

# Bancos de Datos Distribuidos

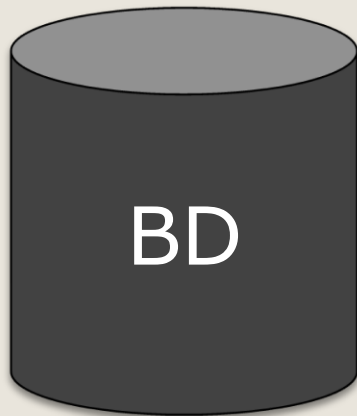


# O que é Sistema Gerenciador de Banco de Dados?



- Um sistema gerenciador de banco de dados (SGBD) é usado como sendo um sistema que permite o gerenciamento dos sistemas de bancos de dados.
- Faz a distribuição dos dados de uma forma transparente para o usuário
- Exemplos: Oracle, SQL Server, MySQL, PostgreSQL, Firebird

# O que é Sistema de Banco de Dados?



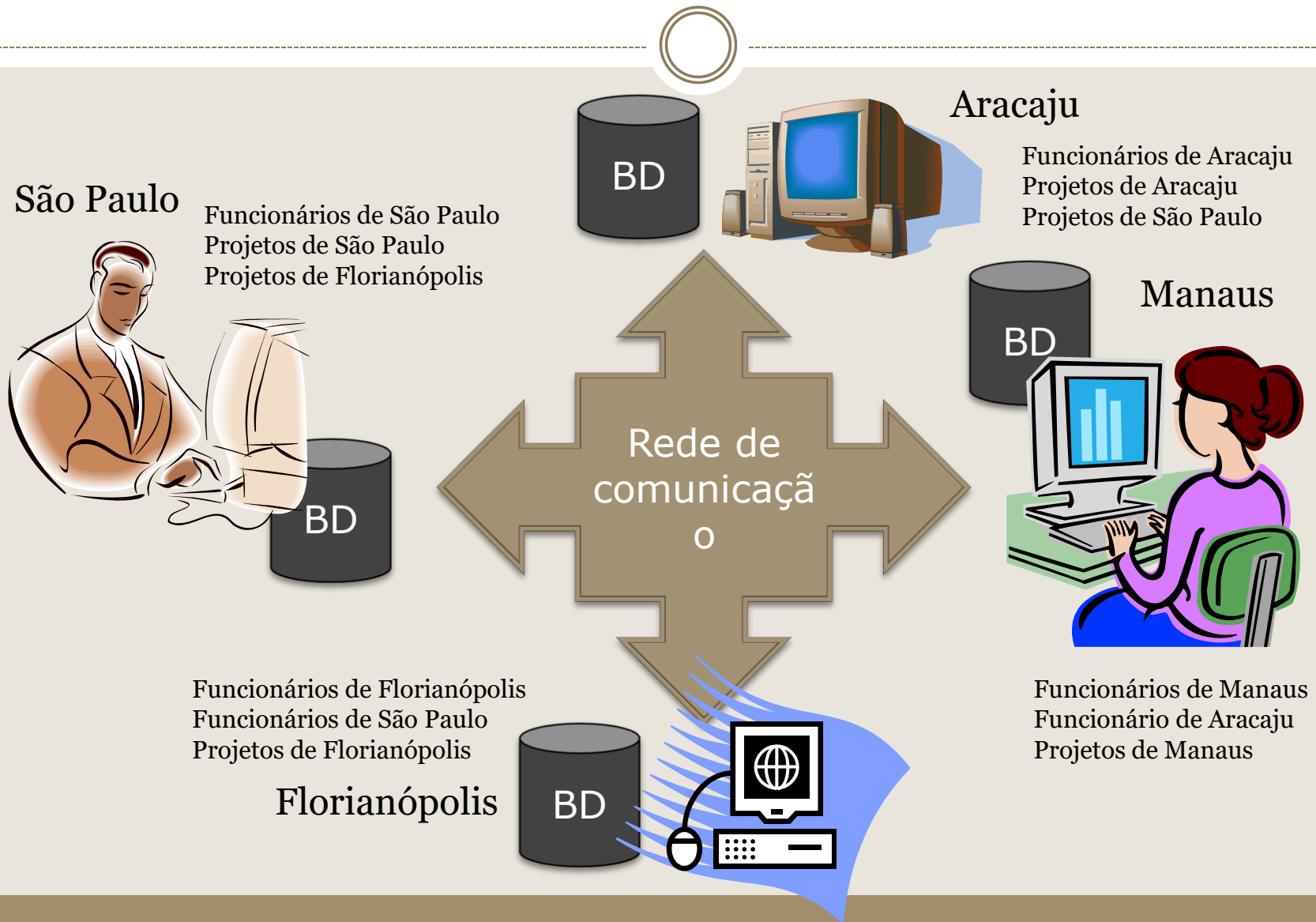
+ SGBD

# O que são Bancos de Dados Distribuídos?



- Segundo (Ozsu e Valduriez 1999), um banco de dados distribuído (BDD) é formado por uma coleção de múltiplos bancos de dados logicamente interrelacionados sobre uma rede de computadores.

# Exemplo de Bancos de Dados Distribuídos



# Quais os objetivos?



- Prover independência de dados num ambiente distribuído
- Transparência para o usuário que “verá” um único banco de dados como se não estivesse distribuído fisicamente
- Disponibilidade

# Vantagens



- Reflete a estrutura organizacional
- Autonomia local: departamento pode controlar seus dados
- Melhor performance: dados mais próximos do local de maior demanda
- Economia de infraestrutura: menor custo de hardware para muitos computadores de médio desempenho do que para um computador de alto desempenho
- Modularidade: sistemas podem ser modificados, adicionados ou removidos sem alterar os outros módulos

# Desvantagens



- Maior custo para desenvolvimento de software
- Maior possibilidade de bugs
- Overhead (aumento de armazenamento e processamento)
- Complexidade: trabalho extra para manter sistemas múltiplos diferentes, ao invés de um único grande
- Segurança: locais remotos devem ser seguros (conexão segura entre os lugares remotos)
