



**UC1:** Desenvolver algoritmos e estrutura de dados  
para aplicativos móveis

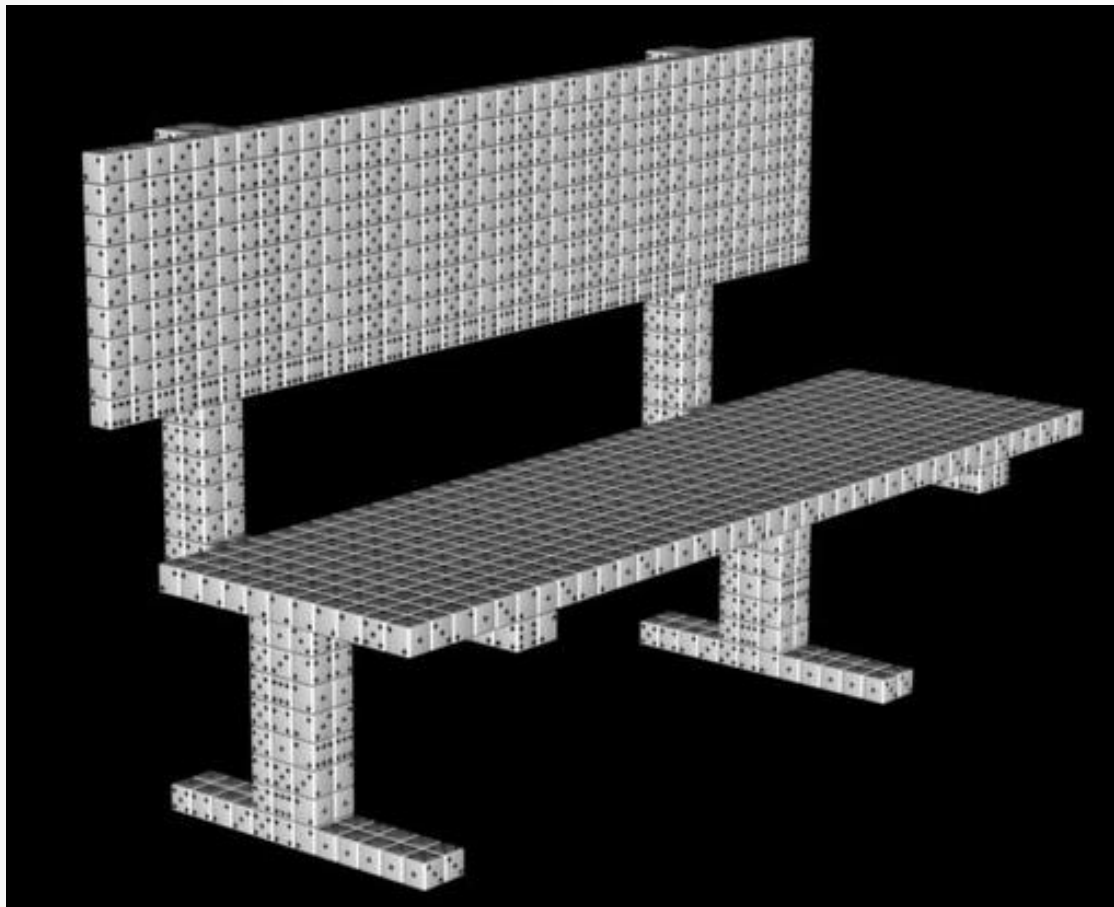
## **Bancos de Dados**

**Modelo Conceitual**

**Docente:** Felipe B. Benutti

# Introdução

- O que é um banco de dados?



# Introdução

- Os sistemas de informação precisam armazenar as informações que são geradas neles.
- Caso não sejam salvas, toda vez que desligarmos o nosso computador, todos os **dados seriam perdidos**.
- Também é válido para **sistemas mobile**, onde devemos salvar as informações antes de fechar o aplicativo, correto?
- É aí que entram os **Bancos de Dados**

# Introdução

- Uma definição para Banco de Dados seria:

**"Bancos de dados** são coleções de ***dados*** interligados entre si e organizados para fornecer ***informações.***"

# Introdução

- **Dados x Informações**
  - **Dados** são fatos brutos, em sua forma primária. E muitas vezes os dados podem não fazer sentido sozinhos.
  - **Informações** consiste no agrupamento de dados de forma organizada para fazer sentido, dado um conhecimento prévio.

**INFORMAÇÃO = Dado + Conhecimento**

- **Dado:** 2001;
- **Informação:** Ano do atentado terrorista às torres gêmeas: 2001;

# Introdução

- Os bancos de dados são constituídos basicamente por **tabelas (entidades)** que armazenam os **dados**.
- As tabelas, assim como toda tabela, tem **linhas e colunas**.
- As colunas das tabelas descrevem os **campos**, ou seja, os **atributos** dessa tabela.
- As **linhas** armazenam os dados de cada entidade propriamente dita.

*Nota: **Registro**, **Linha** e **Tupla** são palavras sinônimas para referenciar a uma linha da tabela.*

# Introdução

- Metadados
  - **Meta** (em inglês) significa “auto referencial”, ou seja, **metadado** significa que um dado é relativo à outro dado.
  - No exemplo anterior, a frase “Ano do atentado às torres gêmeas” é um metadado, pois ele um dado sobre o dado “2001”.
  - Já em banco de dados, por exemplo, um campo telefone de uma tabela cliente tem os seguintes metadados: nome do campo (telefone), tipo do campo (texto), tamanho do campo (30 caracteres), obrigatoriedade (sim ou não), e etc...

# Introdução

- Cada banco de dados tem seu próprio nome.
- Os bancos de dados podem ter dezenas, centenas e até milhares de tabelas!



# Introdução

- Exemplos de Bancos de Dados:
  - Sistema de Escola (alunos, cursos, sala de aula)
  - Rede Social (perfil, postagens, comentários, amigos)
  - E-commerce (login/cliente, produtos, carrinho)

(Desenho conceitual do banco de dados e explicação no Excel)

# Introdução

- Sistema de Escola

- Aluno:

- Nome, data de nascimento, telefone, endereço, etc...

