



Universidade Federal do Ceará  
Centro de Ciências/Departamento de Computação  
Código da Disciplina: CK0084  
Professor: Ismayle de Sousa Santos

# Sistemas de Informações e Banco de Dados

Banco de Dados - conceitos, abstração e  
modelagem

---



IsmayleSantos



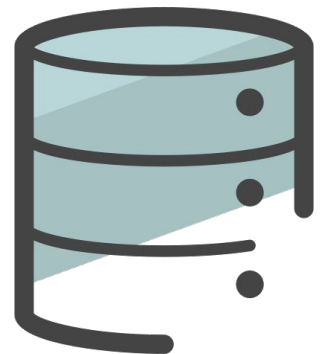
ismayle@ufc.br



@IsmayleSantos

# Atores do Sistema

Vamos começar falando dos atores envolvidos no processo de desenvolvimento de um sistema, bem como vamos aprender sobre o papel dos profissionais da área e distingui-los do usuário final



# Atores do Sistema

- Todo agrupamento de bancos de dados possui um conjunto de **atores** envolvidos na criação de um sistema
- Os atores são distintos dos usuários gerenciados pelo sistema operacional onde o servidor executa o BD
  - **Não existe apenas o usuário final**, ou seja, aquele que usa o sistema
- É necessário aprender a definir os tipos de atores e saber **descrever o propósito e funções** básicas de um

*Atores são usuários que possuem algum papel em relação ao sistema*

---

# Atores do Sistema

- Os **analistas de sistemas** são responsáveis por:
    - Identificar as necessidades dos usuários finais
  - Os **programadores** de aplicações são responsáveis por:
    - Implementar essas especificações como programas
  - Os **usuários finais** são:
    - Pessoas cujas funções exigem acesso ao banco de dados
-

# Atores do Sistema

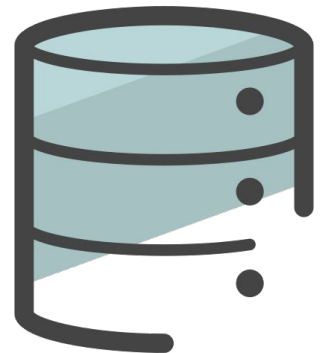
- **Projetistas e implementadores de sistema de SGBD**
    - Projetam e implementam os módulos e as interfaces do SGBD como um pacote de software
  - **Desenvolvedores de ferramentas**
    - Projeta e implanta ferramentas
  - **Operadores e pessoal de manutenção**
    - Responsáveis pela execução e manutenção do ambiente de hardware e software para o sistema de banco de dados
-

# Atores do Sistema

- **Administrador de banco de dados (DBA)** é responsável por:
    - Autorizar o acesso ao banco de dados
    - Coordenar e monitorar seu uso
    - Adquirir recursos de software e hardware
  - **Projetistas de banco de dados** são responsáveis por:
    - Identificar os dados a serem armazenados
    - Escolher estruturas apropriadas para representar e armazenar esses dados
-

# Abstração de Dados

E como apresentar o Banco de dados para diferentes atores?



# Abstração de Dados

- O sistema de banco de dados deve garantir uma visão totalmente abstrata do banco de dados para o usuário
- Uma forma de oferecer os meios necessários para alcançar a abstração é usando o **modelo de dados** para suprimir detalhes da organização e armazenamento dos dados

*Modelo de dados é uma descrição dos tipos de informações que estão armazenadas em um banco de dados*

---



# Modelo de Dados

- **Modelagem de dados** é um processo no qual você projeta ou planeja o modelo a ser utilizado para a construção da sua base de dados (banco de dados), de forma que reflita os conceitos do negócio a ser atendido
  - É importante que haja um entendimento e compreensão das **necessidades do negócio** e refleti-los no projeto de banco de dados
    - É necessário levantar os requisitos, atividade que é feita pelo analista de requisitos
-

# Níveis do Modelo de Dados

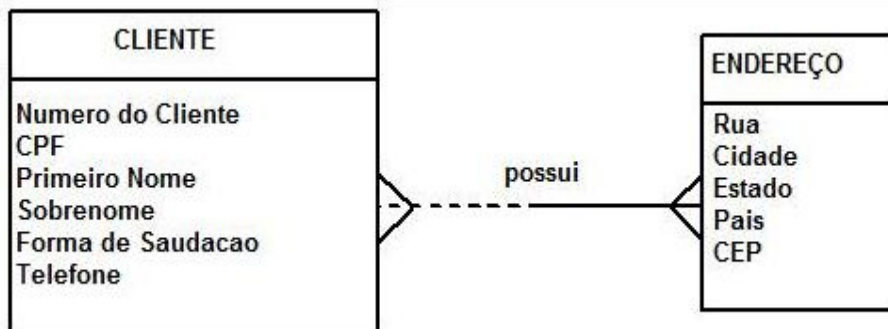
- A abstração se dá em três níveis de modelos de dados:
    - **Alto nível** ou **modelo de dados conceitual** que fornece uma visão mais próxima do modo como os usuários visualizam os dados realmente
    - **Modelos de dados lógico** que são usados para determinar se o SGBD é relacional, orientado a objetos, hierárquico, etc
    - **Baixo nível** ou **modelo de dados físico**, que fornece uma visão mais detalhada do modo como os dados estão realmente armazenados no computador
-