- Difícil de manter a integridade: pode ser muito caro reforçar a integridade ao longo de uma rede.



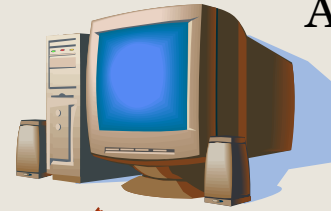
# Como o usuário vê?



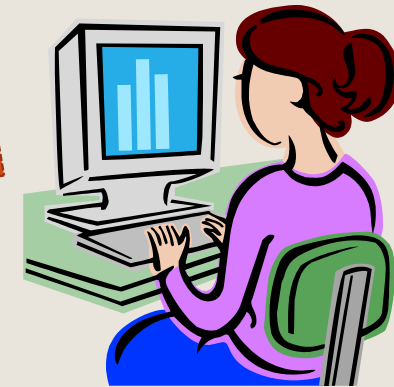
São Paulo



Aracaju



Manaus



Banco de Dados Distribuído

Florianópolis



# BDD Homogêneos ou Heterogêneos?



- **Homogêneos**

- Bancos de dados iguais em todos os nós
- Utilizados quando um projeto inicia seu desenvolvimento, a partir do ponto zero
- Abordagem top-down (de cima para baixo)

- **Heterogêneos**

- Bancos de dados diferentes nos nós
- Utilizados quando há alteração ou migração de um projeto já desenvolvido
- Abordagem bottom-up (de baixo para cima)

# Homogêneos



- Como em um SGBDD homogêneo todos os SGBDs locais oferecem interfaces idênticas, estes últimos usam, então, o mesmo modelo de dados, a mesma DDL e as mesmas DMLs.
- Logo, uma vez fixadas as interfaces locais, é natural que o SGBD global também ofereça estas mesmas interfaces. Assim, qualquer usuário, local ou global, poderá acessar tanto dados locais quanto dados remotos através da mesma DML.

# Heterogêneos



- Em sistemas heterogêneos, os SGBDs locais potencialmente usam modelos de dados e DMLs diferentes. Uma opção seria o SGBD global oferecer ao usuário global, residente em um dado nó, uma visão do banco de dados distribuído no mesmo modelo de dados que o banco local, e permitir que este usuário acesse dados definidos nesta visão através da própria DML local.
- Esta opção é interessante pois não é necessário ensinar uma nova DML aos usuários residentes em um determinado nó para que possam acessar dados remotos. Nesta opção, o SGBD global possui, na verdade, uma interface diferente para cada nó.