- Sala de Aula:

- Número da sala, tipo (lab, auditório, sala comum), etc...

- Cursos:

- Nome do curso, carga horária, área de conhecimento, etc...

# Introdução

- Rede Social

- Perfil:

- Nome, data de nascimento, celular, local de trabalho, local de estudo, foto, etc...

- Postagens:

- Data que postou, texto da postagem, data de alteração (se tiver), texto alterado (se tiver), foto, vídeo, etc...

- Comentários:

- Quem comentou, texto do comentário, data que comentou, data de alteração (se tiver), texto alterado (se tiver), etc...

- Amigos:

- Nome, Data que iniciou amizade, tipo de amizade (melhores amigos, conhecidos, etc)

# Introdução

- E-commerce
  - Cliente:
    - Nome, data de nascimento, telefone, endereço, etc...
  - Produtos:
    - Descrição, quantidade, preço venda, preço de custo, fornecedor, etc...
  - Carrinho:
    - **Coluna que identifique** o cliente, **Coluna que identifique** o Produto, quantidade, valor total, etc...
- Nota-se uma necessidade de **relacionar!**

# Introdução

- Outra composição importante de um banco de dados são os **relacionamentos**.
- Exemplos:
  - Carrinho de compras;
  - Alunos que cursam um determinado curso;
  - Amigos em comum no *facebook*;
- O que difere portanto os bancos de aplicações que gerenciam dados genericamente (por exemplo, o excel) é justamente o fato de serem **relacionais**.  
(além de outros conceitos que serão abordados posteriormente)

# História

- História dos banco de dados
- **Antigamente:**
  - Sistemas de controles de empresas não eram digitais;
  - Arquivos físicos;
  - Muito espaço ocupado;
  - A medida que a quantidade de arquivos aumentava, dificultava a procura por informações específicas neles;

# História

Positivo

Negativo

O valor da informação é crescente

Quantidade de Informação é cada vez maior

Custo de armazenamento

Dificuldade para obter dados relevantes de forma rápida

Alto custo para aquisição e gerenciamento dos dados

# História

- Com o passar do tempo, a era da informação, os sistemas ficaram digitais
- Cada entidade (clientes, funcionários, produtos, etc) era um arquivo de dados que eram acompanhados de um "software simples" para manipular os dados do arquivo
- Houve uma melhora na obtenção de informações, porém as entidades não se relacionavam!

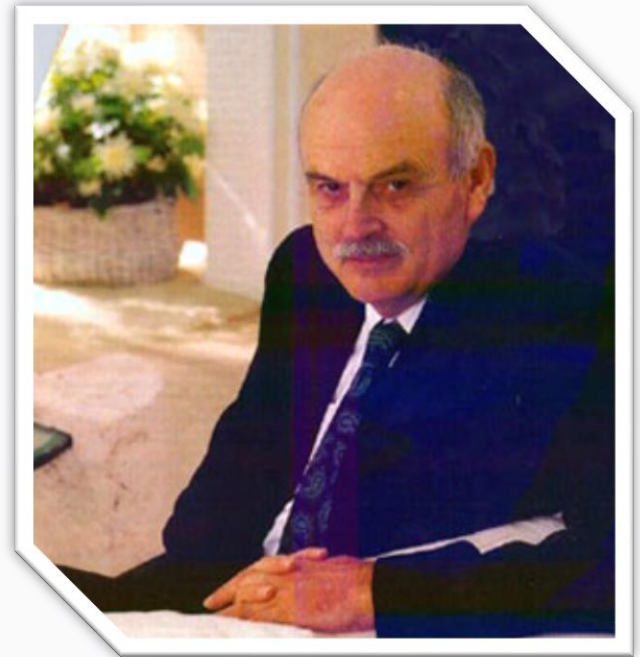


# História

- Relacionar, por exemplo, um produto que é fornecido por um fornecedor, não era uma tarefa fácil.
- Então, na década de 60, a empresa IBM investiu fortemente em pesquisas para solucionar estes problemas.
- Vários bancos de dados surgiram a partir daí, entre eles o modelo **hierárquico** e **rede**, os quais já contavam com relacionamentos.

# História

- Em 1970, o pesquisador Edgar Frank “Ted” Codd mudou a história dos bancos de dados apresentando o **modelo relacional**.
- Toda a sua definição é baseada na lógica de predicados e na **teoria dos conjuntos**.
- Foi o primeiro modelo de dados descrito em linguagem formal (matemática).



# História

- E assim surgiram os **Sistemas de Gerenciamento de Bancos de Dados**.
- Atualmente, os bancos de dados amplamente utilizados são **relacionais**.
- Portanto, focaremos apenas nestes.

# SGBDs

- Sistemas de Gerenciamento de Bancos de Dados
  - Para que os bancos de dados possam existir, existem sistemas que gerenciam os bancos de dados, melhorando a performance, gerenciando a memória utilizada, entre outros aspectos importantes.
  - Pense nos SGBDs como o Sistema Operacional de um banco de dados.
  - *Analogia:*
    - Assim como Word, Excel e outros programas estão para Windows - os **bancos de dados** estão para os **SGBDs**

# SGBDs

Usuários / Programadores



\*Servidor  
\*Dispositivo Móvel

Consultas / Programas



SGBD



Catálogo  
(Meta-Dados)

