



## UNIDADE III

---

# TECNOLOGIA DE INFORMAÇÃO E COMUNICAÇÃO

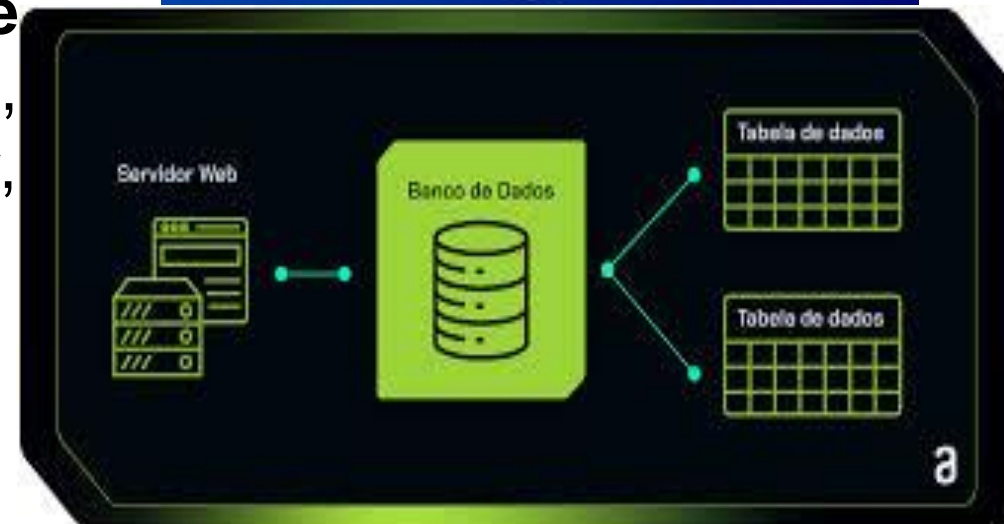
Prof. Me. Filinto Bonte Co

# AULA 03 - BANCO DE DADOS

## Banco de Dados

Um **Banco de Dados** é um conjunto organizado de informações estruturadas para facilitar o armazenamento, recuperação e manipulação de dados.

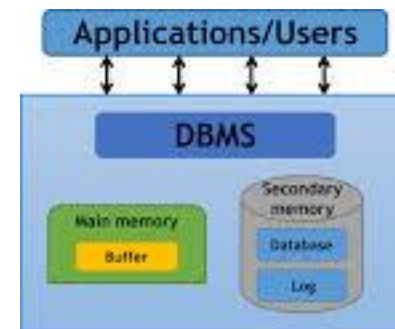
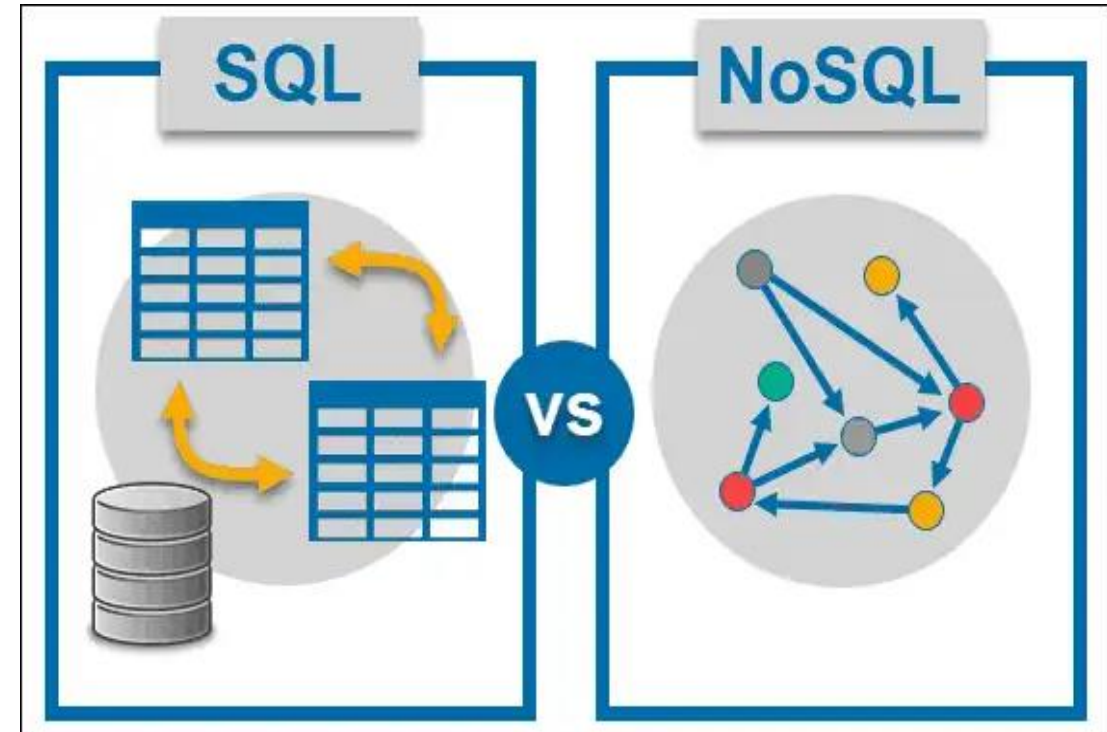
Ele é gerenciado por um **Sistema de Gerenciamento de Banco de Dados (SGBD)**, como: MySQL, PostgreSQL, SQL Server, MongoDB, entre outros.