# Distribuição dos Dados



- **Replicação**

- Existe uma cópia de cada um dos dados em cada nó da rede, tornando as bases de dados iguais
- **Exemplo:** tabela de produtos de uma grande loja
- **Vantagem:** disponibilidade
- **Desvantagem:** overhead (aumento de armazenamento e processamento)

- **Fragmentação**

- Os dados são divididos em todo o sistema, portanto, há uma base de dados diferente em cada nó, se observarmos de forma local, porém, de forma global, é apenas uma base de dados

# Replicação



- **Síncrona**
  - Cada transação é concluída quando todos os nós confirmam que a transação local foi bem sucedida
- **Assíncrona**
  - O nó principal executa a transação enviando confirmação ao solicitante e então encaminha a transação aos demais nós

# Fragmentação



- Vertical

PROJ<sub>1</sub>: information about project budgets

PROJ<sub>2</sub>: information about project names and locations

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

PROJ<sub>1</sub>

PNO	BUDGET
P1	150000
P2	135000
P3	250000
P4	310000
P5	500000

PROJ<sub>2</sub>

PNO	PNAME	LOC
P1	Instrumentation	Montreal
P2	Database Develop.	New York
P3	CAD/CAM	New York
P4	Maintenance	Paris
P5	CAD/CAM	Boston

# Fragmentação



- Horizontal

$PROJ_1$  : projects with budgets less than \$200,000

$PROJ_2$  : projects with budgets greater than or equal to \$200,000

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

$PROJ_1$

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York

$PROJ_2$

PNO	PNAME	BUDGET	LOC
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston



# Desenvolvendo projetos



- Levantamento de Informações
- Análise de viabilidade
- Análise de Requisitos
- Projeto
- Implementação (desenvolvimento)
- Testes
- Implantação
- Manutenção

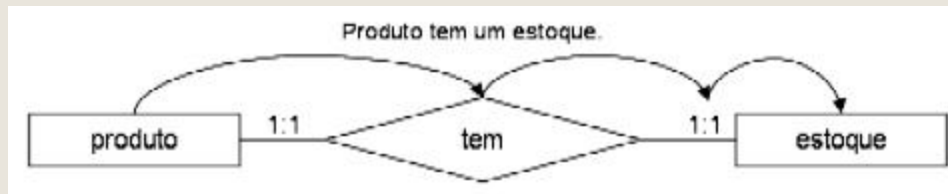
# Desenvolvendo projeto de banco de dados



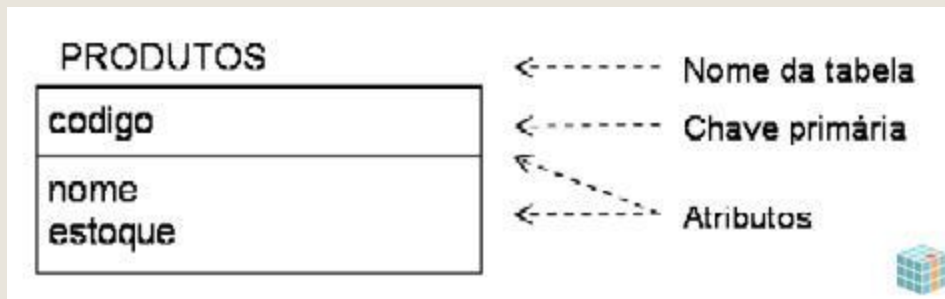
## Relacionamento 1:1

Um produto tem estoque.

*Modelo conceitual:*



*Modelo lógico:*



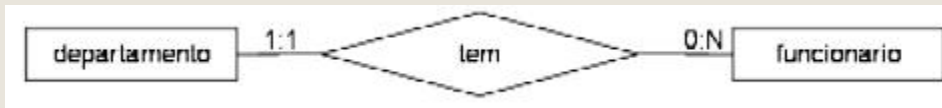
# Desenvolvendo projeto de banco de dados



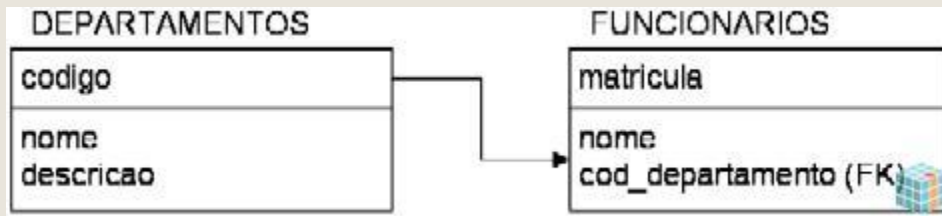
## Relacionamento 1:N

Um departamento tem nenhum ou vários funcionários, mas um funcionário pode pertencer a somente um departamento.