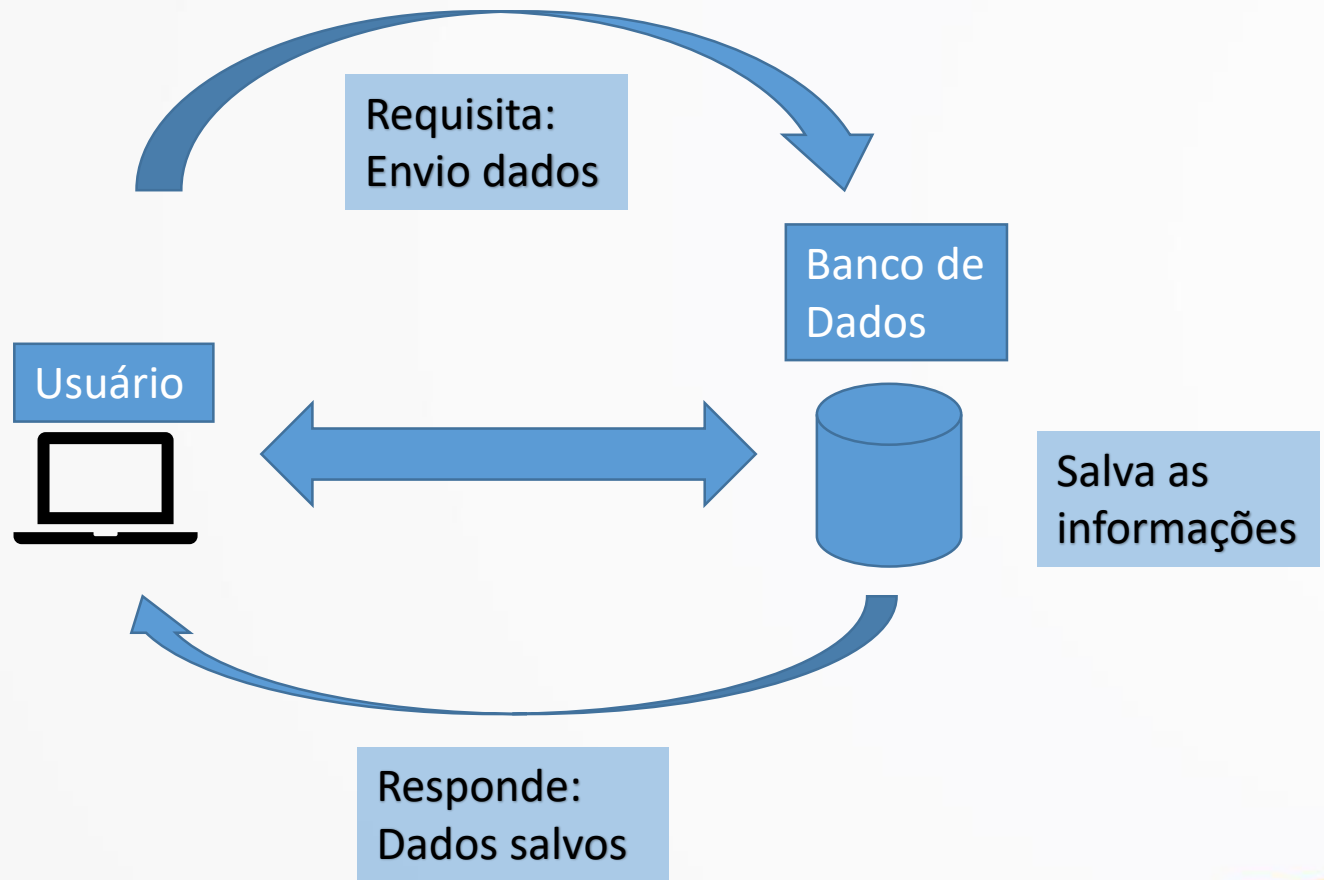


Banco  
De  
Dados

Sistema de Banco de Dados

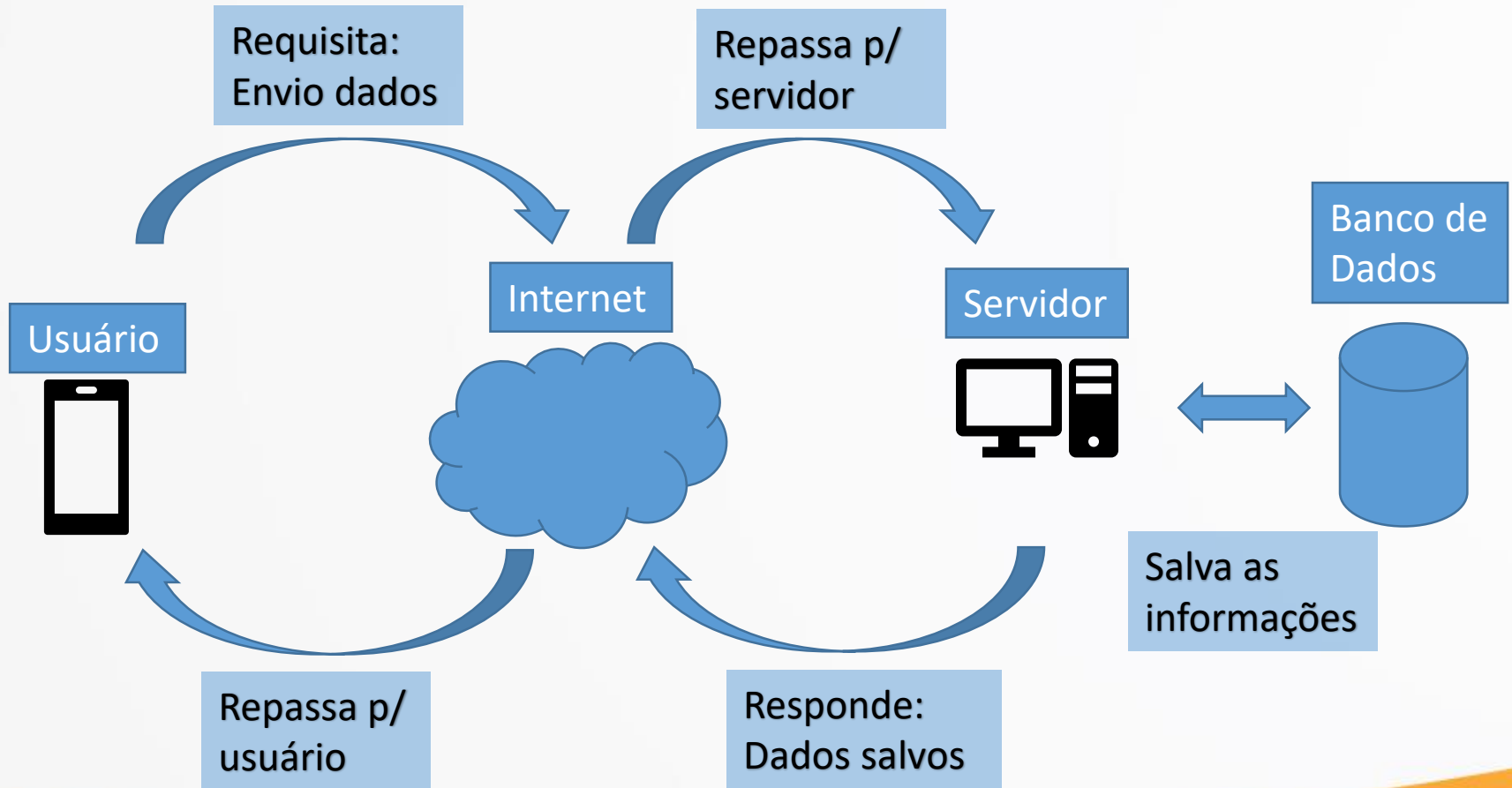
# SGBDs

- Esquema de comunicação (Desktop ou mobile offline)



# SGBDs

- Esquema comunic. Mobile (apps com integração ou bd online)



# SGBDs

- Data Base Management System (DBMS)
- Software projetado para auxiliar na criação, manutenção e utilização de grandes coleções de dados.
- Sempre lembrar que: **“A informação é muitas vezes a coisa mais valiosa das empresas”**



# O que se espera de um SGBD

Permitir a criação de  
manutenção de  
bancos de dados

Permitir aos usuários  
acesso aos dados  
(mecanismos de  
busca)

Permitir aos usuários  
alterar os dados

Dar suporte ao  
armazenamento de  
grandes quantidades  
de dados por um  
grande período de  
tempo

Dar segurança aos  
dados, não permitindo  
a perda por acidente  
ou acesso não  
autorizado

Permitir que vários  
usuários acessem os  
dados ao mesmo  
tempo, sem risco de  
perda dos mesmos

# SGBDs

- Existem diversos SGBDs relacionais no mercado atualmente:
  - **SQLite; (trabalharemos com este!)**
  - **MySQL;**
  - **SQL Server;**