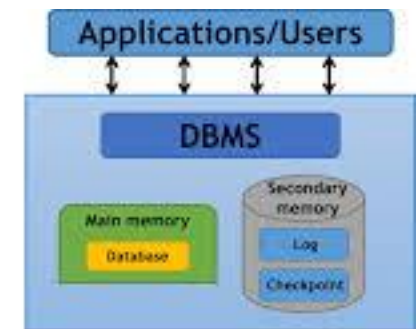
# AULA 03 - BANCO DE DADOS

## Modelos de Dados

1. Modelo Hierárquico
2. Modelo em Rede
3. Modelo Relacional
4. Modelo Orientado a objetos
5. Sistema Objeto-relacionais



(a)

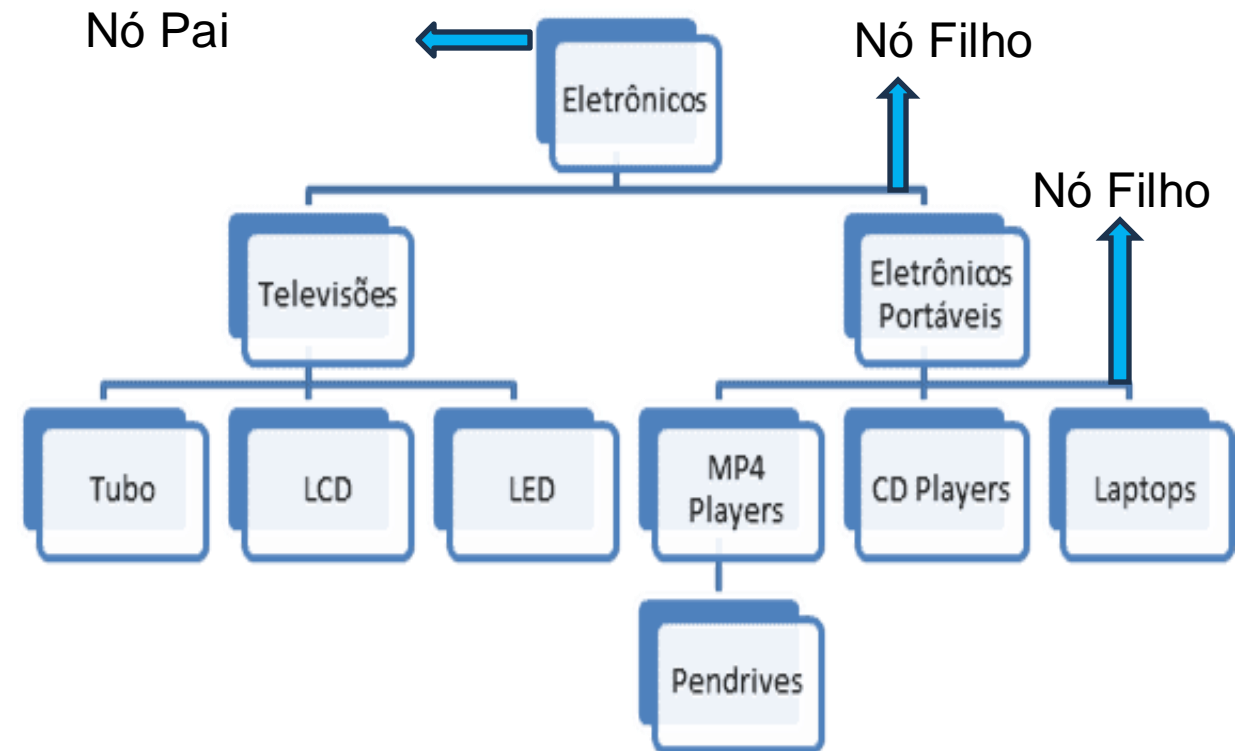


(b)

# AULA 03 - BANCO DE DADOS

## Modelo Hierárquico

- O modelo hierárquico foi o primeiro a ser reconhecido como um modelo de dados.
- Seu desenvolvimento somente foi possível devido à consolidação dos discos de armazenamento endereçáveis.
- Nesse modelo de dados, os dados são estruturados em hierarquias ou árvores.
- Os dados são organizados em uma **estrutura de árvore**, com um **nó raiz (pai)** e vários **nós filhos**.
- Um dos exemplos mais conhecidos desse modelo é o **IBM Information Management System (IMS)**.



# AULA 03 - BANCO DE DADOS

## Modelo Hierárquico

Vantagens:

- ❖ **Acesso rápido aos dados** – A estrutura hierárquica permite localizar informações rapidamente.
- ❖ **Boa integridade referencial** – Como cada nó filho tem apenas um pai, há menos chances de inconsistências nos dados.
