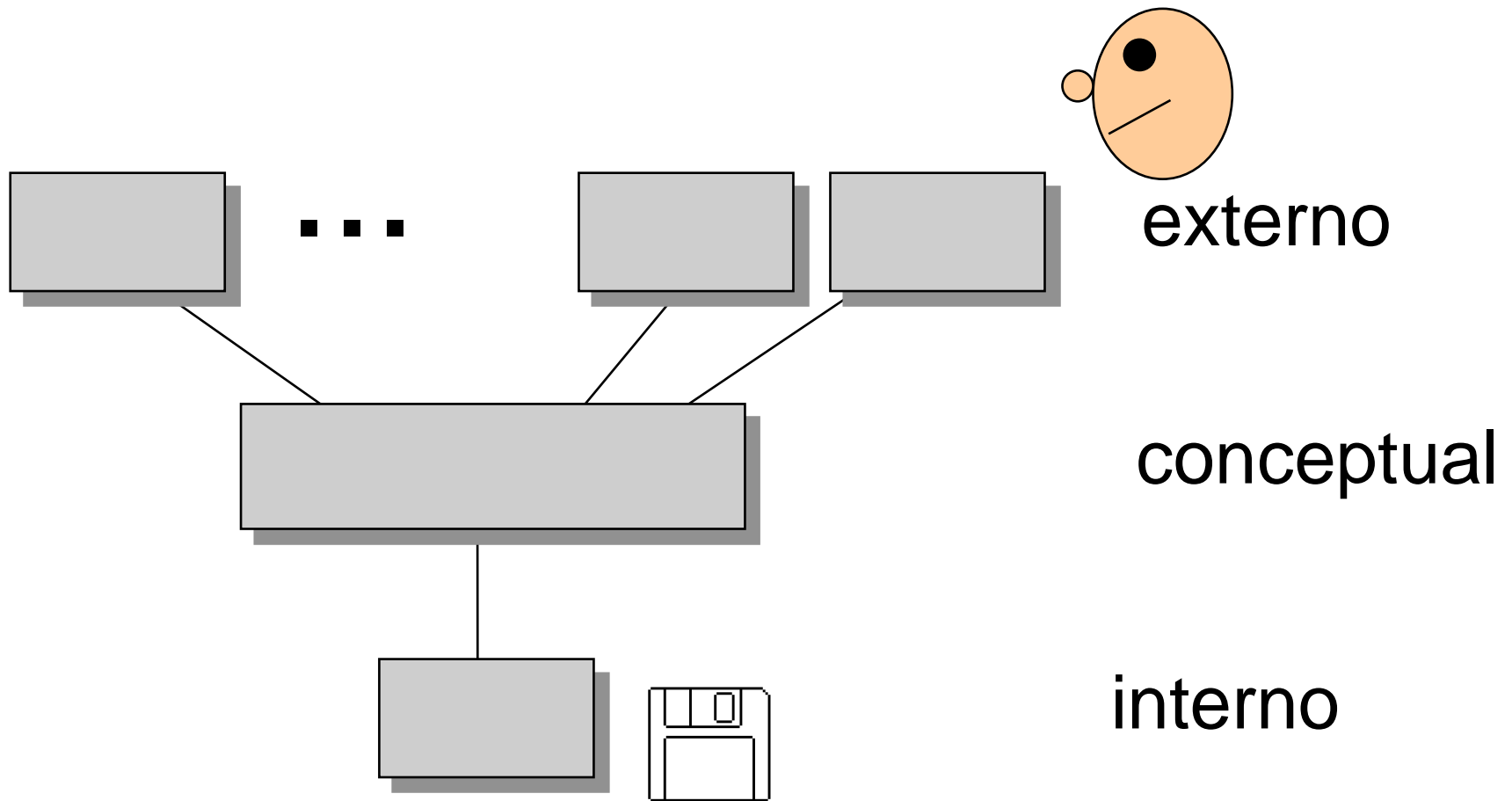
# Componentes principais

- Interrogações
- Indexação
- Controlo de Concorrência
- Gestão de Memória
- Logs e Recuperação
- Gestão do Disco