  - **PostgreSQL;**
  - **Oracle;**
  - **Firebird;**
  - **Entre outros;**

# SGBDs

- Existem diversos outros SGBDs que não são relacionais (também chamados NoSQL) e estão ganhando muito espaço ultimamente, entre eles:
  - **Firestore (Google); (trabalharemos com este!)**
  - **MongoDB;**
  - **Cassandra;**
  - **Redis;**
  - **Amazon DynamoDB;**
  - **Entre outros;**

# SGBDs

- Podemos checar os diversos bandos de dados que existem e seu uso estimado no mundo, através do site *db-engines*:
  - <https://db-engines.com/en/ranking>
- Para uma explicação mais detalhada e comparativa de BDs relacionais e não relacionais:
  - <https://www.mongodb.com/nosql-explained>

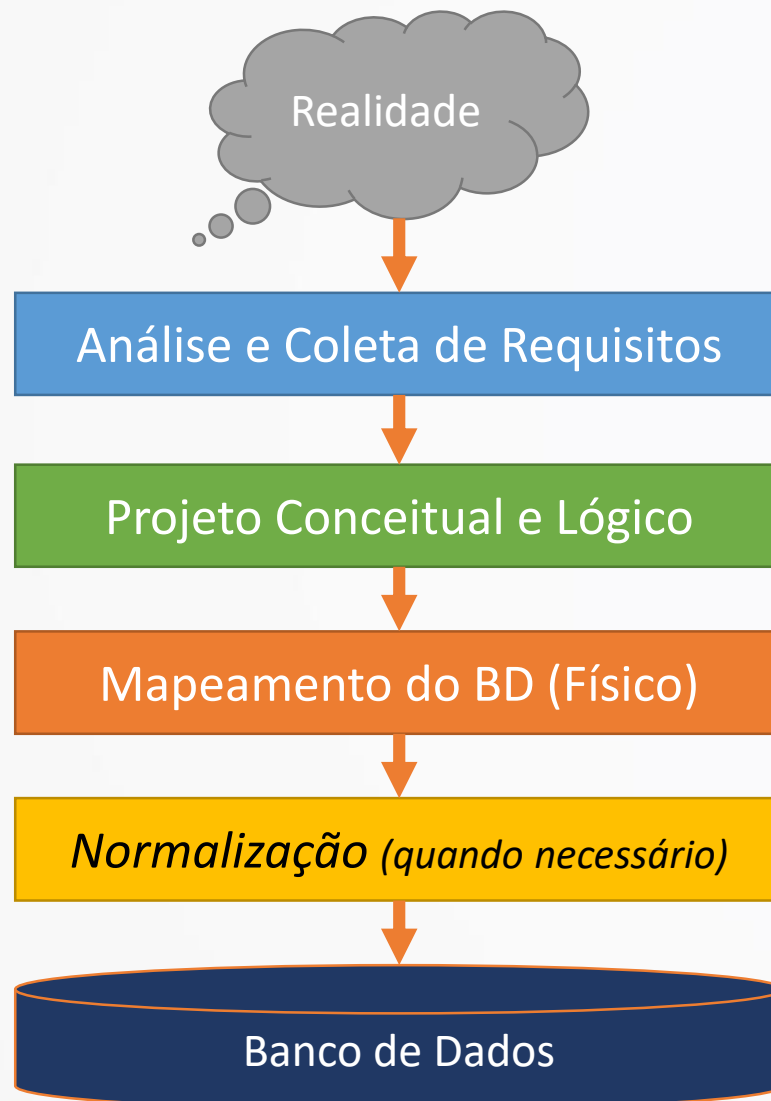
# Modelagem

- Para se começar a utilizar efetivamente um banco de dados, ele deverá passar por um processo de **concepção**.
- Como o banco de dados é o local de armazenamento de informações de um determinado **sistema de informação**, ele deverá ser criado em harmonia à este sistema.
- É na etapa de criação que o banco de dados deverá ser **modelado** e, a partir dessa modelagem conceitual, deverão ser criadas as tabelas propriamente ditas.