# Modelo de Dados Conceituais

- Representação com **alto nível** de abstração que usa os conceitos de
    - **Entidade** que representa um objeto ou conceito do mundo real
      - Ex, funcionário
    - **Atributo** que representa alguma propriedade de interesse que descreve melhor uma entidade
      - Ex, nome ou o salário do funcionário
    - **Relacionamento** entre duas ou mais entidades representa uma associação entre elas
      - Ex, um relacionamento entre um funcionário e um projeto
-

# Modelo de Dados Lógico

- Descrevem os **tipos de entidades lógicas**, os atributos de dados que descrevem essas entidades e os relacionamentos entre as entidades
  - Depende do Banco de Dados
  - Representam os dados em alguma estrutura lógica
  - Indica quais **tabelas** o banco contém e quais os nomes das **colunas**



TipoDeProduto (CodTipoProd, DescrTipoProd)

Produto (CodProd, DescrProd, PrecoProd, CodTipoProd)

CodTipoProd referencia TipoDeProduto

# Modelo de Dados Físicos

- São usados **para projetar o esquema interno de um banco de dados**, descrevendo as tabelas de dados, as colunas de dados das tabelas e o relacionamento entre as tabelas
  - Serve para definir o projeto do banco de dados
  - Define efetivamente de que maneira os dados estão armazenados
    - tipo
    - tamanho do campo
    - restrições
    - etc

```
CREATE TABLE FILME (  
  Codigo Filme INTEGER(6) NOT NULL,  
  Titulo VARCHAR(100),  
  Genero VARCHAR(50),  
  Ano INTEGER(4),  
  Duracao TIME,  
  Diretor VARCHAR(150),  
  FaixaEtaria INTEGER(2) UNSIGNED,  
  PRIMARY KEY(Codigo Filme)  
);
```

---

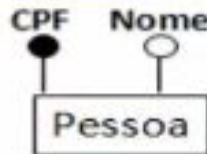
# Abstração de Dados



CPF: 123.456.789-10

Nome: Caio

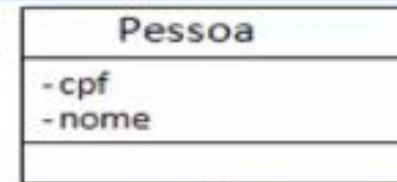
## Conceitual



## Lógico

Pessoa(CPF, Nome)

CPF	Nome
12345678910	Caio



## Físico

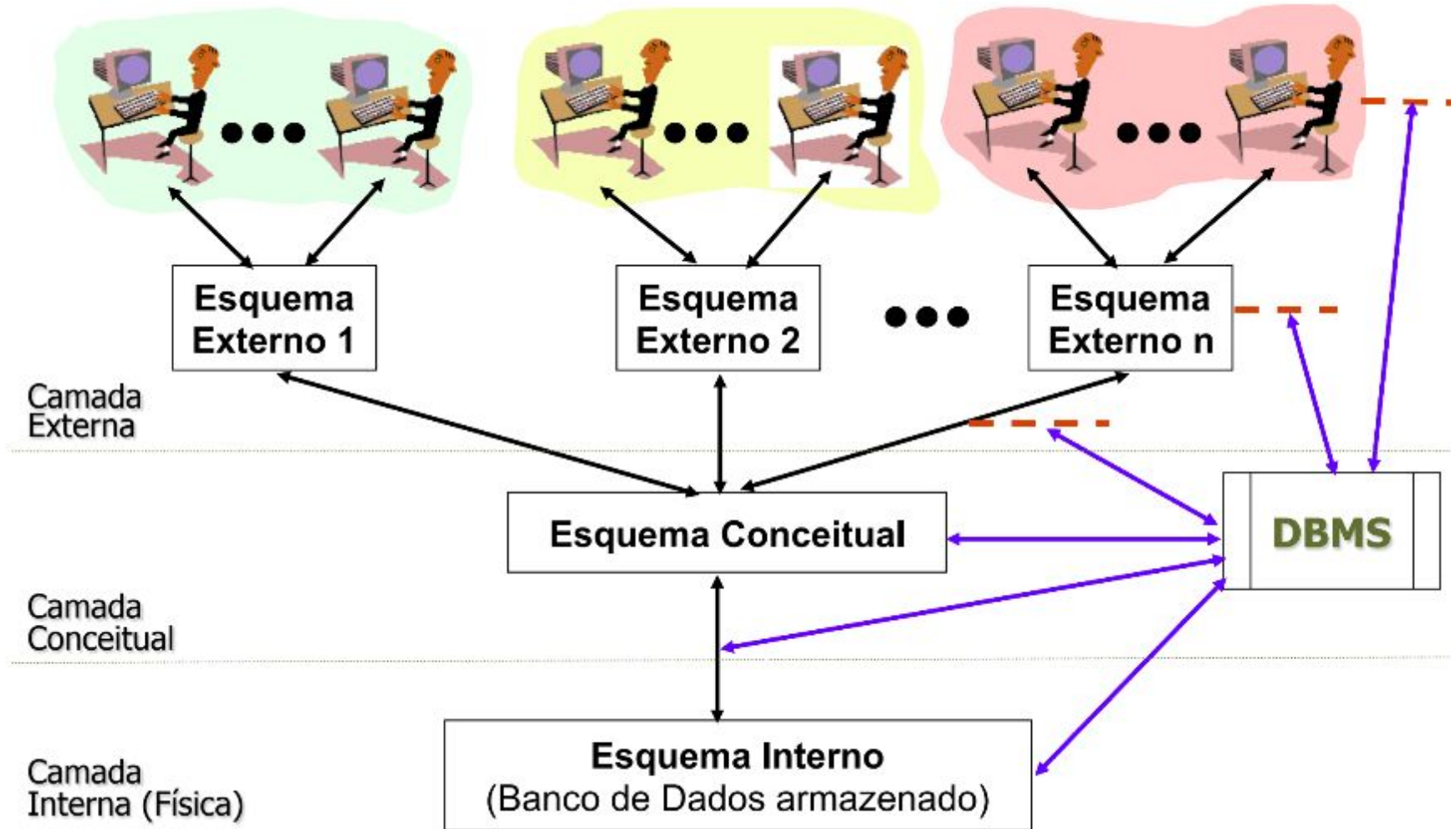
```
CREATE TABLE Pessoa(  
  CPF VARCHAR(11) NOT NULL,  
  Nome VARCHAR(30) NOT NULL,  
  PRIMARY KEY (CPF));  
  
CREATE INDEX ...
```

```
public class Pessoa {  
  private String cpf;  
  private String nome;  
  ...  
}
```

