

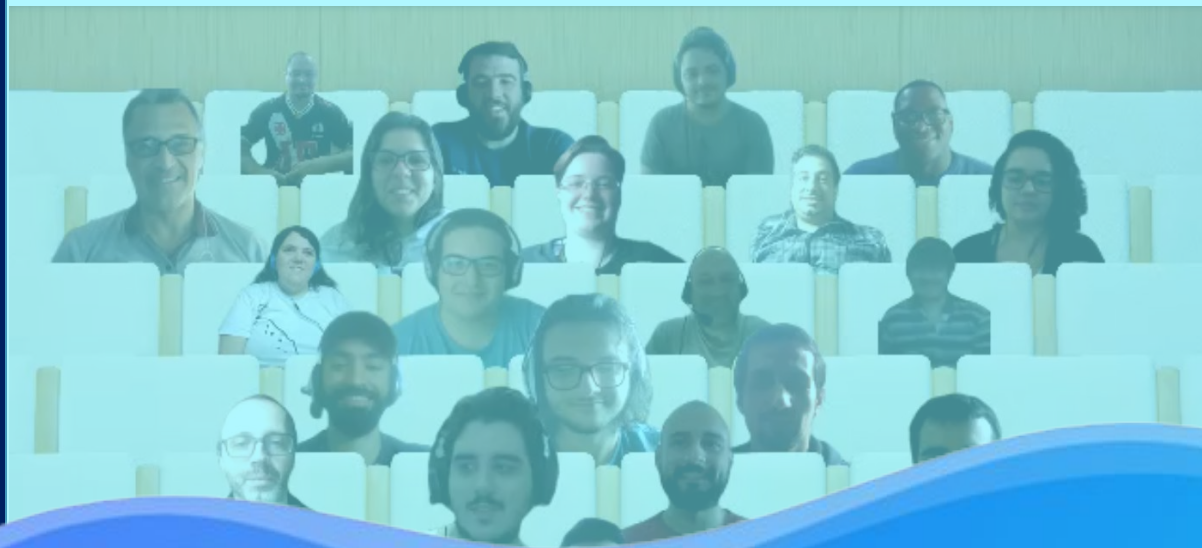
53 45 52 45 49 20 46 49 45 4c 20  
41 4f 53 20 50 52 45 43 45 49 54  
4f 53 20 44 41 20 48 4f 4e 52 41  
20 45 20 44 41 20 43 49 c3 8a 4e  
43 49 41 2c 20 50 52 4f 4d 4f 56  
45 4e 44 4f 20 4f 20 55 53 4f 20  
45 20 4f 20 44 45 53 45 4e 56 4f  
4c 56 49 4d 45 4e 54 4f 20 44 41  
20 49 4e 46 4f 52 4d c3 81 54 49  
43 41 20 45 4d 20 42 45 4e 45 46  
c3 8d 43 49 4f 20 44 4f 20 43 49  
44 41 44 c3 83 4f 20 45 20 44 41  
20 53 4f 43 49 45 44 41 44 45 2e

## RESIDÊNCIA DE SOFTWARE

**CAPACITAR**  
**TREINAR**  
**EMPREGAR**  
**TRANSFORMAR**



## Banco de Dados



Introdução

Roni Schanuel  
30-03-2022

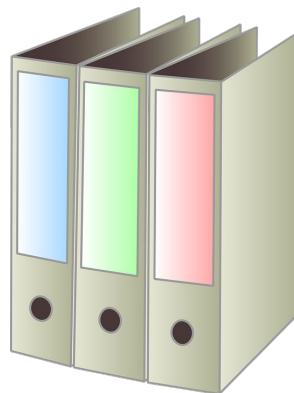
# Objetivos da Disciplina

- Entender a estrutura e os conceitos base de um Sistema Gerenciador de Banco de dados (SGBD)
- Aprender na prática a utilizar a linguagem de pesquisa declarativa padrão para banco de dados relacional, o SQL
- Tornar o aluno capaz de criar e manipular objetos de banco de dados



