



Banco de Dados I  
Eng. Ms. Helio Augusto de Lima Rangel

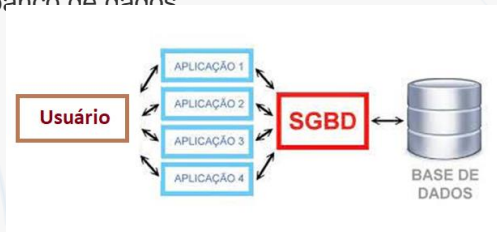
## Porque Banco de Dados?

- Como armazenávamos dados antes do computador?
- Dados armazenados em formato texto;
  - Texto livre;
  - Espaçamento fixo (ISAM - *Indexed Sequential Access Method*);
  - Delimitado por separadores; (arquivos formato .csv do Excel por exemplo)
  - XML etc.
- O modelo hierárquico. (Estrutura de dados: Árvores);
- O modelo em Redes. (Estrutura de dados: Listas duplamente encadeadas);
- **O modelo Relacional;**
  - Os índices
- O modelo Orientado a Objetos;
  - As classes e os objetos
- O modelo não relacional;
  - NO SQL
- O problema de armazenar informações de forma digital;
- O problema de armazenar, encontrar e recuperar dados armazenados;
- Exemplos de algumas aplicações dos Banco de Dados.

## Requisitos de um Banco de Dados

O que é um Sistema Gerenciador De Bancos De Dados **Relacional?** (SGBDR)

É uma coleção de itens de dados com relacionamentos predefinidos entre si. Esses itens são organizados como um conjunto de tabelas com colunas e linhas. As tabelas são usadas para reter informações sobre os objetos a serem representados no banco de dados. Cada coluna da tabela retém um determinado tipo de dado e um campo armazena o valor em si de um atributo. As linhas na tabela representam uma coleção de valores relacionados de um objeto ou de uma entidade. Cada linha em uma tabela pode ser marcada com um único identificador chamado de chave principal. Já as linhas entre as várias tabelas podem ser associadas usando chaves estrangeiras. Esses dados podem ser acessados de várias maneiras diferentes sem reorganizar as próprias tabelas do banco de dados.



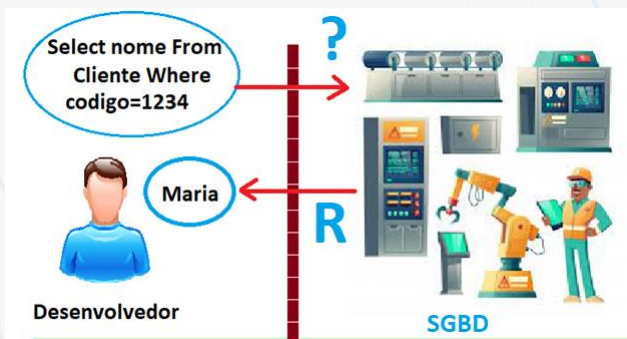
## O que devemos esperar de um bom SGBDR

- Mecanismos eficazes de segurança;
- Bom desempenho;
- Bom custo benefício;
- Controle de Redundância;
- Compartilhamento de Dados;
- Controle de Acesso aos Dados;
- Representação de associações complexas;
- Garantia de restrições de Integridade;
- Recuperação de falhas;
- Tratamento de transações;
- Permitir o uso de *stored procedures* / *triggers* / *views* etc.



## Independência dos Dados

- Permite que os usuários tenham uma **visão abstrata dos dados**, encapsulando detalhes complexos e não relevantes no momento. Desta forma o desenvolvedor não precisa conhecer como os dados estão fisicamente armazenados para que possa trabalhar com eles.



## Alguns Bancos de Dados de Mercado