# Abstração de Dados

- Existe uma outra hierarquia, a da **Arquitetura 3 esquemas (modelos)**, onde também temos três níveis:
  - Externo, Conceitual e Físico
- O termo arquitetura em três esquemas é discutido no livro do Elmasri e Navathe (2011)
  - O objetivo dessa arquitetura é separar as aplicações do usuário do banco de dados físico
  - Não há uma padronização da nomenclatura
- As teorias de abstrações de dados foram construídas por autores diferentes e podem sofrer alterações

# Arquitetura em 3 Esquemas



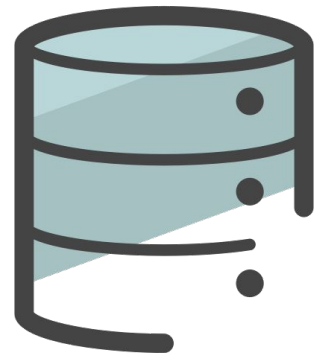


# Arquitetura em 3 Esquemas

- **Nível externo ou esquema de visão**
    - Abrange os esquemas externos ou as **visões dos usuários**
    - Descreve apenas a **parte do BD** de interesse desses usuários
  - **Nível conceitual ou esquema conceitual**
    - Têm um esquema conceitual que **descreve toda a estrutura do banco de dados** para a comunidade de usuários
  - **Nível interno ou esquema interno**
    - Têm um esquema interno que descreve a **estrutura de armazenamento físico** do banco de dados
-

# Modelo Entidade-Relacionamento (ER)

Nessa etapa da aula iremos aprender um pouco mais sobre o modelo de dados conceitual ER desenvolvido com o objetivo de facilitar o projeto de banco de dados por meio de um modelo independente de implementação e de fácil compreensão por parte do usuário



# Modelo Entidade-Relacionamento (ER)

- O modelo Entidade-Relacionamento é o modelo de dados conceitual de **alto nível** ou **modelo de dados conceitual**
  - **Relembrando...**
    - Os conceitos foram projetados para estar o mais próximo possível da visão que o usuário
  - O modelo ER é utilizado principalmente durante o processo de projeto de banco de dados
    - É modelo mais usado para a modelagem conceitual de BDs
-

# Modelo Entidade-Relacionamento (ER)

- Esse modelo descreve os **objetos** (entidades) envolvidos em um domínio de negócios, com suas **características** (atributos) e como elas se **relacionam** entre si (relacionamentos)
  - Ele representa de forma abstrata a estrutura que o banco de dados possuirá
-

# Modelo versus Diagrama

- O **modelo ER** pode ficar abstrato demais para auxiliar no desenvolvimento do sistema, uma vez que ele é apresentado em forma de texto
    - Por isso, o diagrama é tido muitas vezes como sinônimo de modelo
  - O **diagrama** facilita a comunicação entre os integrantes da equipe, pois oferece uma linguagem comum utilizada tanto pelo analista, responsável por levantar os requisitos, e os desenvolvedores, responsáveis por implementar aquilo que foi modelado
-