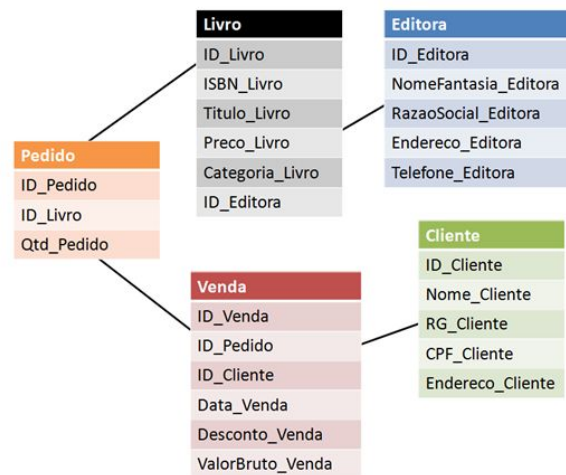
# O que são Banco de Dados ?

- Um banco de dados é uma coleção de dados inter-relacionados, que representam informações sobre um domínio específico.
- São coleções organizadas de dados que se relacionam de forma a criar algum sentido (Informação) e dar mais eficiência durante uma pesquisa ou estudo científico.



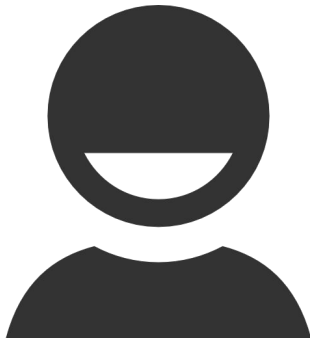
# O que são Banco de Dados ?

- Um banco de dados é uma coleção de dados inter-relacionados, que representam informações sobre um domínio específico.
- Representa alguns aspectos do mundo real, sendo chamado, às vezes, de mini-mundo
- São coleções organizadas de dados que se relacionam de forma a criar algum sentido (Informação) e dar mais eficiência durante uma pesquisa ou estudo científico.



# Sistema de banco de dados

- 4 componentes : Hardware, Software, Dados e Usuários



## **Por que usar bancos de dados?**

- Organização e tratamento estruturado dos dados
- Padronização e eficiência do acesso, útil para o desenvolvimento de aplicações
- Independência de dados em relação às aplicações
- Facilidade de recuperação de informações
- Consistência de acordo com as regras de negócio definidas
- Controle de redundância
- Restrições de integridade
- Backup e restauração
- Controle de concorrência

# Histórico

## **Antes de 1960**

- Registros eram feitos em papéis (listas, arquivos, jornais)
- Acesso difícil e trabalhoso
- Problemas de segurança, registros fora do lugar, incêndios

## 1960s

- O uso de computadores se tornou uma opção viável para organizações privadas
- Dados passam a ser armazenados em arquivos
- Surgimento dos modelos hierárquico (IMS) e rede (CODASYL)

# Histórico

## 1970

- Surgimento do modelo relacional (E.F. Codd, pesquisador da IBM)
- Separação dos níveis lógico e físico do esquema do banco de dados
- Baseado em álgebra relacional
- Se tornou o princípio padrão para sistemas de banco de dados
- Representado por uma coleção de tabelas (entidade/relação) e um conjunto de linhas (tuplas) e colunas (atributos)

Entre 1974 e 1977 Surgimento de dois sistemas de banco de dados

- INGRES, Universidade da Califórnia, Berkeley
- System R, IBM, San Jose

1976 Criação do modelo entidade-relacionamento (Dr. Peter Chen)

- Foco na aplicação dos dados ao invés da estrutura lógica das tabelas



# Histórico

## 1980

- A linguagem SQL (Structured Query Language) se tornou o padrão para consultas em bancos de dados

relacionais (ISO e ANSI)

- Surgimento de novos SGBDs (DB2, ORACLE, PostgreSQL, Dbase)

1985 Surgimento de bancos de dados orientados a objetos

- Objeto: classes, atributos, métodos, herança

## 1990

- Esforço para a criação de bancos de dados que estendem a teoria relacional com formas mais naturais de armazenar objetos

1995 Surgimento das primeiras aplicações na Internet com acesso à bancos de dados

- Demanda maior por arquitetura cliente/servidor (MySQL)

# Histórico

## 2000 até agora

- Surgimento de abordagens NoSQL (não só SQL ou não relacional)
- Orientados a documentos, como XML, Json, MongoDB
- Dados semi-estruturados
- Orientados a grafos: Neo4J
- Relacionamentos são mais naturais (vértices e arestas)
- Chave-valor: Redis
- Conjunto de pares de chave-valor em que uma chave funciona como um identificador exclusivo

# Sistema Gerenciador de Banco de Dados ( SGBD )

- O SGBD é um software que fornece uma interface entre o usuário e os dados armazenados no banco de dados
- O SGBD possui recursos para manipular as informações do banco de dados e interagir com o usuário ou outra aplicação.



