

# Introdução aos Bancos de dados

**Renato Rocha Souza**

**[rsouza.fgv@gmail.com](mailto:rsouza.fgv@gmail.com)**

## Programa:

### **Bancos de dados**

- Introdução ao Ambiente Cliente/Servidor
- Introdução aos Bancos de Dados
  - Bancos de Dados: Conceitos
  - SGBD: Sistemas Gerenciadores de Bancos de Dados
  - Arquitetura de Implementação
  - Hierárquico, Rede, Relacional
  - Integridade de Dados
  - Álgebra Relacional / Cálculo Relacional
  - Linguagem SQL
  - Projeto de Bancos de Dados
  - Normalização
  - Modelo de Entidade e Relacionamento
  - Classificação de Modelos
  - Proteção de Dados
  - Transações
  - Concorrência
  - Segurança

# Introdução ao Ambiente Cliente/Servidor

Neste tópico iremos abordar o ambiente cliente/servidor, caracterizando-o e definindo seus principais aspectos.

Serão definidos os conceitos fundamentais da arquitetura, bem como a nomenclatura e terminologia utilizadas.

# Introdução ao Ambiente Cliente/Servidor

- Arquitetura Cliente/Servidor: Definição
- Servidores
  - Rede
  - Arquivos
  - Impressoras e outros recursos compartilhados
  - Bancos de Dados
- Aplicações em rede

# Introdução ao Ambiente Cliente/Servidor

- Aplicações em ambiente Cliente/Servidor
  - Aplicações em duas camadas
  - Aplicações em três camadas
- Servidores de Processo
- Bancos de Dados Cliente/Servidor
  - API's Nativas
  - ODBC
  - OLEDB

# Ambiente Cliente/Servidor

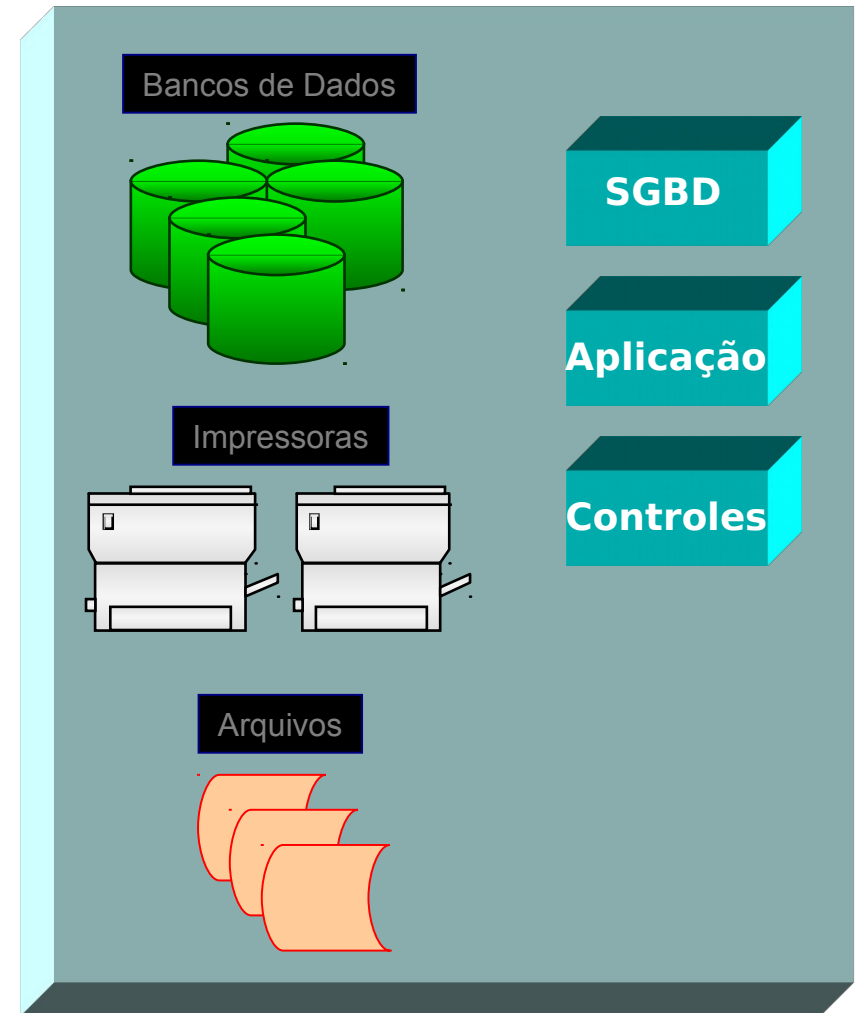
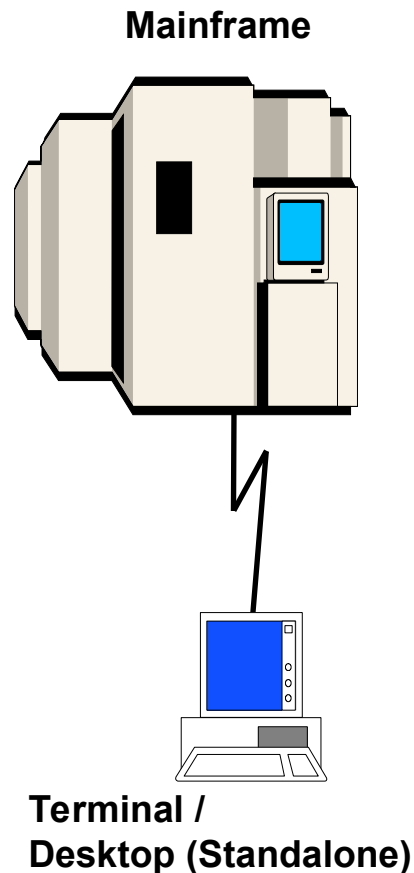
“Computação Cliente/Servidor é o particionamento de uma aplicação em tarefas que são executadas em computadores separados, onde pelo menos um deles é uma estação de trabalho programável, por exemplo, um PC.”