# Diagrama Entidade Relacionamento

- A notação original foi proposta por Peter Chen (1976)
  - Notação do Diagrama ER
    - As **entidades** representadas por retângulos
    - Os **atributos** aparecem listados na própria entidade
      - originalmente como elipses
    - Os **relacionamentos** por losangos, ligados às entidades por linhas, contendo a cardinalidade (1..1, 1..n ou n..n)
    - **Cardinalidade** é a quantidade máxima de ocorrência de entidades que podem estar associadas
    - As **chaves primárias** são representadas por \* ou sublinhado
-

# Notação de Peter Chen (1976)

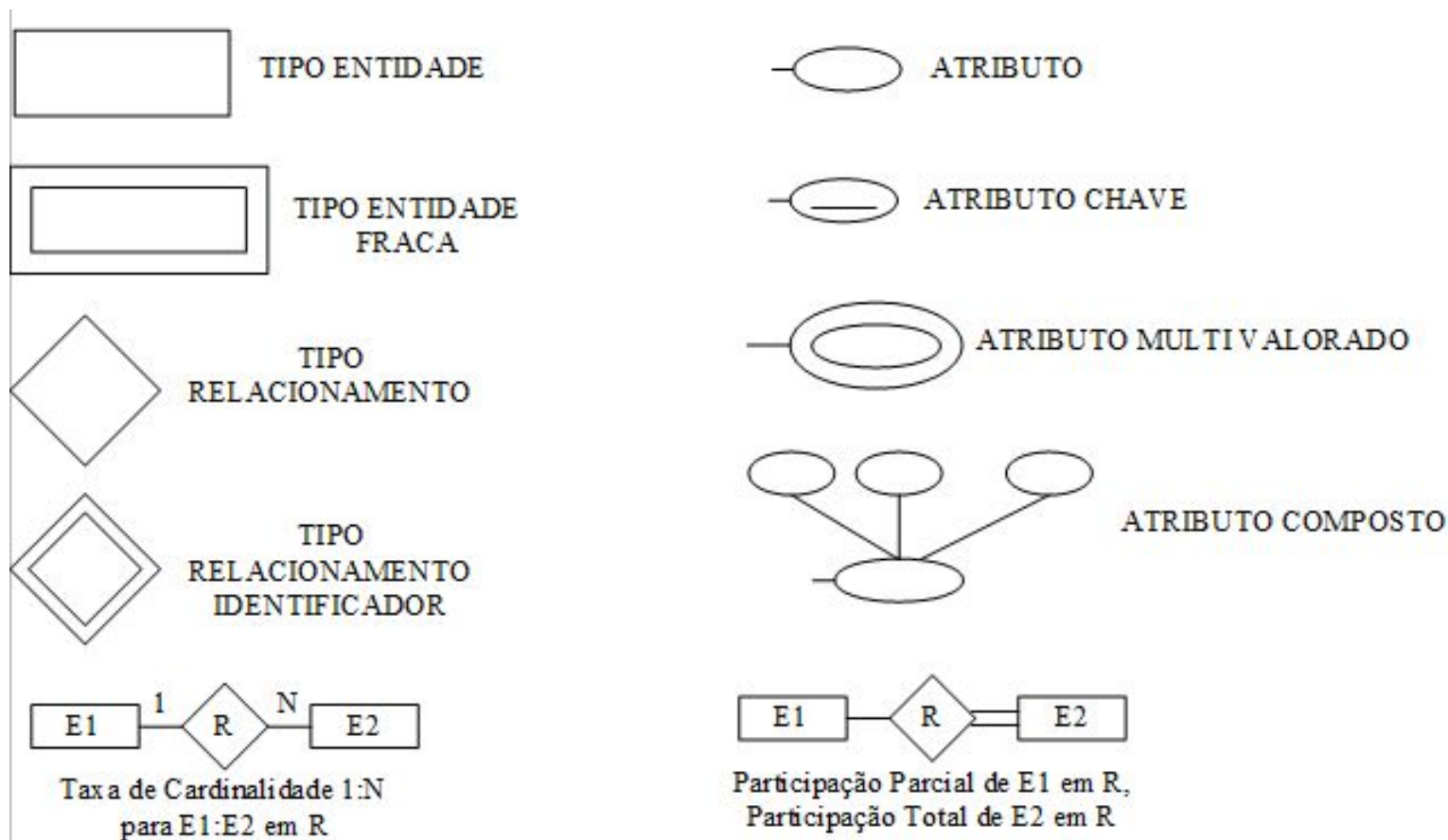
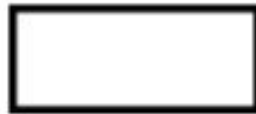


Figura 11- Objetos que Compõem o Diagrama ER

# Variação da Notação Chen (1976)

- Baseado na UML

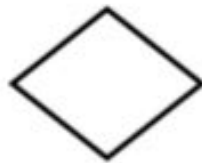


**Entidade**

objeto" do mundo real :  
um ser, um fato, coisa,  
organismo social, etc.