# Sistema Gerenciador de Banco de Dados ( SGBD )

- Para que uma pessoa ou aplicação possa consultar ou operar sobre o banco de dados uma linguagem de interação é necessária.
- A linguagem de consulta utilizada dependerá do **modelo** do banco de dados.
- Neste curso estudaremos os banco de dados relacionais, e neste tipo de banco de dados, a linguagem de consulta amplamente utilizada é o **SQL**.

# Ranking SGDB

<https://db-engines.com/en/ranking>

## **Profissões Banco de Dados**

### **Administrador de Banco de Dados(DBA)**

Responsável por manter a disponibilidade, segurança, integridade, recuperação e otimização do banco de dados.

### **Administrador de Dados(AD)**

O AD se dedica mais aos dados em si onde há um enfoque maior em projetar e como obter as informações.

### **Analista de BI(Business Intelligence)**

Responsável por montar os modelos de negócio, levantamento de requisitos, organização e análise das informações.

### **Cientista de Dados**

Especialista com habilidade para analisar grande volume de dados e interpretar informações de valor e apoiar na tomada de decisão dos negócios.

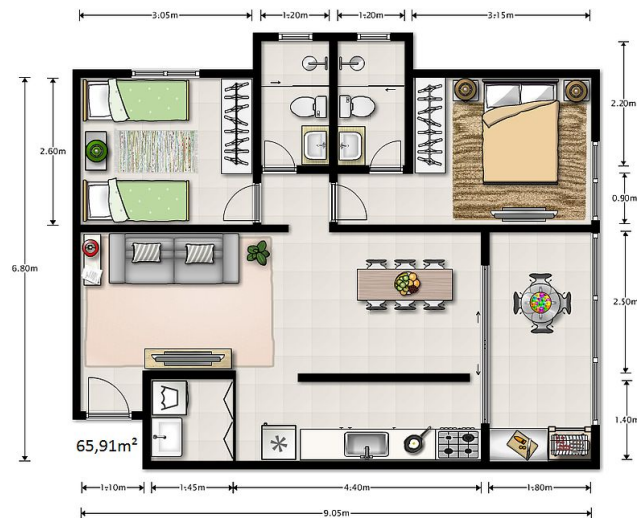


Mas vamos deixar a linguagem para depois

# Modelo de Banco de Dados

- Um modelo de banco de dados é um modelo lógico de representação de dados. No modelo, não temos que nos preocupar com questões físicas, como formato dos dados, etc.
- Se pensarmos no mundo real, podemos encontrar diversos tipos de modelos

$$Y_i = f(X_{1i}, X_{2i}, \dots, X_{ki}) + \mu_i \quad i = 1, 2, \dots, n$$





# **Modelo de Banco de Dados**

**1. Conceitual**

**2. Lógico**

**3. Físico**

## **1. Conceitual**

- Primeira fase da modelagem
- Faz o levantamento dos dados necessários
- Representação do modelo em um alto nível de abstração, independente do SGBD que será utilizado
- Registra quais dados podem aparecer no banco, mas não registra como estes dados estão armazenados no SGBD

### **Exemplos**

#### **Cadastro de alunos em um curso**

- Dados necessários: nome, nascimento, endereço e etc.