# Modelagem

- O **projeto de banco de dados** visa captar a realidade e através de uma sucessão de passos. Um banco de dados deve ser implementado para que represente esta realidade o máximo possível.
- Cada passo têm como resultado um documento que serve como base para a atividade seguinte.

# Esquema de modelagem



# Visão Geral de um Projeto (etapas)

## 1. Modelagem com ER (**Conceitual**)

- Modelagem de entidades (e seus atributos), relacionamentos (e suas cardinalidades)
- Proposta de solução.

## 2. Transformação entre Modelos (**Lógico**)

- Transformação de ER para Relacional

## 3. Definição do BD em SQL (**Físico**)

- Criação de Scripts DDL;



# Modelo Entidade Relacionamento (MER)

- É um **modelo conceitual** e foi desenvolvido para facilitar o projeto de banco de dados.
- Permite a especificação de um esquema que representa a **estrutura conceitual** de um banco de dados.
  - Foi projetado para estar o mais próximo possível da visão do usuário, não se preocupando em como os dados estão armazenados.
  - Mapeamento de significados e interações.
- Muito utilizado como ferramenta por muitos bancos de dados comerciais.

# Modelo Entidade Relacionamento (MER)

Conjunto de Entidades

Conjunto de Atributos

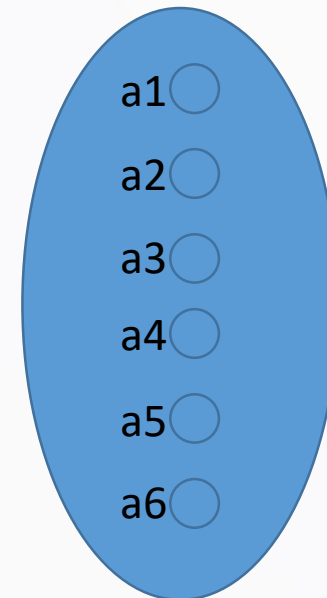
Conjunto de Relacionamentos

# Modelo Entidade Relacionamento (MER)

- Entidade
  - Pode ser vista como uma “coisa” ou “objeto” no mundo real que é distinguível de todos os outros objetos.
- Conjunto de Entidades
  - Grupo de entidades do mesmo tipo que compartilham os mesmos atributos



Aluno



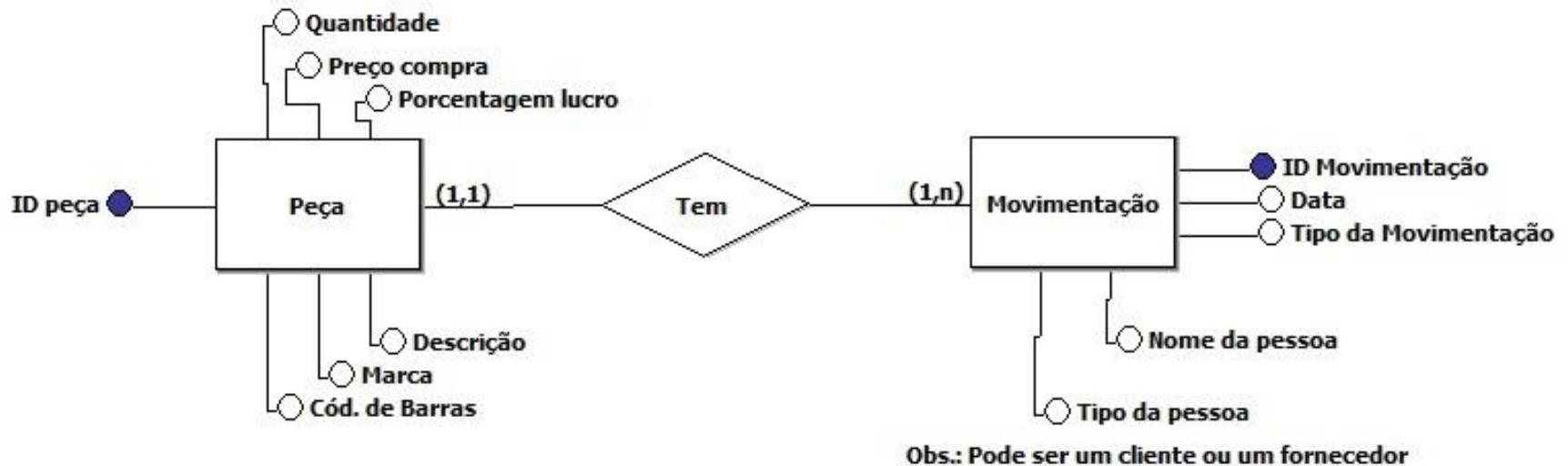
# Diagrama Entidade Relacionamento (DER)

- O MER é apenas um modelo conceitual. Aí entra o Diagrama Entidade Relacionamento (DER). Ele é a representação gráfica e principal ferramenta para aplicar os conceitos do MER.
- O diagrama é tido muitas vezes como sinônimo de modelo, uma vez que sem uma forma de visualizar as informações, o modelo pode ficar abstrato demais.