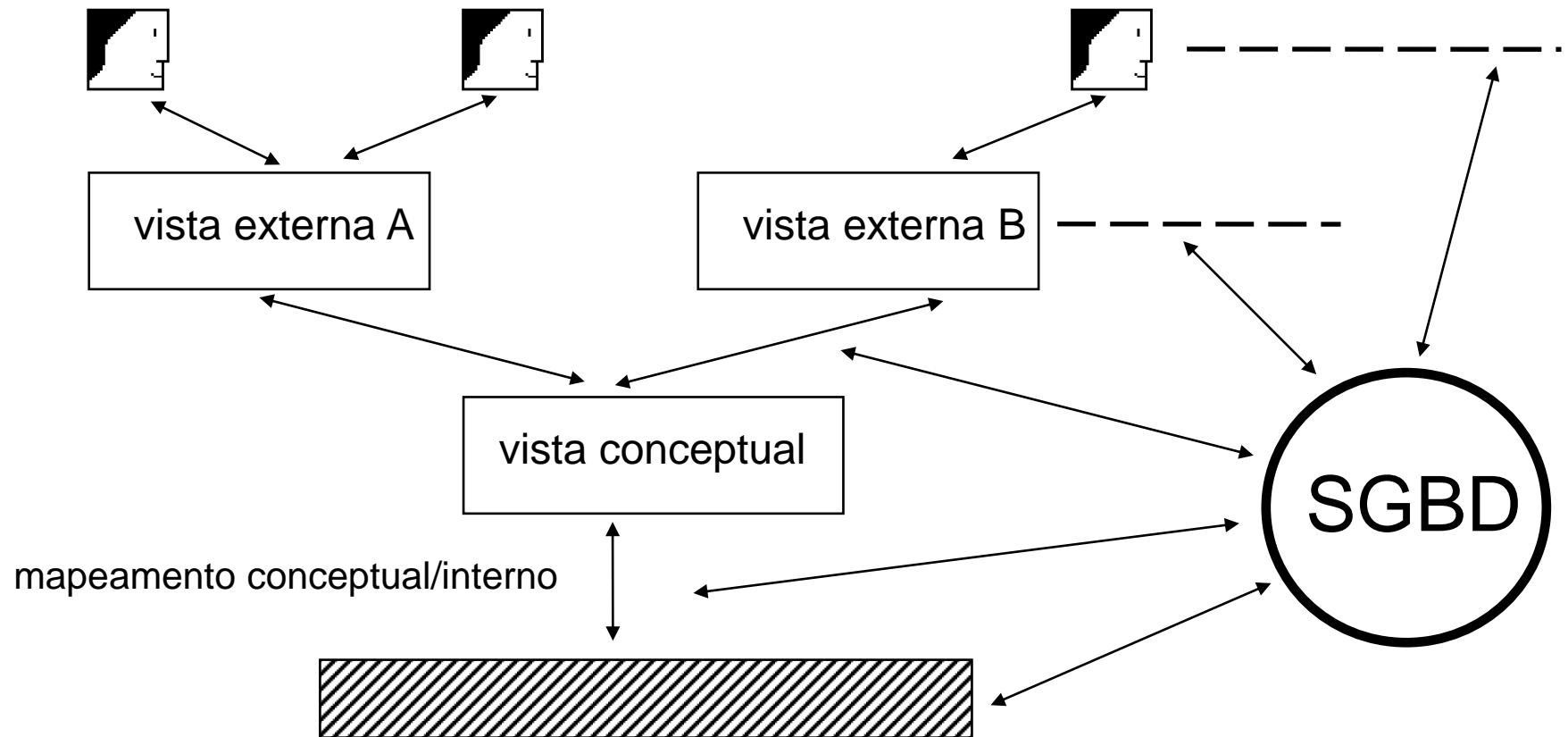
# Arquitectura de um SGBD

- definida pelo ANSI/SPARC Study Group on DBMS
- é uma arquitectura a três níveis:
  - interno: próximo do armazenamento físico
  - externo: mais próximo do utilizador
  - conceptual: nível intermédio entre os dois níveis anteriores

# Níveis da Arquitectura



# Diagrama detalhado





# Nível externo

- Existem vários níveis externos (várias vistas sobre os mesmos dados)
- cada tipo de utilizador tem uma linguagem à sua disposição:
  - o utilizador final dispõe de uma interface gráfica de interrogação
  - o programador dispõe de uma linguagem de programação proprietária, ou convencional (C++, Java...)

# Nível conceptual

- Define a estrutura dos dados
- representação completa da informação de uma forma abstracta, se comparada com a representação física dos dados

# Nível interno

- representação a nível máquina (física)
- as aplicações não devem operar a este nível
- várias técnicas de representação e de indexação são utilizadas

# Os dados

- O Modelo Relacional de Dados é o modelo dominante
- A única estrutura de dados é a tabela (relação)
- Uma tabela tem colunas, que definem as características dos dados
  - Uma coluna tem um tipo (char, int, date,....)
- As linhas da tabela são os dados
- Tabela ALUNO {Nº, Nome, Morada}