**Atributo**

informações que se deseja  
guardar sobre o objeto



**Relacionamento**

associação existente entre  
elementos de entidades

1 - 1  
1 - N  
N - N

**Cardinalidade**

número de ocorrências  
possíveis de cada entidade  
envolvida num relacionamento

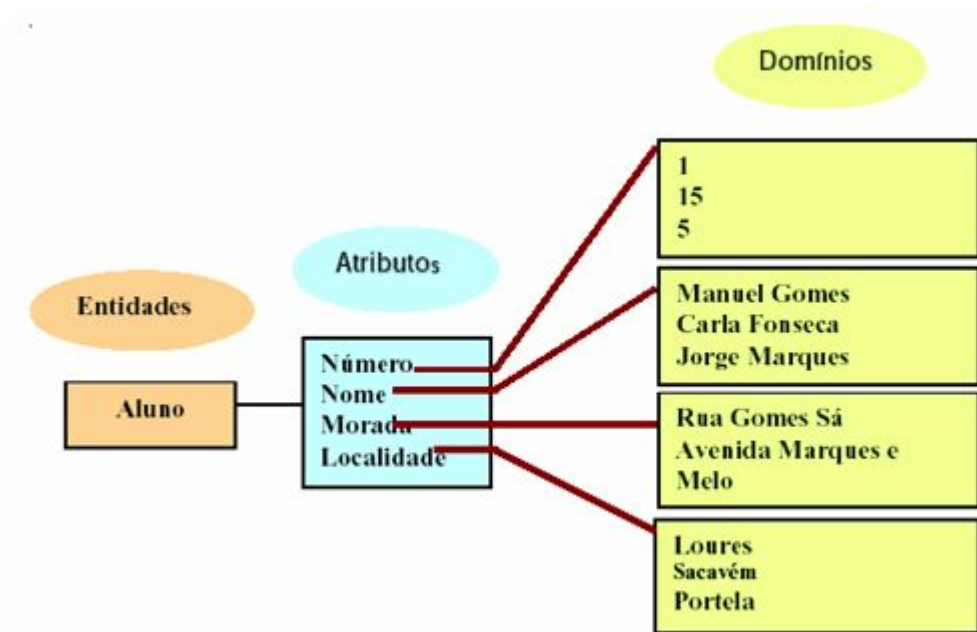


# Entidades

- O objeto básico tratado pelo modelo ER é a “entidade”, que pode ser definida como um objeto do mundo real, concreto ou abstrato e que possui existência independente
  - As entidades podem ser classificados como:
    - **Física** - são aquelas realmente tangíveis, existentes e visíveis no mundo real (tangíveis ou palpáveis)
      - Exemplo: uma pessoa ou uma casa
    - **Lógica** - são aquelas que existem geralmente em decorrência da interação entre entidades físicas
      - Exemplo: um projeto ou uma disciplina
-

# Entidades

- As entidades são nomeadas com substantivos concretos ou abstratos que representem de forma clara sua função dentro do domínio
  - Exemplo: Cliente, Produto, Venda, Turma, etc...



# Entidades

- São classificadas segundo o motivo de sua existência:
    - **Entidades fortes** - são aquelas que por si só elas já possuem total sentido de existir.
      - Ex. funcionário
    - **Entidades fracas** - são aquelas que dependem de outras entidades para existir
      - Não possui atributo chave
      - Ex.: dependente
    - **Entidades associativas** - acontecem quando há um relacionamento do tipo muitos para muitos
      - É necessária a criação de uma entidade intermediária cuja identificação é formada pelas chaves primárias
-

# Atributos

- Atributos são as características que descrevem cada entidade dentro do domínio
    - Exemplo: nome da pessoa, endereço, função, etc...
  - Os atributos podem ser divididos quanto a estrutura:
    - **Atributo composto** - é um atributo pode ser dividido em diversas subpartes com significado independente entre si
      - Exemplos: nome, endereço, etc...
    - **Atributo simples ou atômico** - é um atributo que não pode ser subdividido
      - Exemplos: sexo, cpf
-

# Atributos

- Os atributos podem ser classificados quanto à sua função:
  - **Descritivos** - representam característica intrínsecas de uma entidade, tais como nome ou cor
  - **Nominativos** - além de serem também descritivos, estes têm a função de definir e identificar um objeto, tais como nome, código ou número
  - **Referenciais** - representam a ligação de uma entidade com outra em um relacionamento
    - Exemplo: uma venda possui o CPF do cliente, que a relaciona com a entidade cliente

# Atributos

- Os atributos ainda são divididos em:
    - **Atributo simplesmente valorado**
      - São atributos que podem assumir apenas um determinado valor em uma determinada instância
        - Ex.: nome
    - **Multi valorado**
      - São atributo que podem assumir diversos valores em uma mesma instância
        - Ex. Telefone
    - **Atributo derivado**
      - É um atributo que é gerado a partir de outro atributo e não precisa ser armazenado
        - Ex.: Idade (se no banco tem a data de nascimento)
-