- ❖ **Eficiência em grandes volumes de dados** – Ideal para aplicações que lidam com muitos registros estruturados, como sistemas bancários antigos.

# AULA 03 - BANCO DE DADOS

## Modelo Hierárquico

Desvantagens:

- ❖ **Falta de flexibilidade** – Se precisar modificar a estrutura, pode ser difícil reorganizar os dados.
- ❖ **Relacionamentos limitados** – Só permite relações de **um para muitos** (não suporta relações complexas como **muitos para muitos**).
- ❖ **Dificuldade na consulta de dados** – Para acessar informações específicas, é necessário seguir toda a hierarquia.

# AULA 03 - BANCO DE DADOS

## Modelo em Rede

O modelo em redes surgiu como uma extensão ao modelo hierárquico, eliminando o conceito de hierarquia e permitindo que um mesmo registro estivesse envolvido em várias associações.

- ❖ Ele foi desenvolvido na década de 1960 como uma evolução do modelo hierárquico e foi padronizado pelo **CODASYL (Conference on Data Systems Languages)**.
- ❖ Os **dados são organizados em registros** (entidades) conectados por **ligações** (relacionamentos).
- ❖ Diferente do modelo hierárquico, **um registro pode ter vários pais e vários filhos**, permitindo relações **muitos para muitos**.