# Diagrama Entidade Relacionamento (DER)

- Exemplo de um Diagrama simples de apenas duas entidades:

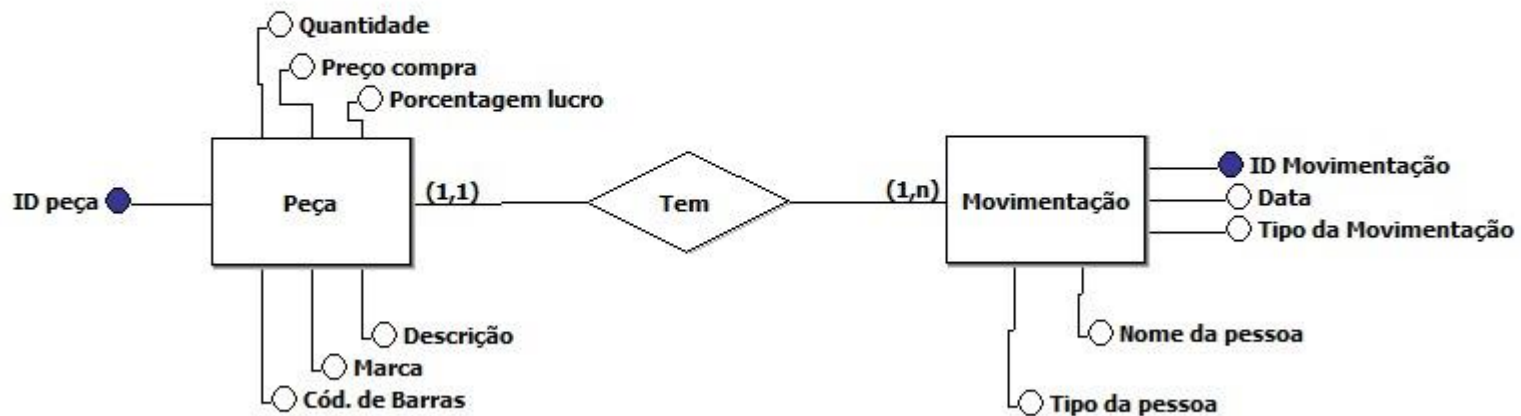
## Controle de estoque - Loja de peças



# Diagrama Entidade Relacionamento (DER)

- As “bolinhas” são os **atributos**.
- Os “retângulos” são as **entidades**.
- Os “losangos” são os **relacionamentos**.

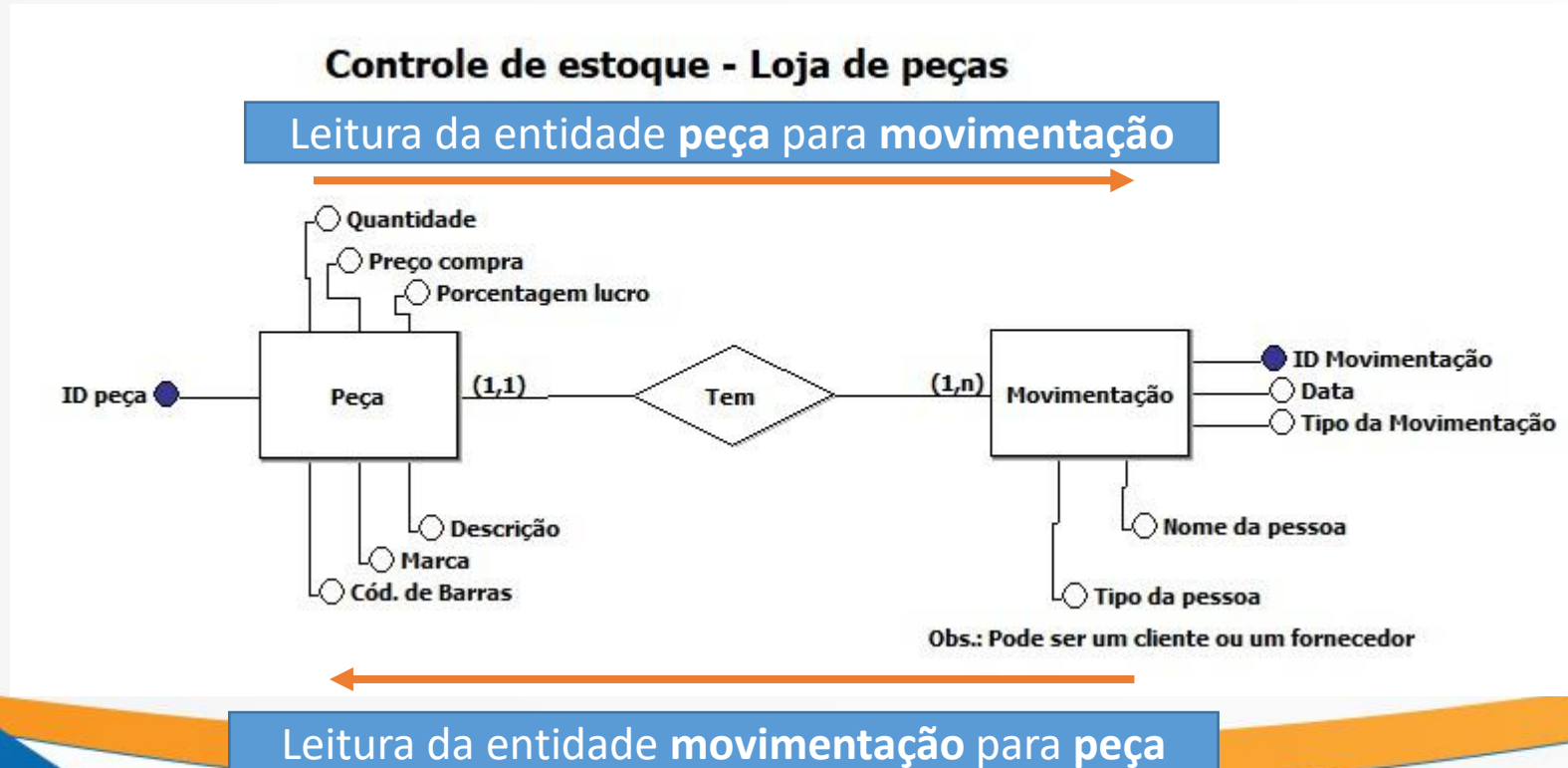
## Controle de estoque - Loja de peças



Obs.: Pode ser um cliente ou um fornecedor

# Diagrama Entidade Relacionamento (DER)

- A leitura é sempre **bidirecional**. Sempre perguntamos o máximo e o mínimo de relacionamento entre as entidades:
  - Peça tem no mínimo xxx movimentações e no máximo xxx movimentações.
  - Movimentação tem no mínimo xxx peças e no máximo xxx peças.