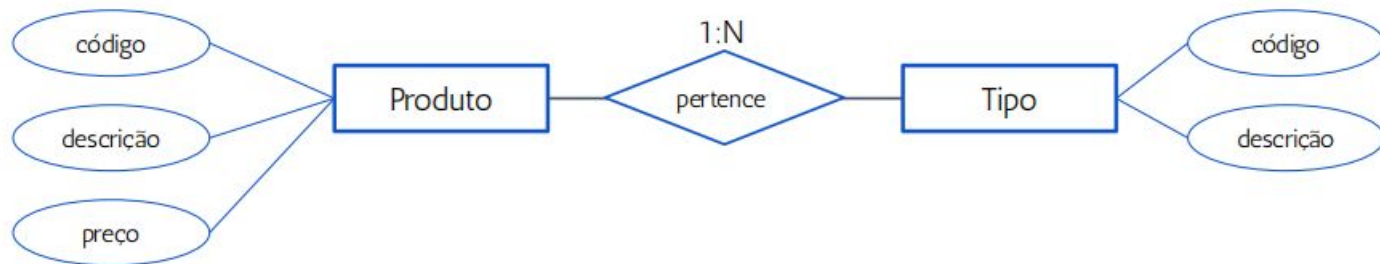
#### **Cadastro de pedidos**

- Dados necessários: código do produto, quantidade, código do cliente, código do vendedor

## 2. Lógico

O modelo lógico implementa recursos como adequação de padrão e nomenclatura, define as chaves primárias e estrangeiras, normalização, integridade referencial, entre outras. Para o modelo lógico deve ser criado levando em conta os exemplos de modelagem de dados criados no modelo conceitual.

- Descreve as estruturas que serão armazenadas no banco de dados, consiste na especificação lógica dos dados.
- Utiliza representação gráfica dos dados de uma maneira lógica, nomeando os componentes e ações que exercem uns sobre os outros
- Temos informações em um nível de detalhe maior
- Ainda é independente do SGBD



### 3. Físico

- Descreve de um banco de dados no nível de abstração visto pelo usuário do SGBD
- Diferente do conceitual e lógico, esse depende do SGBD que está sendo usado
- Indica quais tabelas, campos, tipos de valores, etc. serão utilizados

```
CREATE TABLE turma (  
    idturma INTEGER(4) NOT NULL AUTO_INCREMENT,  
    capacidade INTEGER(2) NOT NULL,  
    idProfessor INTEGER(4) NOT NULL,  
    PRIMARY KEY (idturma),  
    FOREIGN KEY(idProfessor) REFERENCES professor(idProfessor),  
    UNIQUE KEY idturma (idturma)  
)
```

```
CREATE TABLE professor (  
    idProfessor INTEGER(4) NOT NULL AUTO_INCREMENT,  
    telefone INTEGER(10) NOT NULL,  
    nome CHAR(80) COLLATE NOT NULL DEFAULT '',  
    PRIMARY KEY (idProfessor),  
    FOREIGN KEY(idTurma) REFERENCES turma(idturma),  
    UNIQUE KEY idProfessor (idProfessor)  
)
```

## MER

É um modelo de dados conceitual que representa de forma sistemática um modelo de negócios

- Descreve os objetos envolvidos no domínio (mini-mundo)  
Entidades, atributos e relacionamentos
- É uma forma abstrata de indicar qual vai ser a estrutura do banco de dados
- O diagrama entidade-relacionamento (DER) é utilizado para representar graficamente esses objetos

Componentes do DER



Entidade



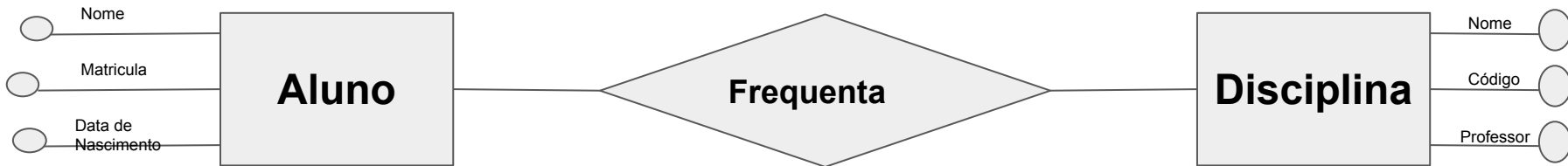
Atributo



Relacionamento

# Modelo Entidade-Relacionamento

- Antes de implementarmos o Banco de Dados, é comum passarmos por uma fase de Análise, onde geralmente utiliza-se uma representação gráfica das **entidades** envolvidas e seus **relacionamentos**.
- O modelo Entidade-Relacionamento é baseado em símbolos gráficos que representam as **entidades** e seus **atributos**, e os **relacionamentos** entre as entidades.