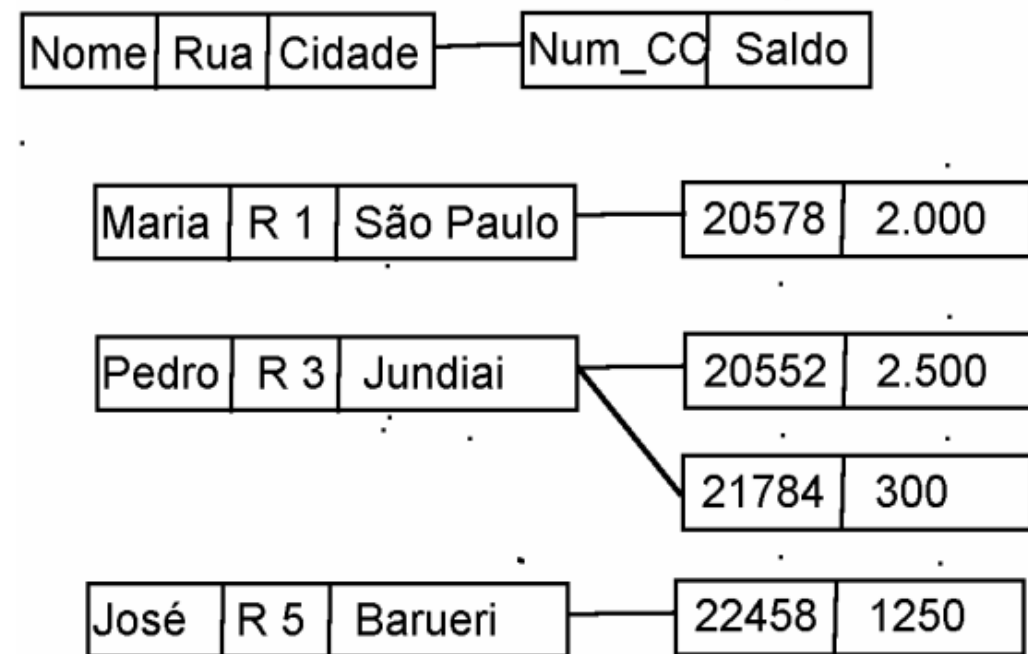
# AULA 03 - BANCO DE DADOS

## Modelo em Rede

O acesso aos dados geralmente é feito por meio de **ponteiros** ou **listas encadeadas**, o que melhora a navegação entre registros.

### Vantagens:

- ❖ **Maior flexibilidade** – Permite relações complexas (muitos para muitos), sendo mais versátil que o modelo hierárquico.
- ❖ **Eficiência no acesso aos dados** – Como os registros são conectados diretamente, as buscas podem ser rápidas.
- ❖ **Maior reutilização dos dados** – Um mesmo registro pode ser acessado por diferentes caminhos.





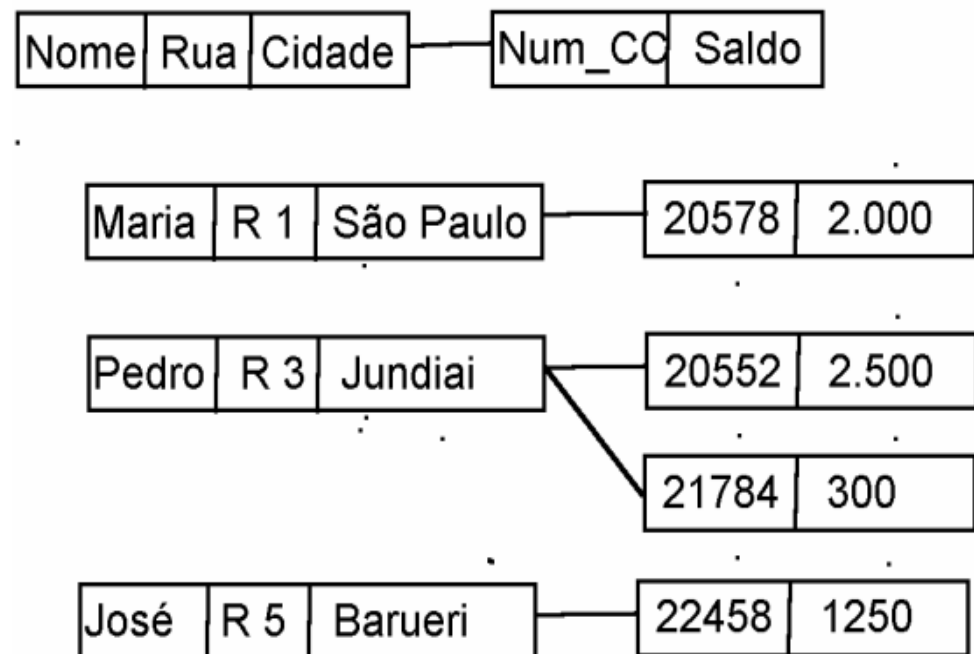
# AULA 03 - BANCO DE DADOS

## Modelo em Rede

O acesso aos dados geralmente é feito por meio de **ponteiros** ou **listas encadeadas**, o que melhora a navegação entre registros.