# Cardinalidades de Mapeamento



- Considere as seguintes questões:
  - Um empregado pode **não ter** dependentes?
    - *Cardinalidade mínima.*
  - Determinado empregado pode **possuir mais de um** dependente?
    - *Cardinalidade máxima.*
  - **Pode existir** dependente sem algum empregado associado?
    - *Cardinalidade mínima*
  - Um dependente pode **ter mais de um** empregado associado?
    - *Cardinalidade máxima*

*As respostas dependem do problema sendo modelado!!*



# Cardinalidades de Mapeamento

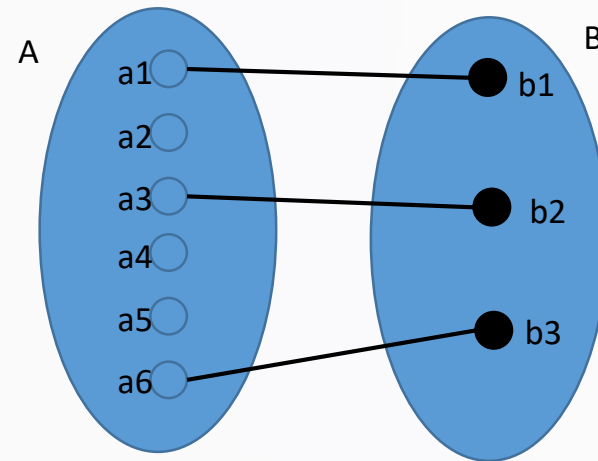
- Cardinalidades de Mapeamento ou fatores de cardinalidade
  - Expressam o número de entidades ao qual outra entidade pode ser associada por um relacionamento.
  - Entre um conjunto de entidades A e B, os relacionamentos podem ser:
    - Um-para-Um
    - Um-para-Muitos
    - Muitos-para-Muitos

*Nota: Para verificação de cardinalidade, de modo geral, é levada em conta a cardinalidade máxima.*

# Cardinalidades de Mapeamento

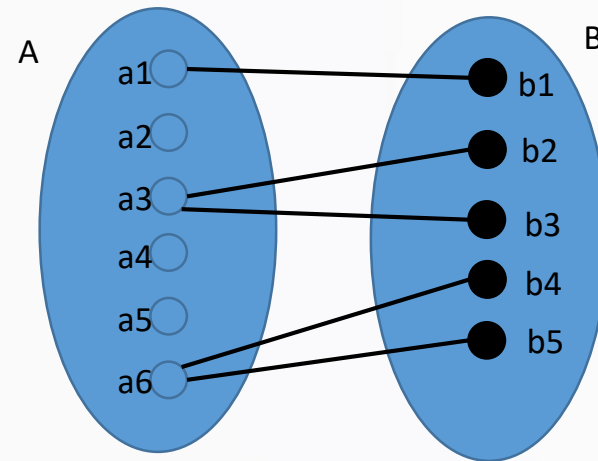
- Um-para-Um

- Uma entidade em A é associada a **no máximo** uma entidade em B.
- Uma entidade em B é associada a **no máximo** uma entidade em A.



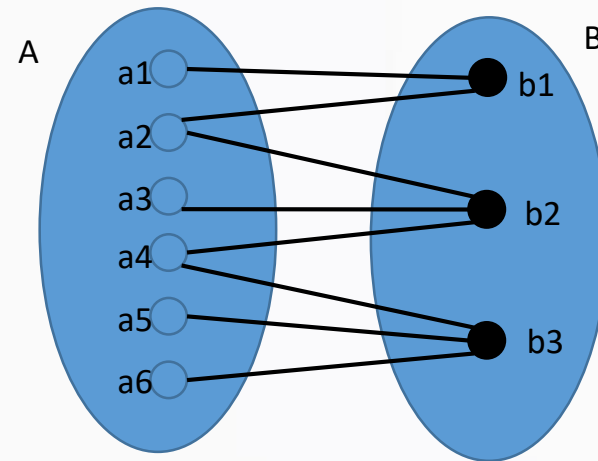
# Cardinalidades de Mapeamento

- Um-para-Muitos
  - Uma entidade em A é associada a **qualquer número de entidades** (zero ou mais) em B.
  - Uma entidade em B é associada a **no máximo** uma entidade em A.



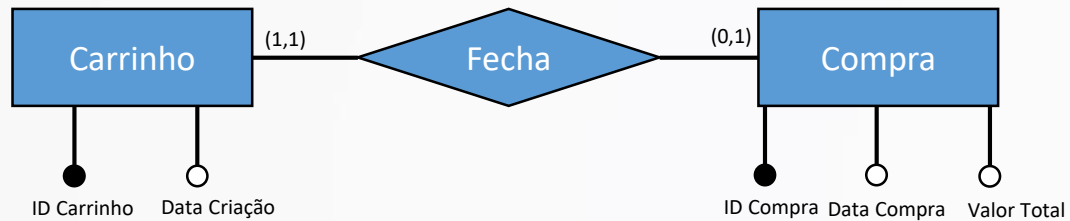
# Cardinalidades de Mapeamento

- Muitos-para-Muitos
  - Uma entidade em A é associada a **qualquer número de entidades** (zero ou mais) em B.
  - Uma entidade em B é associada a **qualquer número de entidades** (zero ou mais) em A.