*Modelo conceitual:*



*Modelo lógico:*



# Desenvolvendo projeto de banco de dados



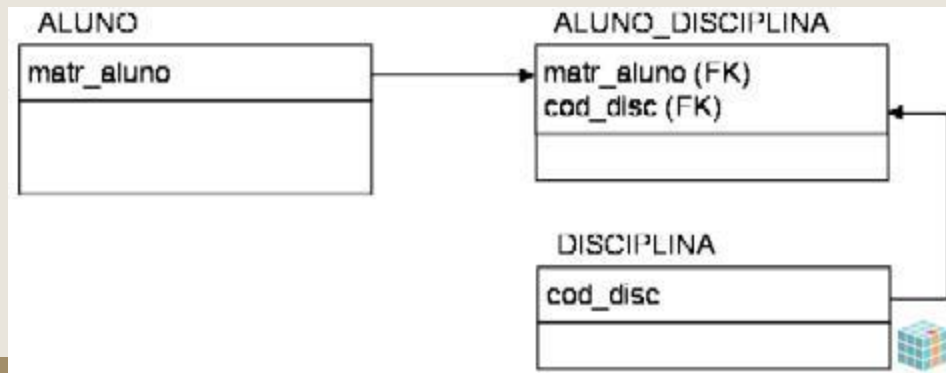
## Relacionamento N:N

Um aluno tem aulas de nenhuma ou várias disciplinas e uma disciplina é cursada por nenhum ou vários alunos.