### Desvantagens:

- ❖ **Complexidade na implementação** – A estrutura exige um gerenciamento cuidadoso dos ponteiros e das relações.
- ❖ **Dificuldade de manutenção** – Adicionar novos tipos de relacionamentos pode ser complicado.
- ❖ **Menor independência dos dados** – A forma como os dados estão organizados pode afetar a forma como são acessados.



# AULA 03 - BANCO DE DADOS

## Modelo Relacional

O modelo relacional apareceu devido às seguintes necessidades:

- ❖ Aumentar a independência de dados nos sistemas gerenciadores de banco de dados;
- ❖ Prover um conjunto de funções apoiadas em álgebra relacional para armazenamento e recuperação de dados; permitir processamento *ad hoc*.

The diagram illustrates a relational table with the following structure and annotations:

	Cód	Mercadoria	Qtde_Estoque	Fornecedor	Validade
	189	Azeitonas Pretas	50	05	02/08/2018
	222	Peixe Congelado	26	08	22/09/2019
	236	Enlatado	48	04	15/11/2018

Annotations:

- Linhas:** Indicated by a bracket on the left side of the table, pointing to the data rows.
- Colunas:** Indicated by a bracket at the top of the table, pointing to the column headers.
- Chave Primária:** An arrow points to the 'Cód' column, indicating it is the primary key.
- Chave Estrangeira:** An arrow points to the 'Fornecedor' column, indicating it is a foreign key.
- Coluna com restrição NOT NULL:** An arrow points to the 'Validade' column, indicating it has a NOT NULL constraint.

# AULA 03 - BANCO DE DADOS

## Modelo Relacional

- ❖ O modelo relacional, tendo por base a teoria dos conjuntos e álgebra relacional, foi resultado de um estudo teórico realizado por CODD.
- ❖ O Modelo relacional revelou-se ser o mais flexível e adequado ao solucionar os vários problemas que se colocam no nível da concepção e implementação da base de dados.
- ❖ Uma relação é constituída por um ou mais atributos (campos) que traduzem o tipo de dados a armazenar.
- ❖ Cada instância do esquema (linha) é chamada de tupla (registro).

# AULA 03 - BANCO DE DADOS

## Modelo Relacional

❖ O **modelo relacional** é uma abordagem de organização de dados baseada em **tabelas** (ou **relações**), onde as informações são armazenadas em linhas (**tuplas**) e colunas (**atributos**). Foi proposto pelo matemático **Edgar F. Codd** em 1970 e é o modelo mais utilizado atualmente em bancos de dados como **MySQL**, **PostgreSQL**, **SQL Server** e **Oracle**.

## Comparando os Modelos

Relacional ou SQL	Document ou NoSQL
Rows (linhas ou registros)	Documentos
Colunas	Propriedades
Schema rígido	Schema-free
Altamente normalizado	Tipicamente denormalizado

Tabela: Funcionário

ID	Nome	Admissão
10	Luis Renato	25/05/2019

Tabela: Férias

Cod	ID	Período	Diretor
201	10	07/2020	Cristiano
202	10	05/2021	Carlos
203	10	09/2022	Carlos

Document

```
{
  "ID": 10,
  "Nome": "Luis Renato",
  "Admissão": "25/05/2019",
  "Férias": [
    {"Período": "07/2020", "Diretor": "Cristiano"},
    {"Período": "05/2021", "Diretor": "Carlos"},
    {"Período": "09/2022", "Diretor": "Carlos"},
  ]
}
```