# Entidades e Atributos

- Alguns atributos representam valores únicos que identificam a entidade dentro do domínio e não podem se repetir
    - Em um cadastro de clientes, por exemplo, esse atributo poderia ser o CPF
      - **Chave Primária**
  - Atributos referenciais são chamados de **Chave Estrangeira** e geralmente estão ligados à chave primária da outra entidade
    - A chave estrangeira permite a referência a registros de outras tabelas
-

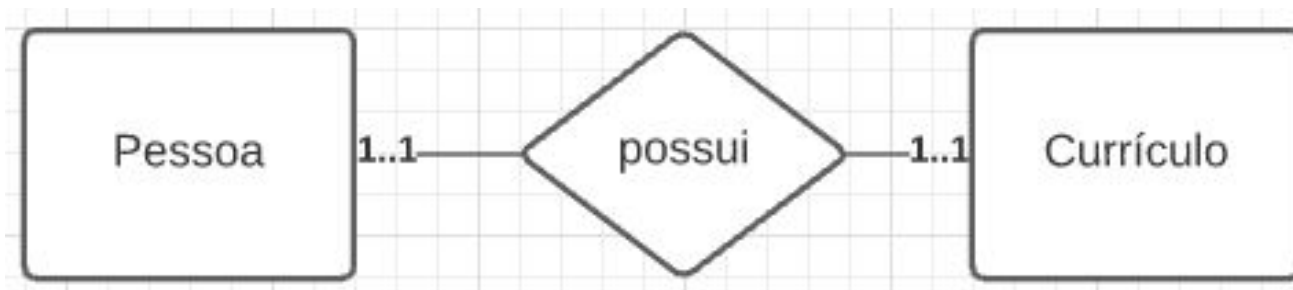
# Instâncias de Relacionamento

- Uma vez que as entidades são identificadas, deve-se então definir como se dá o relacionamento entre elas
  - De acordo com a quantidade de objetos envolvidos em cada lado do relacionamento, podemos classificá-los de três formas:
    - Relacionamento 1..1 (um para um)
    - Relacionamento 1..n ou 1..\* (um para muitos)
    - Relacionamento n..n ou \*..\* (muitos para muitos)
-



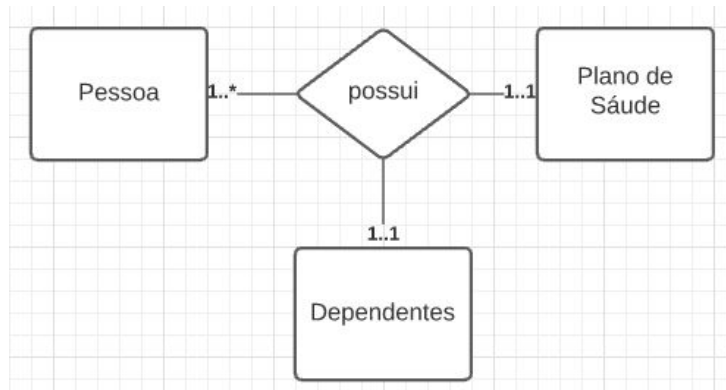
# Relacionamento 1.1

- Cada uma das duas entidades envolvidas referenciam obrigatoriamente apenas uma unidade da outra
- Por exemplo, em um banco de dados de currículos, cada usuário cadastrado pode possuir apenas um currículo na base, ao mesmo tempo em que cada currículo só pertence a um único usuário cadastrado



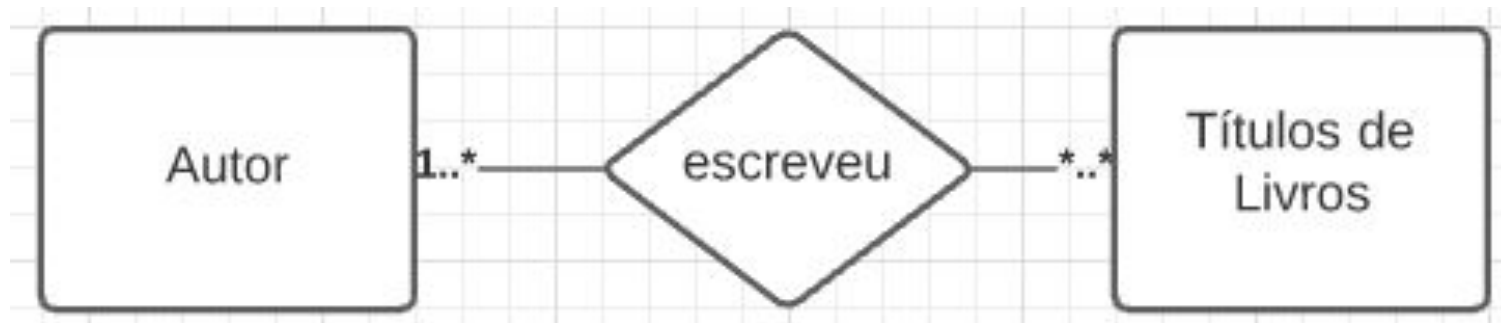
# Relacionamento 1..n ou 1..\*

- Uma das entidades envolvidas pode referenciar várias unidades da outra, porém, do outro lado cada uma das várias unidades referenciadas só pode estar ligada uma unidade da outra entidade
- Por exemplo, em um sistema de plano de saúde, um usuário pode ter vários dependentes, mas cada dependente só pode estar ligado a um usuário principal