Gartner Group.

# Ambiente Cliente/Servidor

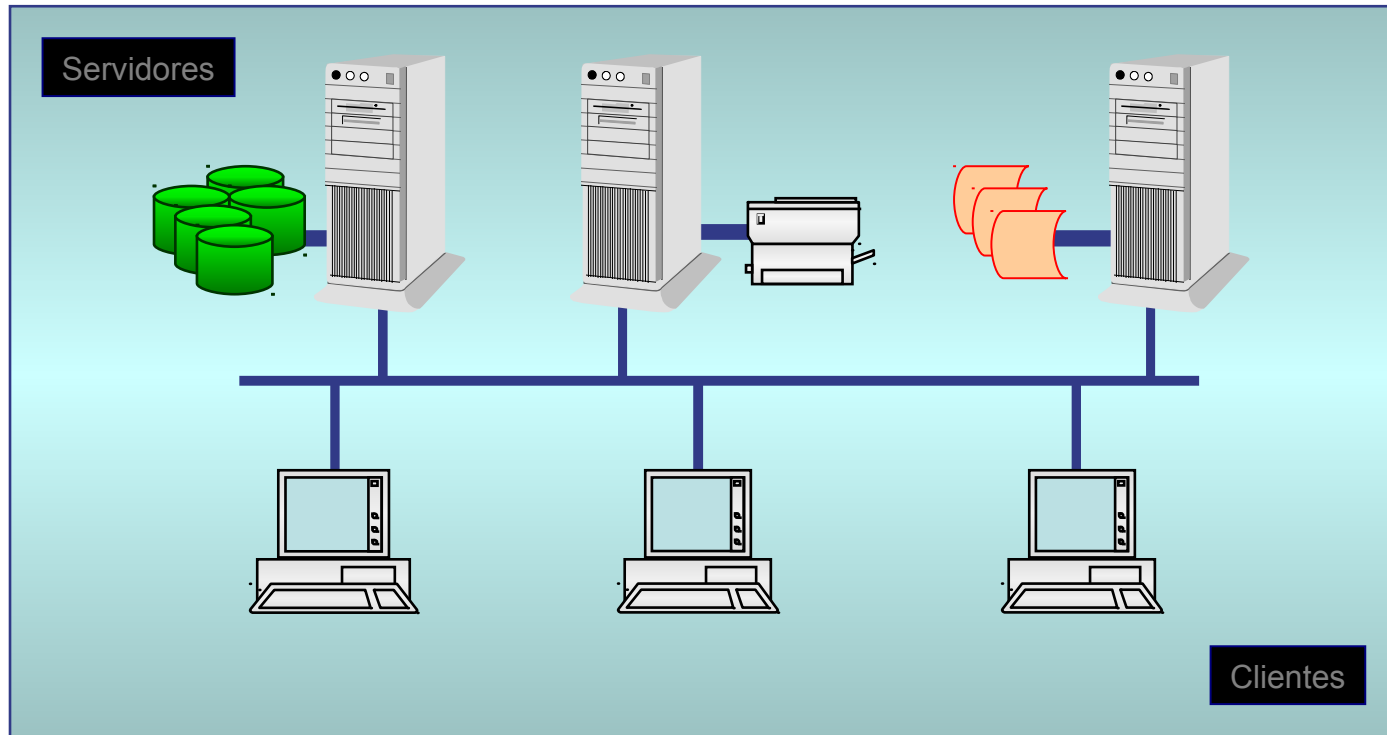


# Introdução ao Ambiente Cliente/Servidor





# Ambiente Cliente/Servidor



# Ambiente Cliente/Servidor

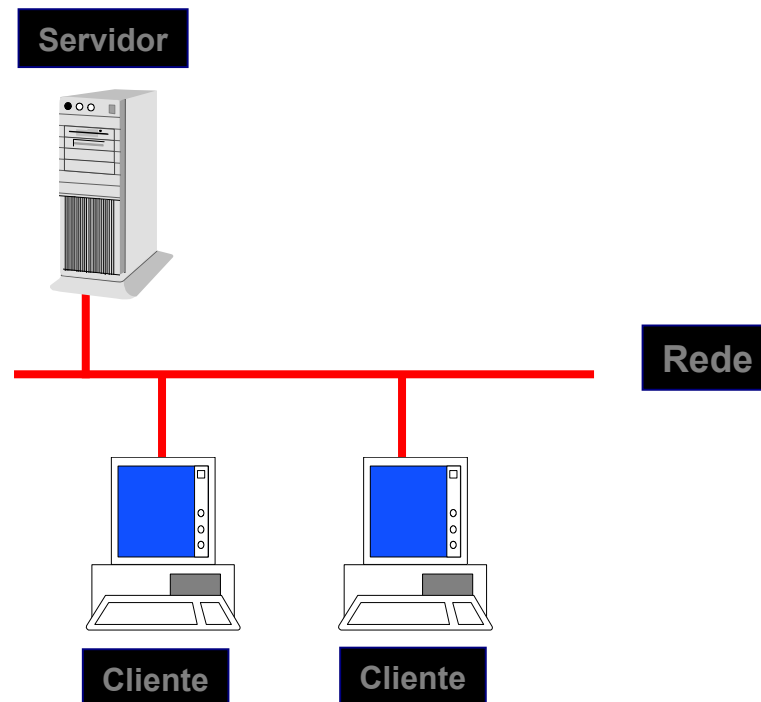
## Vantagens:

- Especialização
- Custo
- Escalabilidade

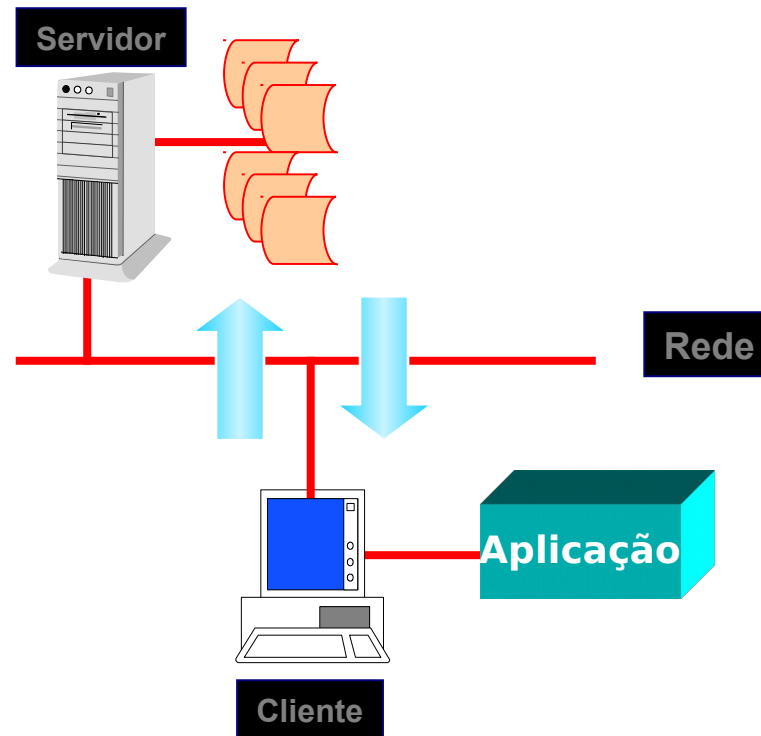
## Desvantagens:

- Falta de Padronização
- Maior necessidade de Gerenciamento
- Complexidade no Desenvolvimento de Aplicações