# AULA 03 - BANCO DE DADOS

## Modelo Relacional

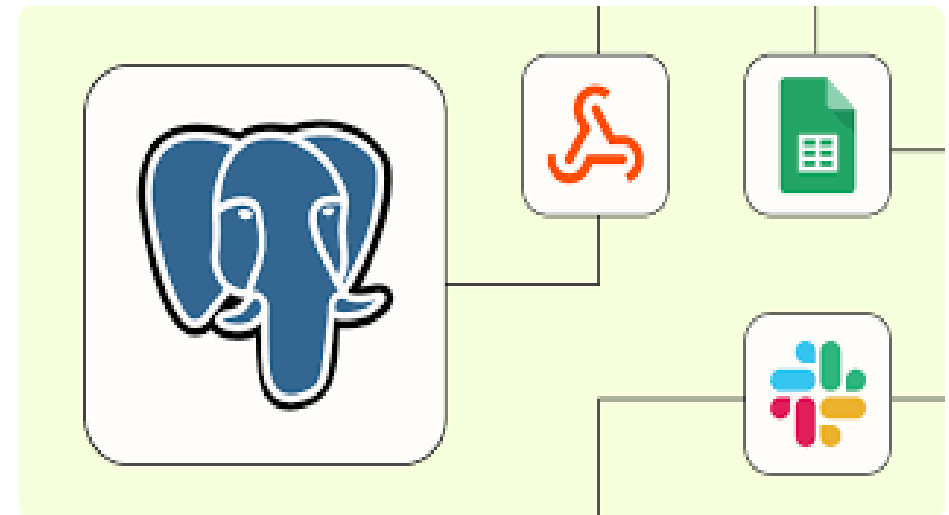
- ❖ O **MySQL** é um dos bancos de dados relacionais mais populares do mundo, conhecido por sua **simplicidade, desempenho e confiabilidade**. Ele é amplamente usado por empresas como Facebook, Twitter e YouTube. Possui suporte para transações ACID, mas por padrão usa o mecanismo de armazenamento **InnoDB**, que é otimizado para leitura e escrita rápida.



# AULA 03 - BANCO DE DADOS

## Modelo Relacional

- ❖ O **PostgreSQL** é um banco de dados relacional avançado, conhecido por seu **suporte a SQL estendido, alta escalabilidade e conformidade com ACID**. Ele é utilizado em aplicações que exigem alta integridade e suporte a transações complexas. Além disso, suporta **tipos de dados personalizados, JSON e XML**.



# AULA 03 - BANCO DE DADOS

## Modelo Relacional

- ❖ O **SQL Server** é um banco de dados relacional da Microsoft, amplamente utilizado em **ambientes corporativos**. Ele oferece **alta integração com produtos Microsoft**, como **Azure, Power BI e .NET**. Sua versão gratuita (SQL Server Express) tem limitações de armazenamento e processamento.



# AULA 03 - BANCO DE DADOS

## Modelo Relacional

- ❖ O **Oracle Database** é um dos bancos de dados mais robustos e confiáveis do mercado, projetado para grandes empresas que precisam de **alta disponibilidade, escalabilidade e segurança**. Ele suporta funcionalidades avançadas como **processamento distribuído, clustering e replicação**.

---

**ORACLE®**  
DATABASE





# AULA 03 - BANCO DE DADOS

## Modelo Relacional

### Vantagens

- ❖ **Simplicidade** – Dados organizados em tabelas são fáceis de entender e gerenciar.
- ❖ **Flexibilidade** – Permite criar diferentes tipos de relacionamentos (**1 para 1, 1 para muitos, muitos para muitos**).
- ❖ **Integridade dos Dados** – Usa restrições como **chaves primárias** e **chaves estrangeiras** para evitar inconsistências.
- ❖ **Facilidade de Consulta** – Usa **SQL (Structured Query Language)** para manipular e recuperar dados eficientemente.