## Relacionamento n..n ou \*.\*

- Cada entidade, de ambos os lados, podem referenciar múltiplas unidades da outra
- Por exemplo, em um sistema de biblioteca, um título pode ser escrito por vários autores, ao mesmo tempo em que um autor pode escrever vários títulos



# Relacionamentos

- Os relacionamentos participam apenas de uma entidade de cada tipo entidade, porém, uma entidade pode participar de mais do que um relacionamento
  - Os relacionamentos em geral são nomeados com **verbos** ou **expressões** que representam a forma como as entidades interagem, ou a ação que uma exerce sobre a outra
  - Essa nomenclatura pode variar de acordo com a direção em que se lê o relacionamento
    - Por exemplo: um autor escreve vários livros, enquanto um livro é escrito por vários autores
-

# Relacionamentos

**Relacionamento binário:**



**Relacionamento ternário:**

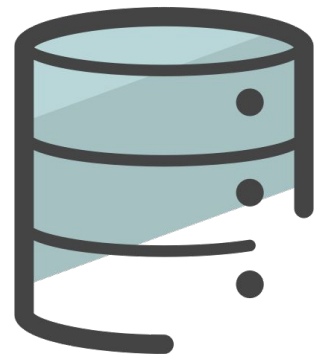


**Relacionamento n-ário:**



# Ferramentas para Modelagem de BDs

Essa parte da aula iremos explorar ferramenta online e gratuita Lucidchart para montarmos um Modelo ER simples

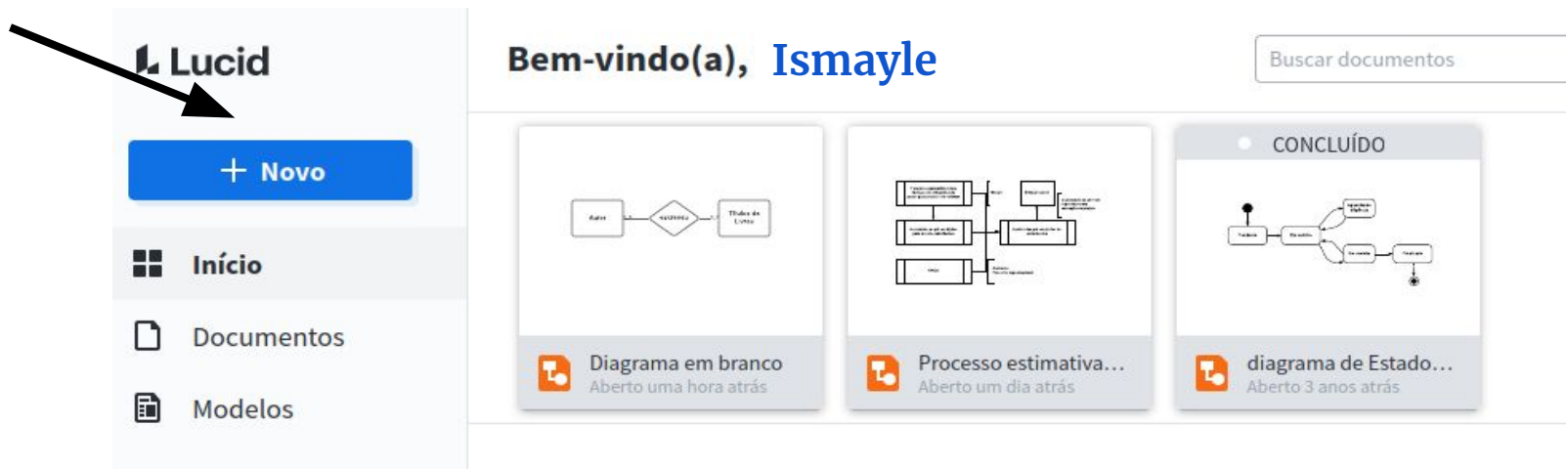


# Ferramentas para Modelagem de BDs

- Modelar e desenvolver um banco de dados é imprescindível para o sucesso de todo o sistema
  - Abaixo estão descritas ferramentas que auxiliam na construção de um diagramas ER:
    - MySQL Workbench
    - Astah Professional
    - DBDesigner 4
    - SQL Power Architect
    - Draw.io
    - [Lucidchart \(on line\)](#)
-

# Lucidchart

- A partir de agora vamos modelar um exemplo na ferramenta on line Lucidchart
- Vamos todos entrar com o usuário do Google (mais fácil) na página inicial do Lucidchart e criar um novo documento
  - <https://www.lucidchart.com/pages/pt>