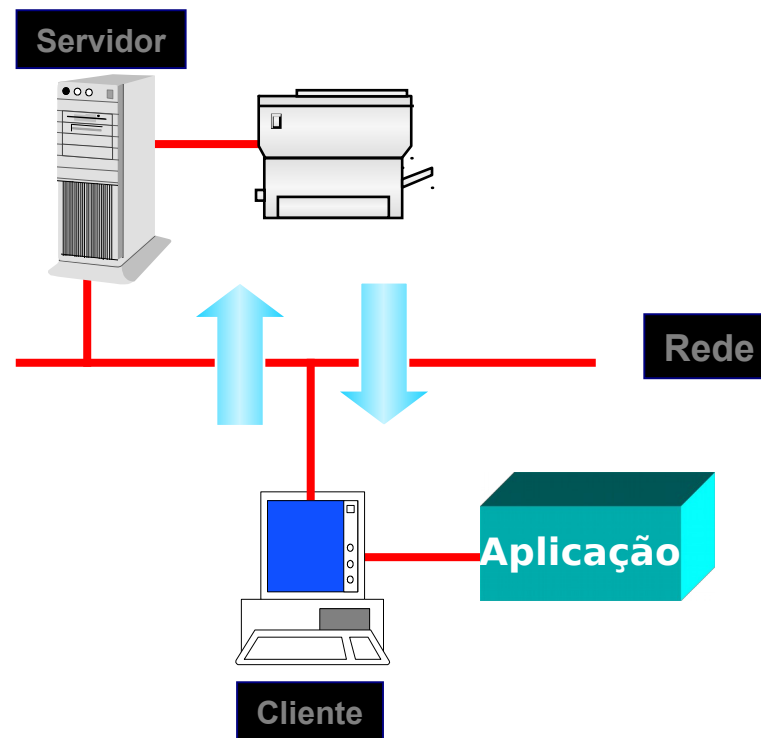
# Servidores de Rede



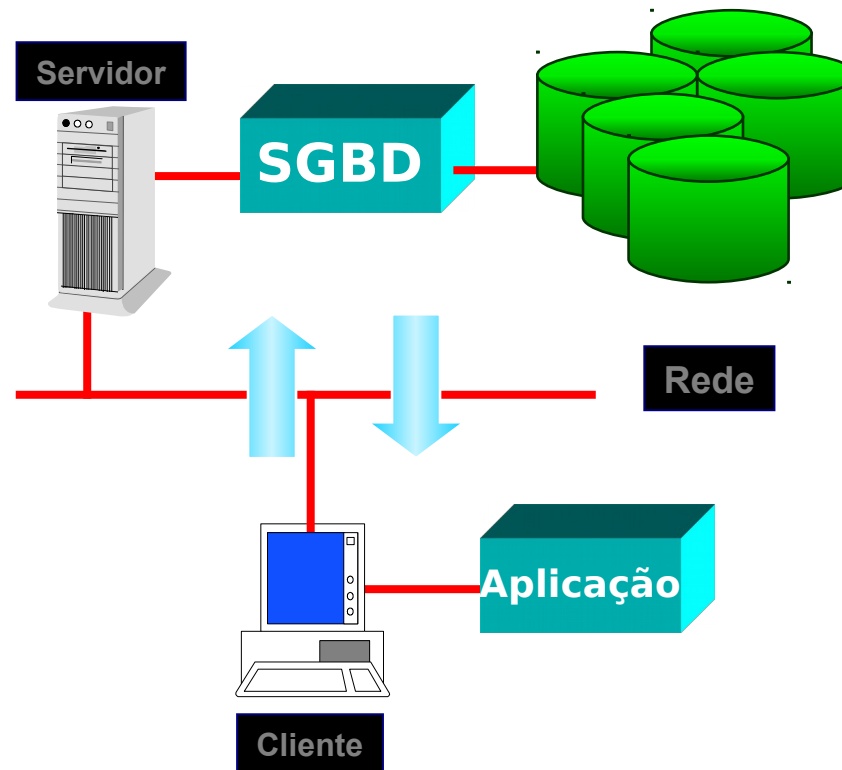
# Servidores de Arquivos



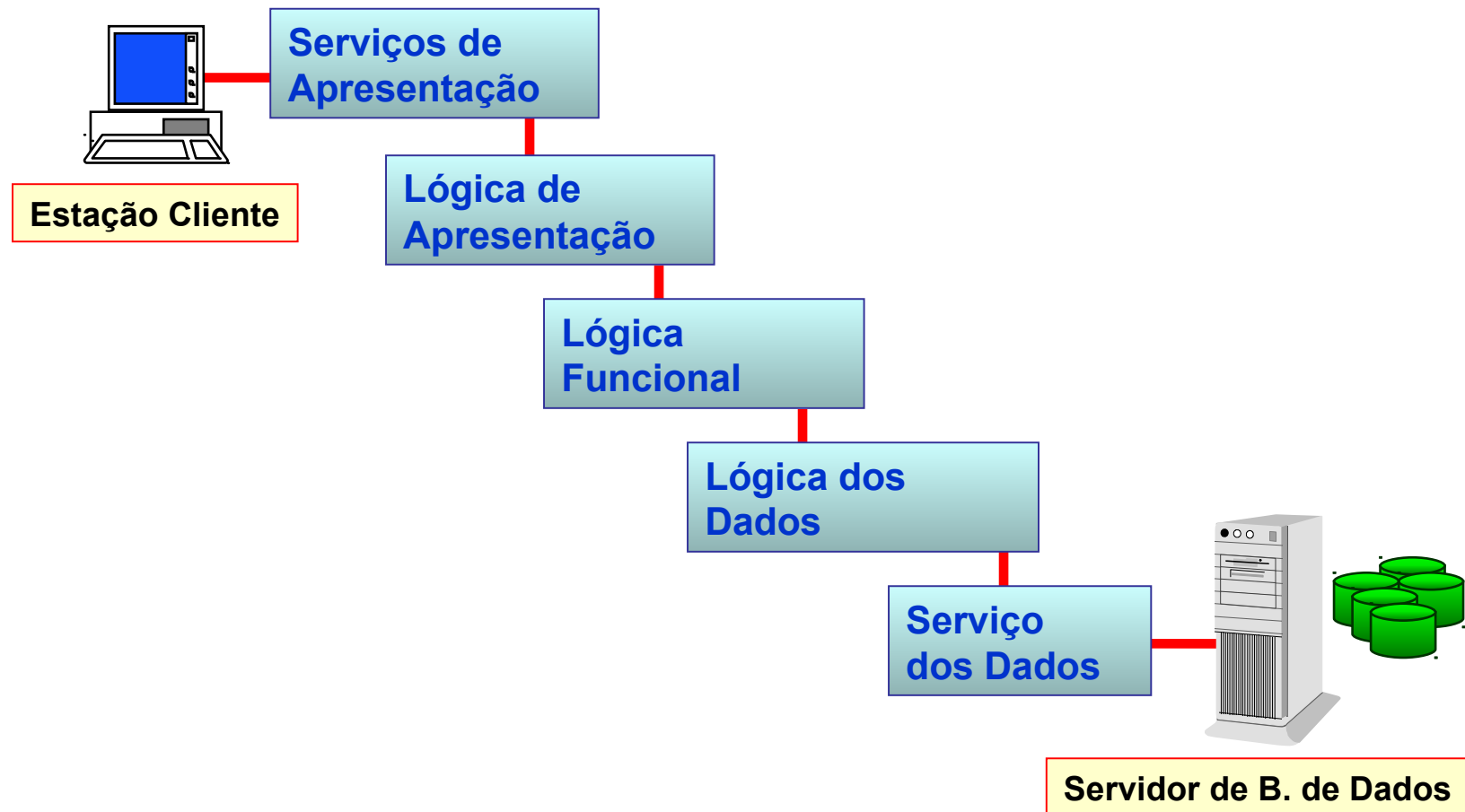
# Servidores de Impressora e Outros Recursos Compartilhados