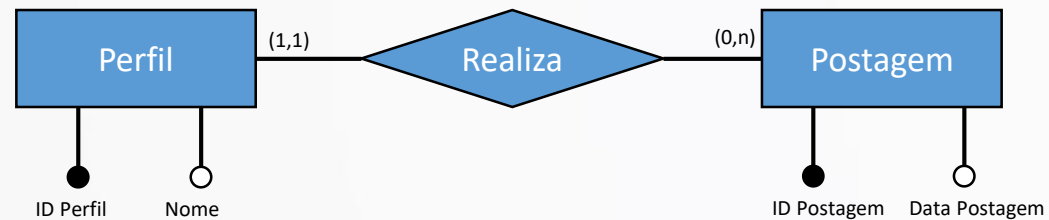


# Cardinalidades

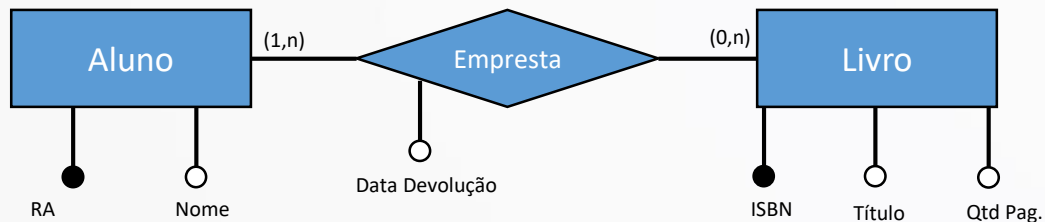
○ 0:1, 1:1



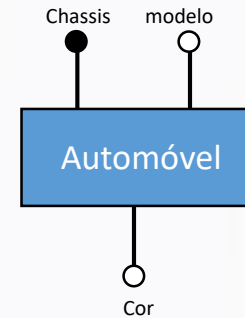
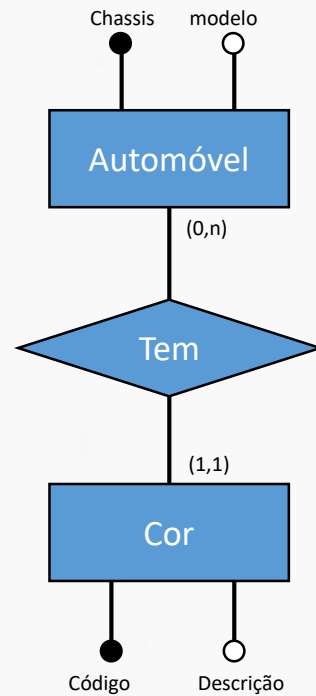
○ 0:n, 1:n



○ n:n

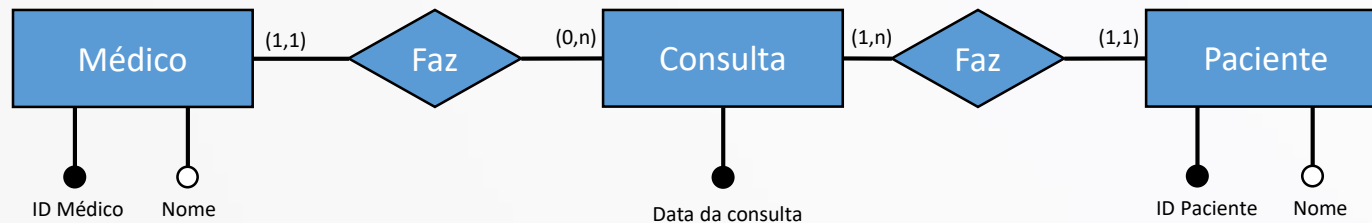
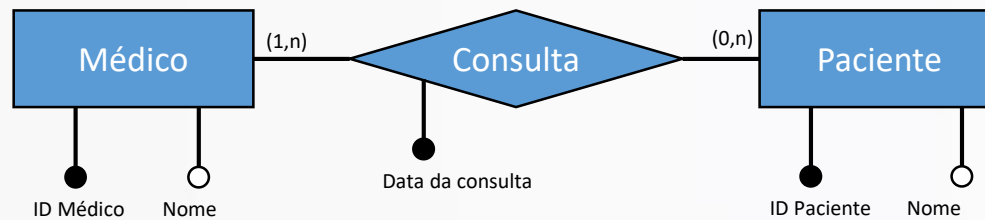


# Atributo Vs. Entidade Relacionada



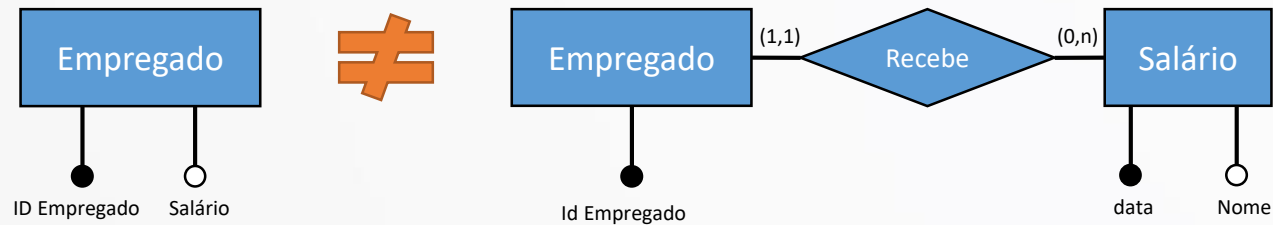
# Propriedades de Modelos ER

- O modelo ER é formal, preciso e não ambíguo
- Diferentes modelos podem ser equivalentes



# Verificação do modelo

- Modelo deve ser completo:
  - Contemplar todas as propriedades desejáveis do BD especificadas durante o projeto
  - Verificar com alguém que tem profundo conhecimento do sistema a ser implementado (especialista)
- Modelo deve refletir o aspecto temporal





# Diagrama Entidade Relacionamento (DER)

- Ferramenta para realizar o desenvolvimento dos diagramas:
  - brModelo;
- Explicação na prática:
  - Realizar a modelagem de um banco de dados simples utilizando o programa brModelo;

# Diagrama Entidade Relacionamento (DER)

- Exercícios de fixação – Criar o DER para os seguintes projetos de sistemas:
  1. Biblioteca;
  2. Loja de materiais de construção (estoque);
  3. Rede Social (Instagram);

# Diagrama Entidade Relacionamento (DER)

- Exercícios extra – Criar o DER para os seguintes projetos de sistemas:
  1. Escritório de advocacia (clientes);
  2. Corretor de imóveis;