# Lucidchart

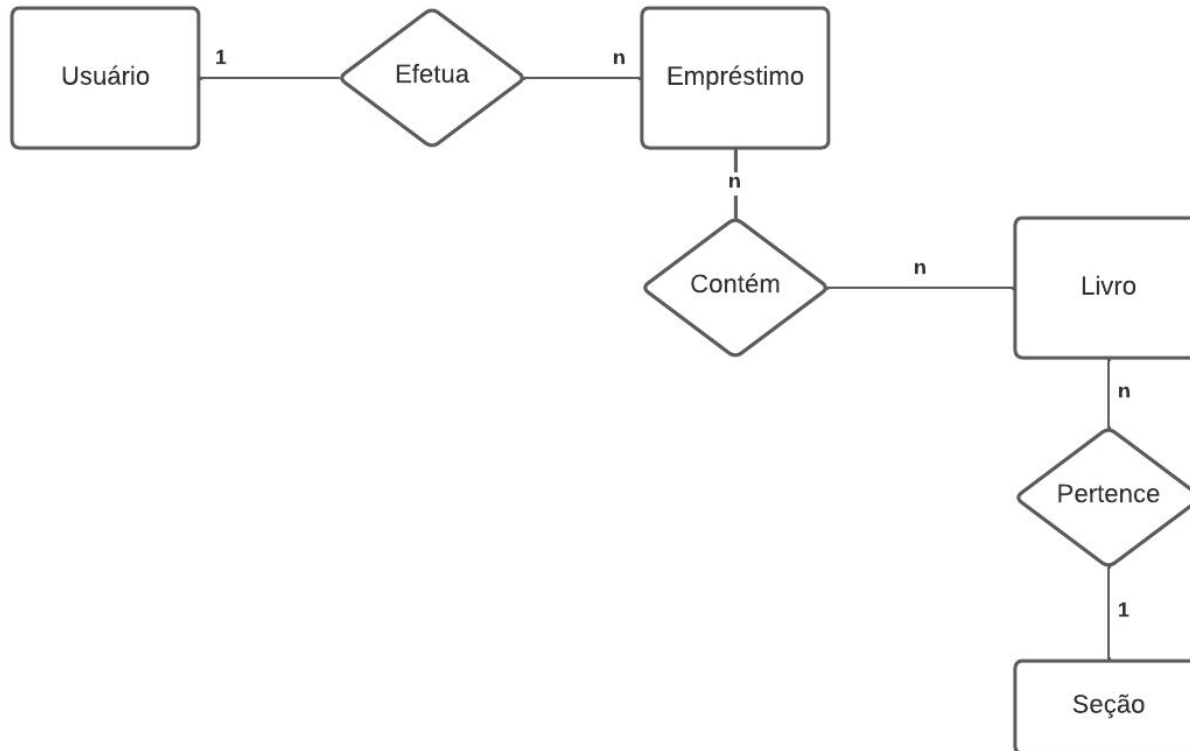
- Vamos modelar um sistema de bibliotecas, focando especificamente no empréstimo de livros
  - As entidades físicas existentes são o **Usuário** da biblioteca e o **Livro** que será emprestado, o qual fica em uma **Seção** da biblioteca
  - A entidade lógica **Empréstimo**, que está relacionada tanto com o usuário quanto com o livro

**Vamos desenhar!**

---

# Lucidchart

- É pra ficar assim!



# Lucidchart

- Você pode selecionar um modelo

## Modelos

Os documentos criados com template irão para: [Meus documentos](#)



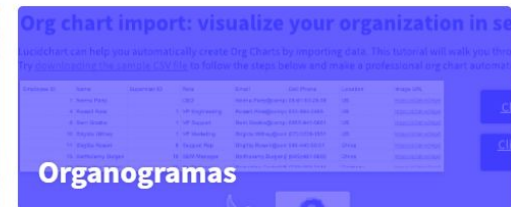
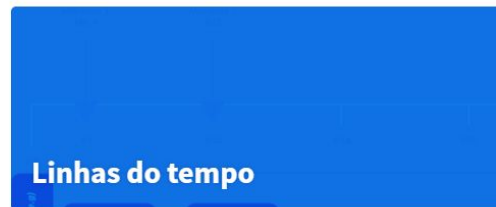
Lucidchart Lucidspark USAR

Pesquisar modelos...



### Recomendado para você

- ▼ Padrão
  - ▶ Trabalhe remotamente
  - ▶ Construir fluxogramas
  - ▶ Planejar e acompanhar pr...
  - ▶ Visualizar sistemas técnicos
  - ▶ Brainstorming de ideias
  - ▶ Organizar pessoal
  - ▶ Design de interface do usu...
  - ▶ Aumentar a eficiência das ...



# *Obrigado!*

## *Por hoje é só pessoal...*

### **Dúvidas?**



**IsmayleSantos**



**ismayle@.ufc.br**



**@IsmayleSantos**

---