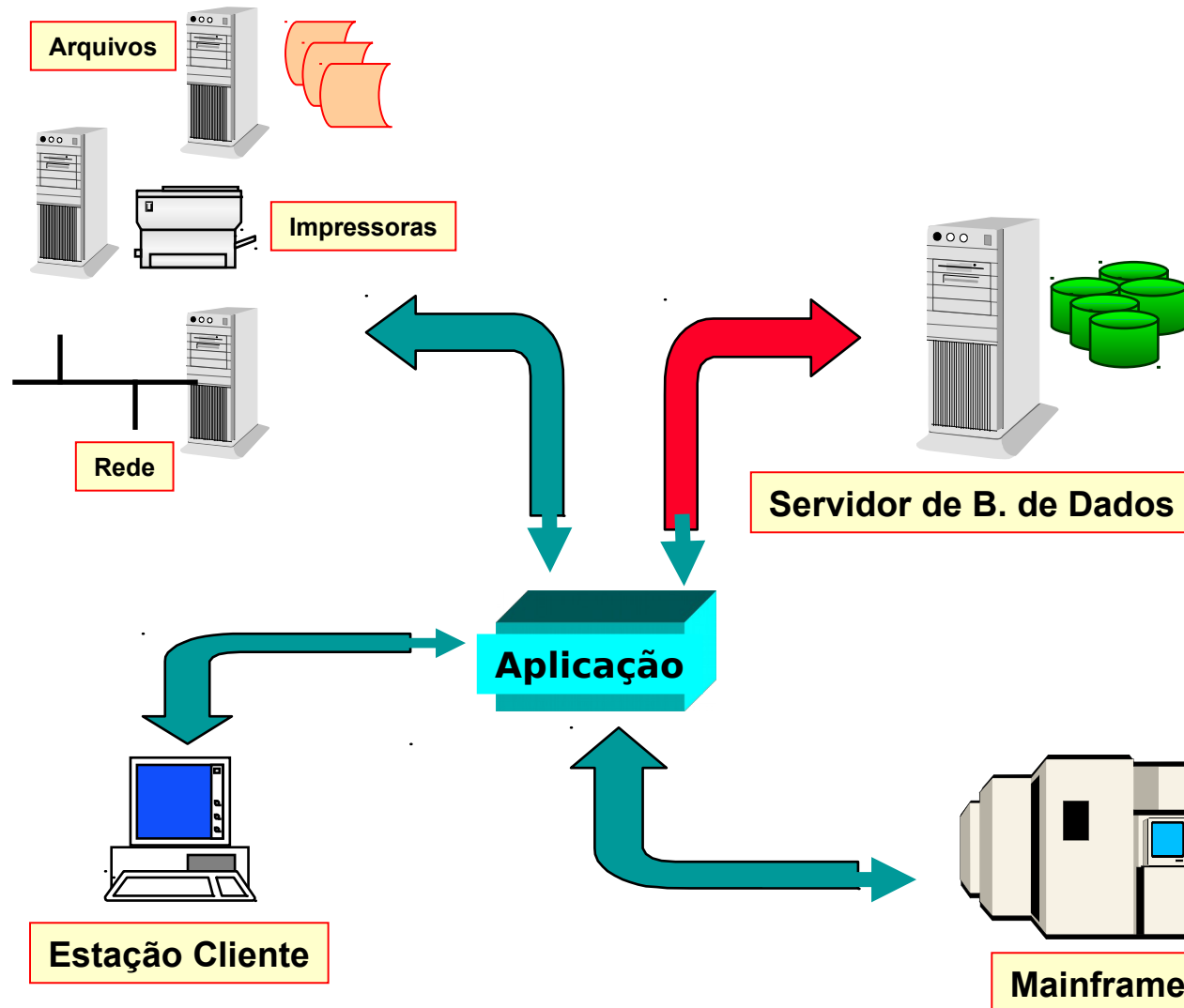
# Servidores de Bancos de Dados



# Servidores de Bancos de Dados



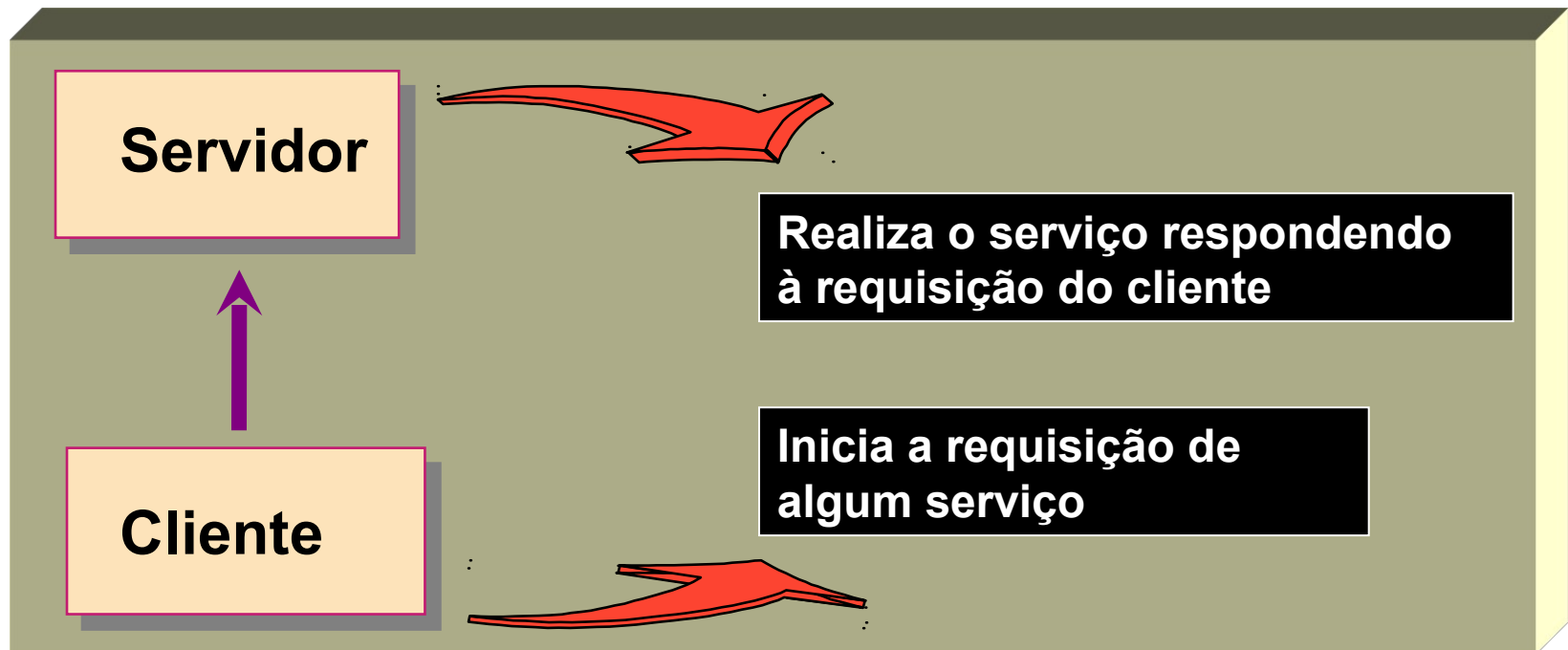
# Aplicações em Rede





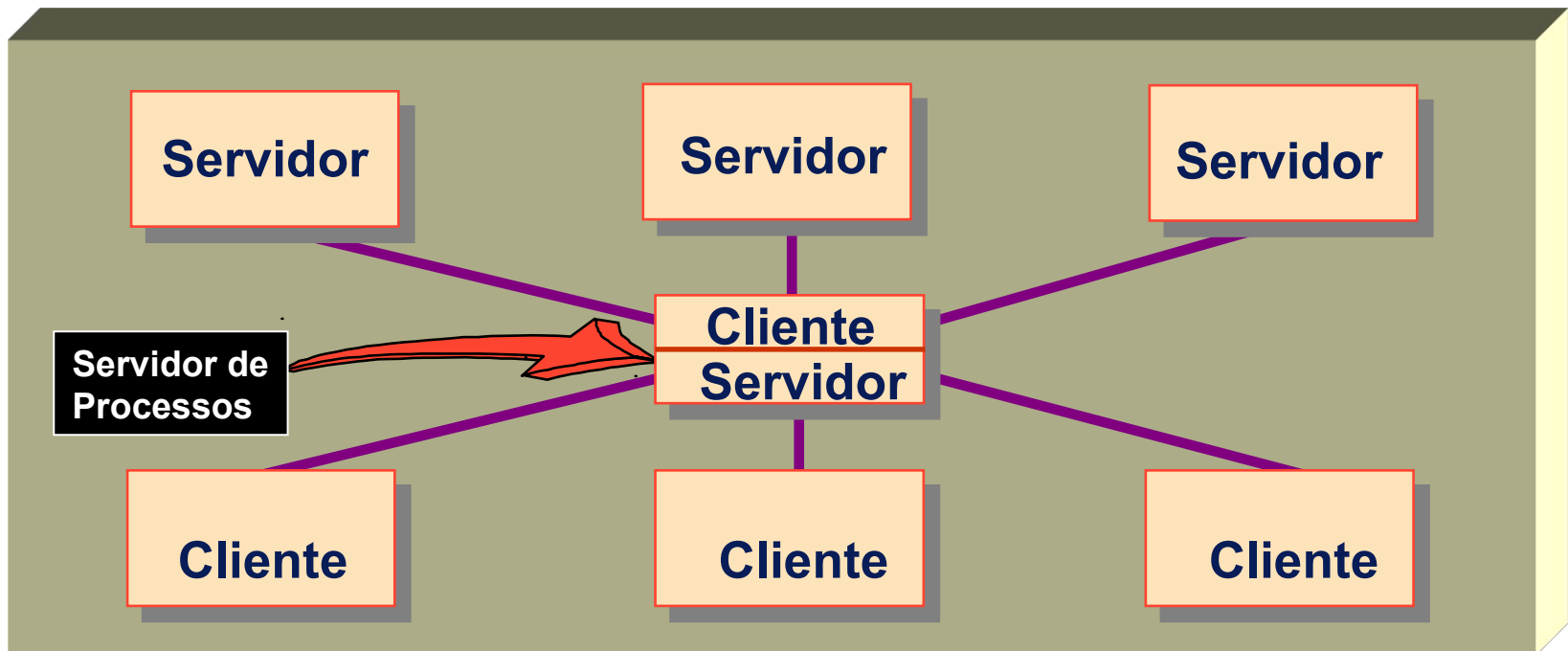
# Aplicações em Ambiente Cliente/Servidor

## Arquitetura Two-Tier



# Aplicações em Ambiente Cliente/Servidor

## Arquitetura Three-Tier



## Arquitetura “Cloud”

“Native” to Cloud

“Cloud Capable”