# Entidade

- Uma Entidade é um conjunto de objetos que deseja-se manter informações no banco de dados
- Em geral é representado por um substantivo no singular e cada entidade deve representar uma única “coisa”
- Sua representação no MER é dada por retângulos dentro dos quais deve ser colocado o nome da entidade.

**Aluno**

**Disciplina**

**Sala**

**Professor**

# Entidade

## **Boas práticas**

- ❖ Começar com uma letra e estar no singular
- ❖ De preferência letras minúsculas
- ❖ Sem espaços ou caracteres especiais



# Entidade

## Forte

Sua existência independe de outras entidades. São entidades que por si só possuem total sentido de existir.



## Fraca

Precisa de outra entidade para garantir a sua existência. O identificador de uma entidade fraca possui em sua composição o(s) atributo(s) identificador(es) da entidade forte à qual está associada.

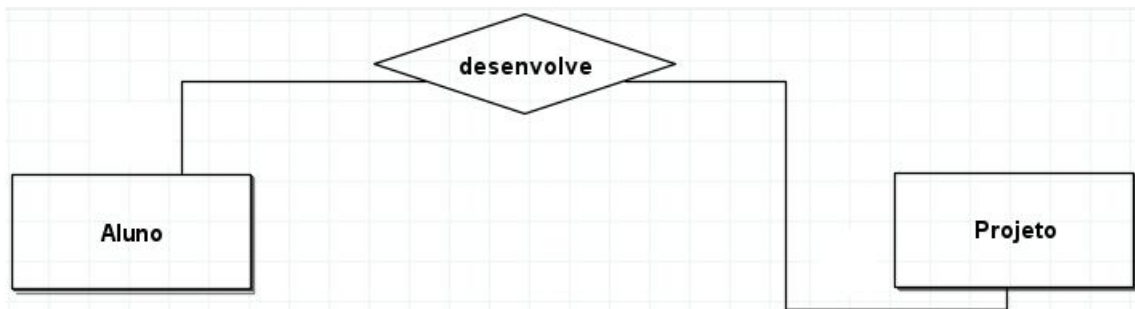


# Entidade

## Exercícios

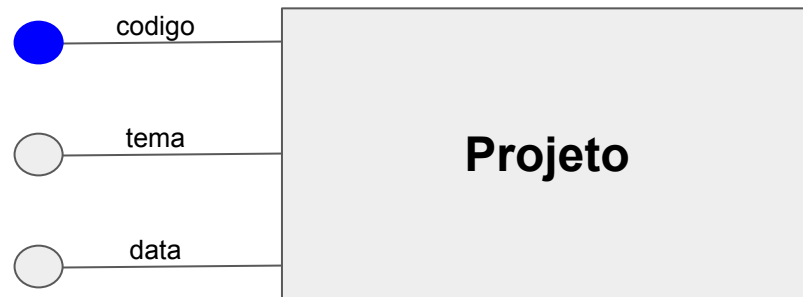
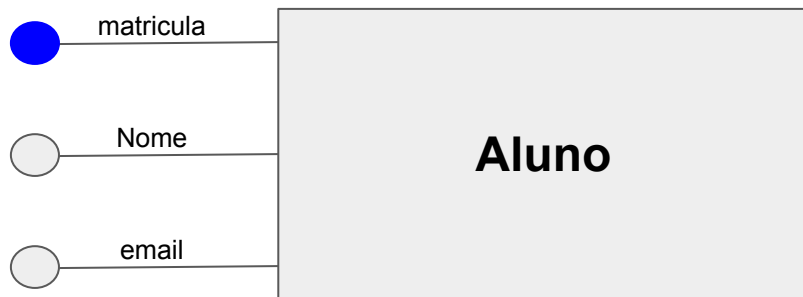
Desenvolva o DER para as seguintes regras de negócio abaixo:

Um aluno desenvolve vários projetos. Um projeto é desenvolvido por um ou mais alunos.



# Atributo

- Os atributos são propriedades que descrevem cada entidade
- São representados por um círculo e ligados a uma entidade
- Toda Entidade deve possuir um atributo identificador chamado de chave primária. Este atributo é um código único.