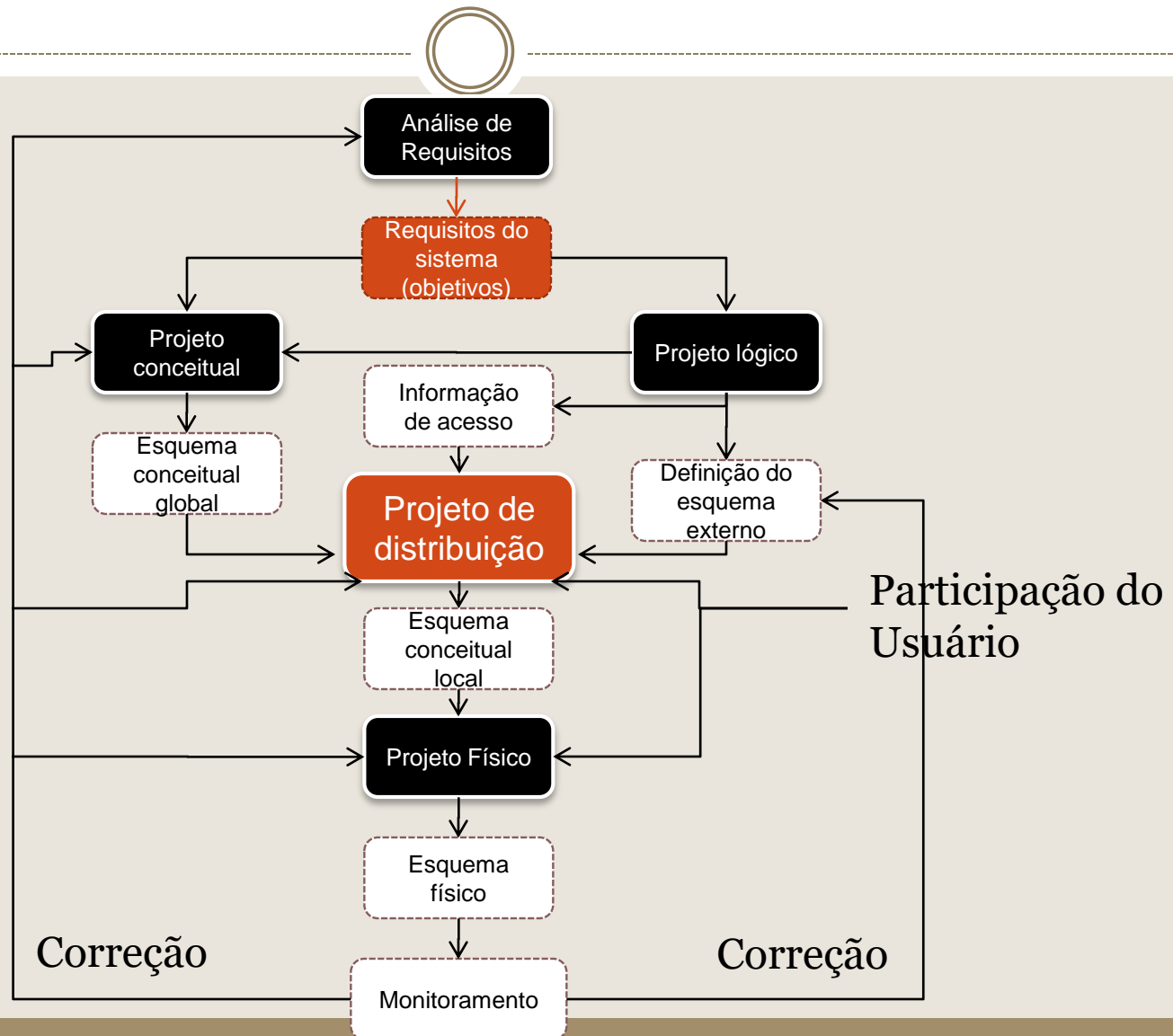
*Modelo conceitual:*



*Modelo lógico:*



# Abordagem top-down



# Compreendendo a abordagem top-down



- Análise de requisitos, projeto conceitual, projeto lógico e projeto físico são etapas tradicionais da abordagem top-down
- Projeto de distribuição é específico de banco de dados distribuído
- Observe que o projeto físico é implementado depois do projeto de distribuição

# Análise de requisitos



- Coleta de informações sobre os dados, suas restrições e seus relacionamentos na organização
- Realizada por meio de reuniões com os usuários onde se observa o funcionamento da organização
- Ao final, é gerado um documento contendo os requisitos de especificação

# Projeto Conceitual



- Modelagem de dados e de seus relacionamentos, independente da estrutura de representação do sistema de banco de dados distribuídos (modelagem conceitual).
- Realizado por meio da análise dos requisitos de especificação
- Ao final, é gerado um diagrama (esquema) conceitual com as devidas restrições de integridade dos dados



# Projeto Lógico



- Conversão do esquema conceitual para o esquema de um Sistema de Banco de Dados Distribuídos (esquema lógico)
- Realizado por meio da aplicação de regras de conversão, tradução para modelo relacional do banco de dados distribuído
- Ao final, é gerado um esquema lógico com tabelas, stored procedures, views, autorizações de acesso, etc

# Projeto de Distribuição



- Definição de como os dados e programas deverão ser alocados, fragmentados nos diversos pontos da rede de computadores
- Em alguns casos, a própria rede de computadores é projetada e construída para atender as necessidades do projeto do banco de dados distribuído
- Esta etapa é a mais crítica no projeto de um banco de dados distribuído

# Projeto Físico



- Definição do esquema lógico em um SBDD adequado ao modelo de dados
- Realizado através de instruções SQL
- O resultado é um esquema físico local em concordância com o estabelecido no projeto de distribuição dos dados
- Ao concluir o projeto físico de cada ponto de rede, o banco de dados distribuído estará pronto para ser usado
- Inicia-se o processo de monitoramento para descoberta de possíveis erros, que serão enviados aos responsáveis pelo projeto do banco de dados distribuído.

# Resumindo



- É muito importante avaliar o ambiente organizacional da empresa detentora dos dados antes da implementação
- É necessário conhecer a função de cada ponto da rede, para melhor distribuição dos dados e criação dos esquemas locais
- Os objetos (views, stored procedures, etc) são criados de acordo com cada esquema físico local

# Como os usuários acessam BDD?



- **Aplicações locais**
  - Não requerem dados de outros lugares
- **Aplicações globais**
  - Requerem dados de outros lugares