# AULA 03 - BANCO DE DADOS

## Modelo Relacional

### Desvantagens

- ❖ **Pode ser mais lento para grandes volumes de dados** – A normalização pode levar a muitas junções de tabelas, tornando algumas consultas mais lentas.
- ❖ **Estrutura rígida** – Qualquer alteração na estrutura das tabelas pode exigir mudanças significativas.
- ❖ **Não é ideal para dados altamente não estruturados** – Modelos **NoSQL** são mais indicados para esse tipo de aplicação.

# AULA 03 - BANCO DE DADOS

## Modelo Orientado a objetos

Os bancos de dados orientados a objeto começaram a se tornar comercialmente viáveis em meados de 1980.

- ❖ A motivação para seu surgimento está em função dos limites de armazenamento e representação semântica impostas no modelo relacional.
- ❖ Alguns exemplos são os sistemas de informações geográficas (SIG), os sistemas CAD e CAM, que são mais facilmente construídos usando tipos complexos de dados.
- ❖ A habilidade para criar os tipos de dados necessários é uma característica das linguagens da POO.
- ❖ Diferente do **modelo relacional**, que usa tabelas e SQL, o **OODB** armazena os dados como objetos, assim como em linguagens de programação como **Java**, **C++**, **Python**, etc.

# AULA 03 - BANCO DE DADOS

## Modelo de Classes em um Sistema de Banco de Dados OODB

Aqui, o objeto **Cliente** contém um objeto **Endereço** dentro dele, sem precisar de tabelas separadas como no modelo relacional.

java

```
class Cliente {
    String nome;
    int idade;
    Endereco endereco; // Associação com outro objeto

    void comprarProduto() {
        System.out.println(nome + " realizou uma compra.");
    }
}

class Endereco {
    String rua;
    String cidade;
    String estado;
}
```

# AULA 03 - BANCO DE DADOS

## Modelo de Classes em um Sistema de Banco de Dados OODB

Se armazenarmos um cliente no banco de dados orientado a objetos, ele ficaria assim:

**Diferente do modelo relacional**, onde teríamos tabelas separadas para clientes e endereços, aqui os dados são armazenados juntos em um único objeto.



```
Cliente {  
    nome: "João",  
    idade: 30,  
    endereco: {  
        rua: "Av. Brasil, 123",  
        cidade: "São Paulo",  
        estado: "SP"  
    }  
}
```

## AULA 03 - BANCO DE DADOS

### Sistema Objeto-Relacional (ORDBMS - Object-Relational Database Management System)

A área de atuação dos sistemas Objeto-Relacional tenta suprir a dificuldade dos sistemas relacionais convencionais, que é o de representar e manipular dados complexos, visando ser mais representativos em semântica e construções de modelagens.

- ❖ A solução proposta é a adição de facilidades para manusear tais dados utilizando-se das facilidades SQL (Structured Query Language) existentes.
- ❖ Permite o uso de **herança, encapsulamento e polimorfismo**, mantendo compatibilidade com o SQL.

# AULA 03 - BANCO DE DADOS

## Sobre Trabalho (NP1)

Análise de um caso de uso real de tecnologia da informação aplicada ao cotidiano, destacando desafios e soluções.

- 1. Aplicação de sistemas de telemedicina em comunidades remotas: impactos e superação de barreiras tecnológicas**
- 2. Uso de plataformas de educação online para inclusão digital em regiões com baixa conectividade.**
- 3. Inteligência Artificial no gerenciamento do trânsito urbano: casos práticos e adaptações locais.**
- 4. Sensores IoT na agricultura familiar: modernização sustentável e desafios de implementação.**

# AULA 03 - BANCO DE DADOS

Tipo de Relacionamento	Exemplo	Implementação
<b>1 para 1 (1:1)</b>	Pessoa ↔ CPF	Uma chave estrangeira
<b>1 para Muitos (1:N)</b>	Cliente ↔ Pedidos	Chave estrangeira na tabela "muitos"
<b>Muitos para Muitos (M:N)</b>	Alunos ↔ Disciplinas	Criar uma tabela intermediária



**ATÉ A PRÓXIMA!**