# Atributo

## Tipos de atributos

- Simples: atômico, indivisível. Ex.: nome
- Composto: pode ser subdividido em outros atributos. Ex.: endereço
- Multivalorado: vários valores para um mesmo registro. Ex.: idioma
- Determinante: define de forma única a instância de uma entidade. Ex.: matrícula
- Derivado: atributo que tem relação com outro. Ex. idade (nascimento)
- Chaves: identifica uma instância da entidade. Ex.: CPF

# Relacionamentos

- Relacionamentos servem para ligar duas entidades mostrando como estas entidades se relacionam.
- É através destes relacionamentos que o SGBDR permite realizar as seguintes buscas:
  - Que médico atendeu que paciente?
  - Qual livro foi emprestado para qual aluno?
  - Qual produto foi enviado para qual cliente?

# Relacionamentos

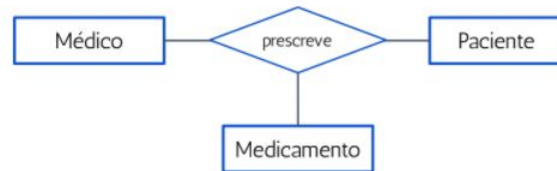
## Grau de um relacionamento

Identificado pelo número de entidades que participam de um relacionamento.

**Unário** - Uma única entidade participante (auto-relacionamento)

**Binário** - Duas entidades participando do relacionamento

**Ternário** - Três entidades participando do mesmo relacionamento



# Cardinalidade

Número de ocorrências que podem estar associadas a um registro da outra entidade.

## Tipos de cardinalidade

**Mínima:** número mínimo de instâncias de entidade que devem participar de um relacionamento

**0 - opcional ou 1 - obrigatório**

**Máxima:** número máximo de instâncias de entidade que podem participar de um relacionamento

**1 ou N - muitos**

## Tipos de relacionamento

- **Relacionamento 1:1 (um para um):** cada uma das duas entidades envolvidas referenciam obrigatoriamente

apenas uma instância da outra.

**Exemplo: Usuário - Currículo**

- **Relacionamento 1:n ou 1:\* (um para muitos):** uma das entidades envolvidas pode referenciar várias instâncias da outra, porém, do outro lado cada uma das várias unidades referenciadas só pode estar ligada uma instância da outra entidade.

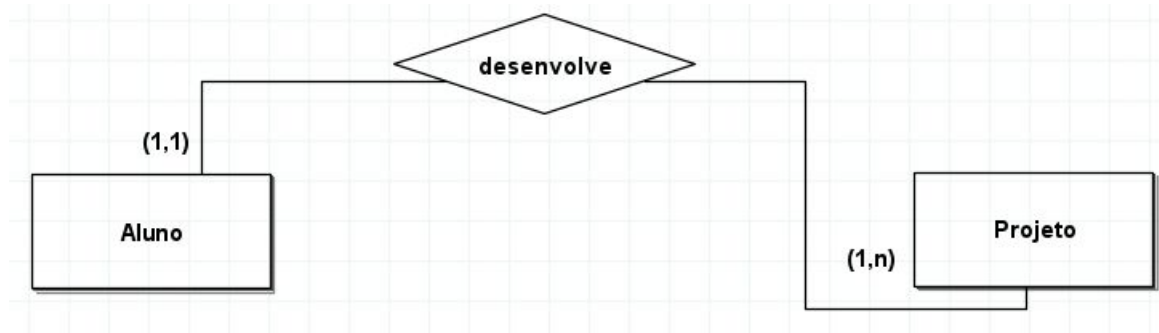
**Exemplo: Usuário - Dependente**

- **Relacionamento n..n ou \*.\* (muitos para muitos):** cada entidade, de ambos os lados, podem referenciar múltiplas instâncias da outra.

**Exemplo: Livro - Autor**

# Relacionamentos

Um aluno desenvolve vários projetos. Um projeto é desenvolvido por um ou mais alunos.





# Prática e Exercícios



### **Montar o DER a seguir:**

- 1) Um diretor gerencia no máximo um departamento.  
Um departamento tem no máximo um diretor.
- 2) Um autor escreve vários livros. Um livro pode ser escrito por vários autores.
- 3) Um autor escreve vários livros. Um livro pode ser escrito por vários autores.