Relational



ORACLE

PostgreSQL



... and other RDBMSs

Non-Relational



AppEngine  
Datastore



SimpleDB



MapReduce



HYPERTABLE



mongoDB



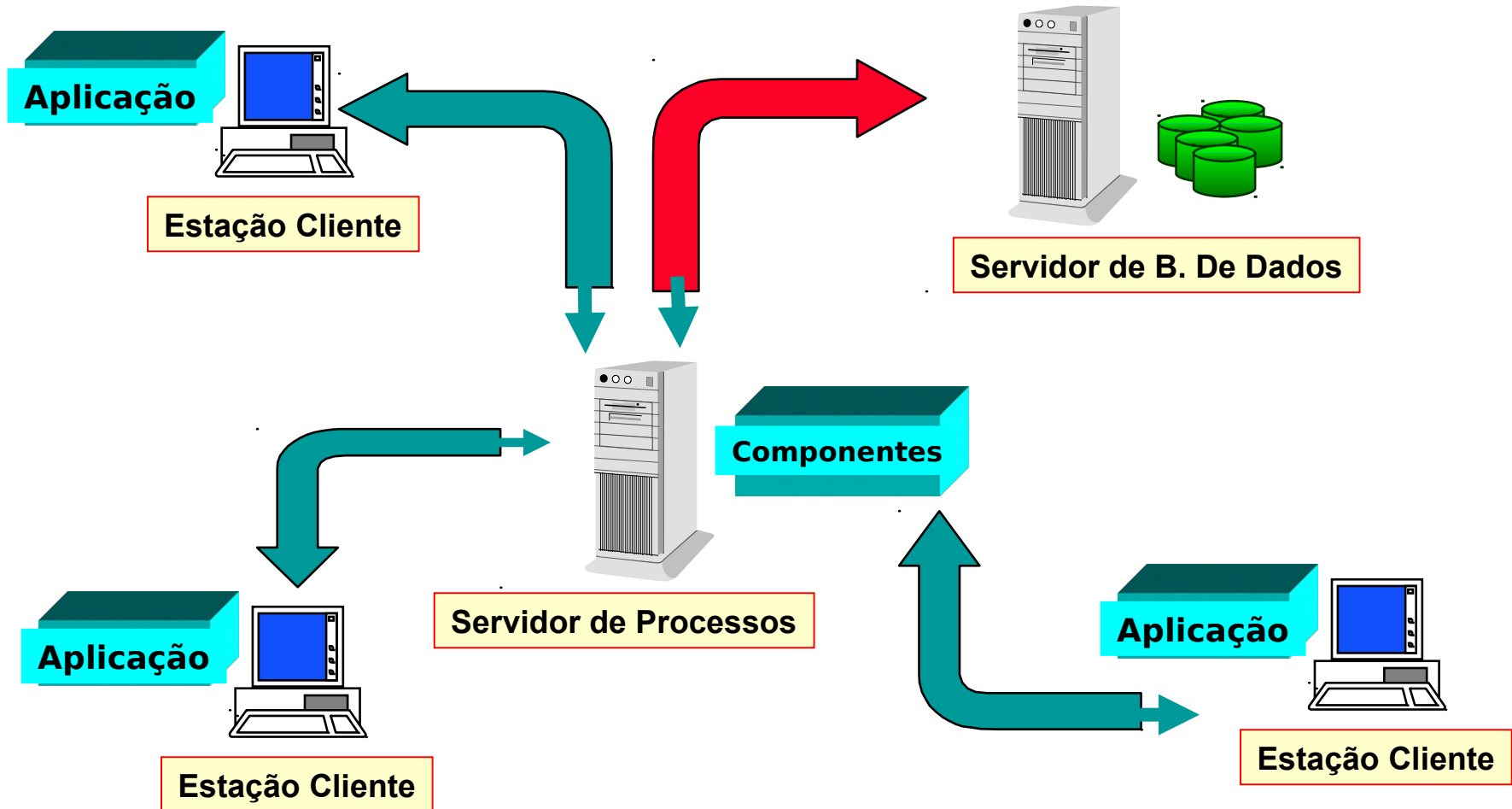
CouchDB  
relax

HBase

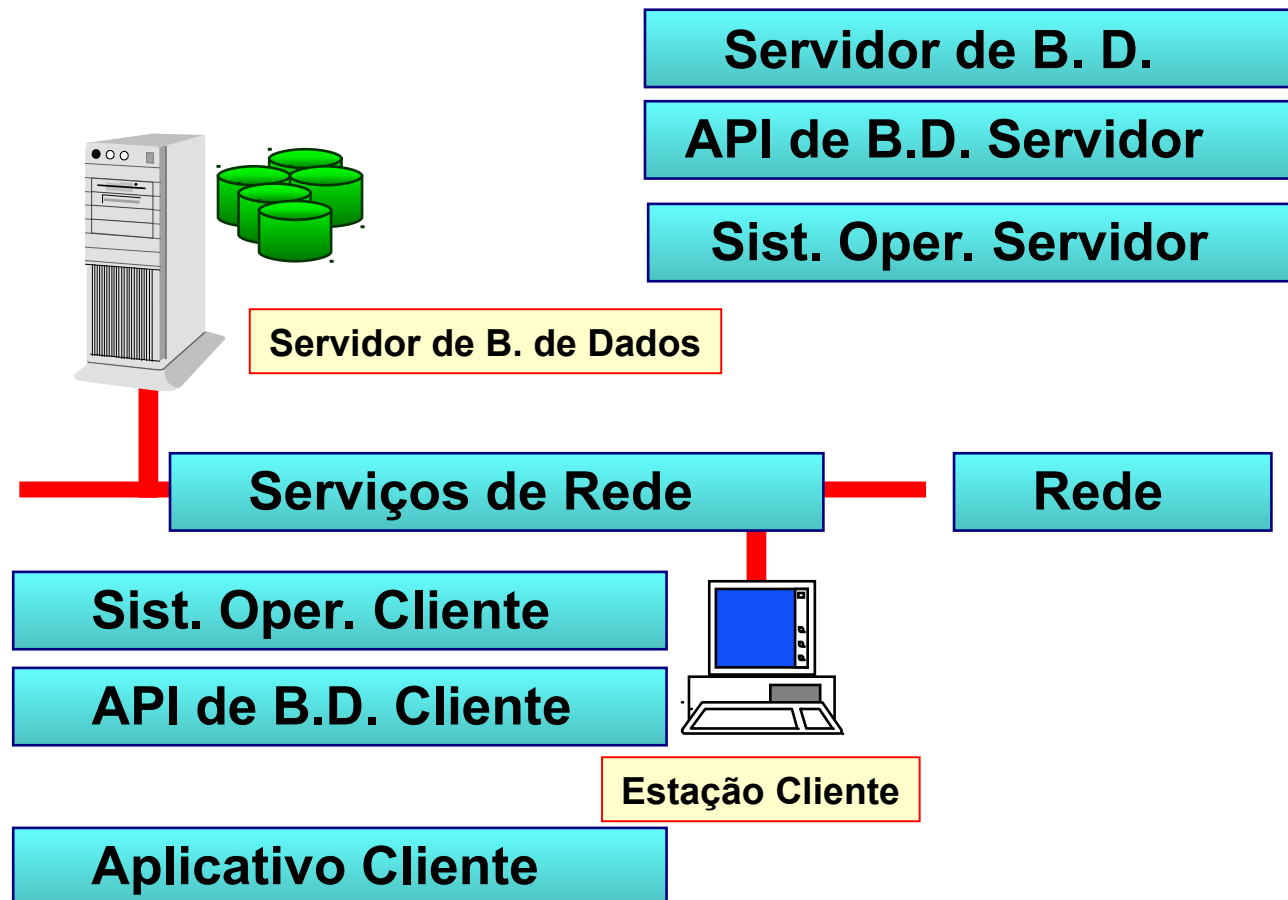
Project Voldemort

Project Cassandra

# Servidores de Processo



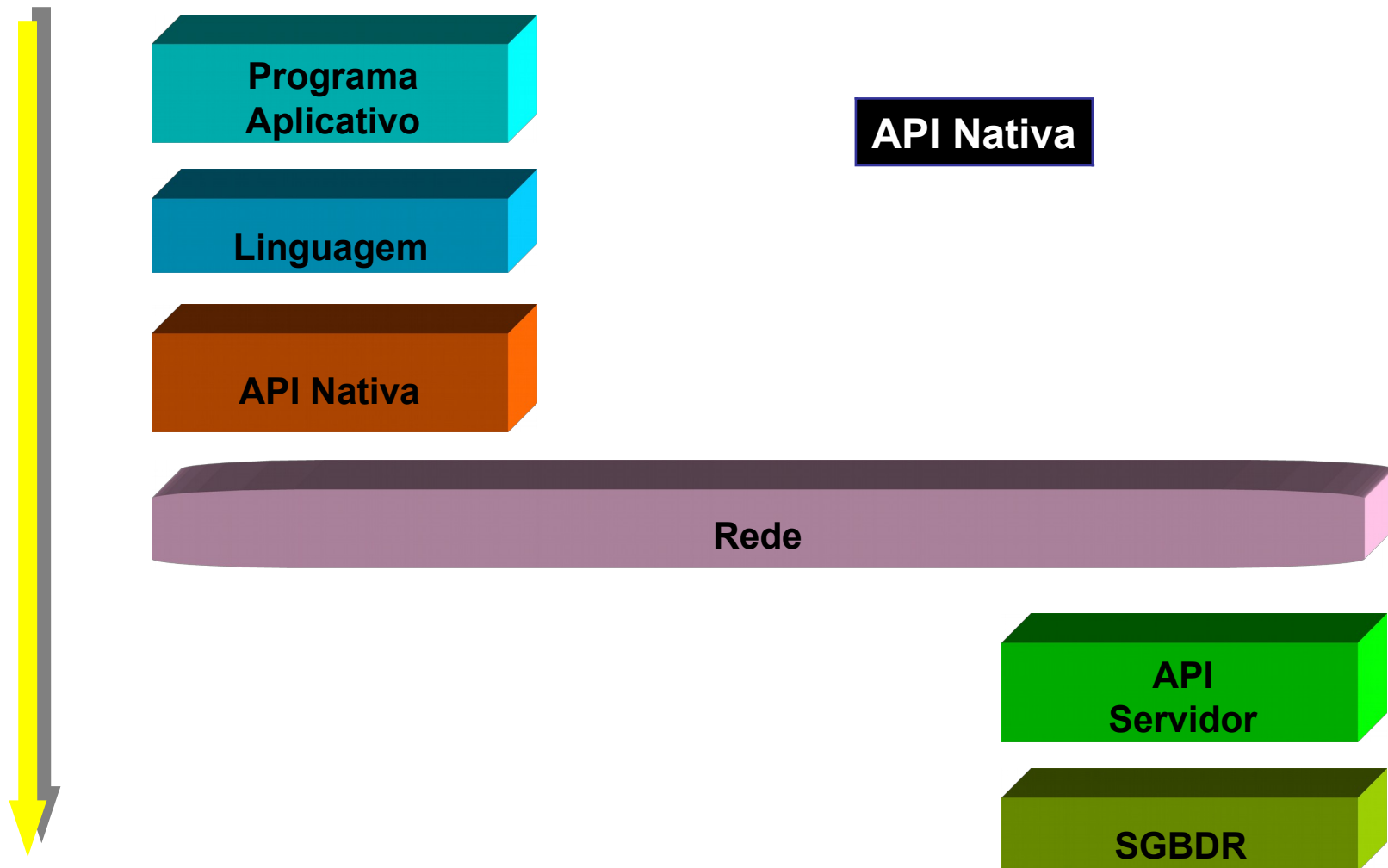
# Bancos de Dados Cliente/Servidor



# Bancos de Dados Cliente/Servidor

- API's Nativas
- ODBC
- OLEDB

# Bancos de Dados Cliente/Servidor



# Bancos de Dados Cliente/Servidor

## API (Access Programming Interface)

- **Possibilita o acesso completo à funcionalidade do SGBDR**
- **Permite a construção de aplicações específicas para um determinado SGBDR**

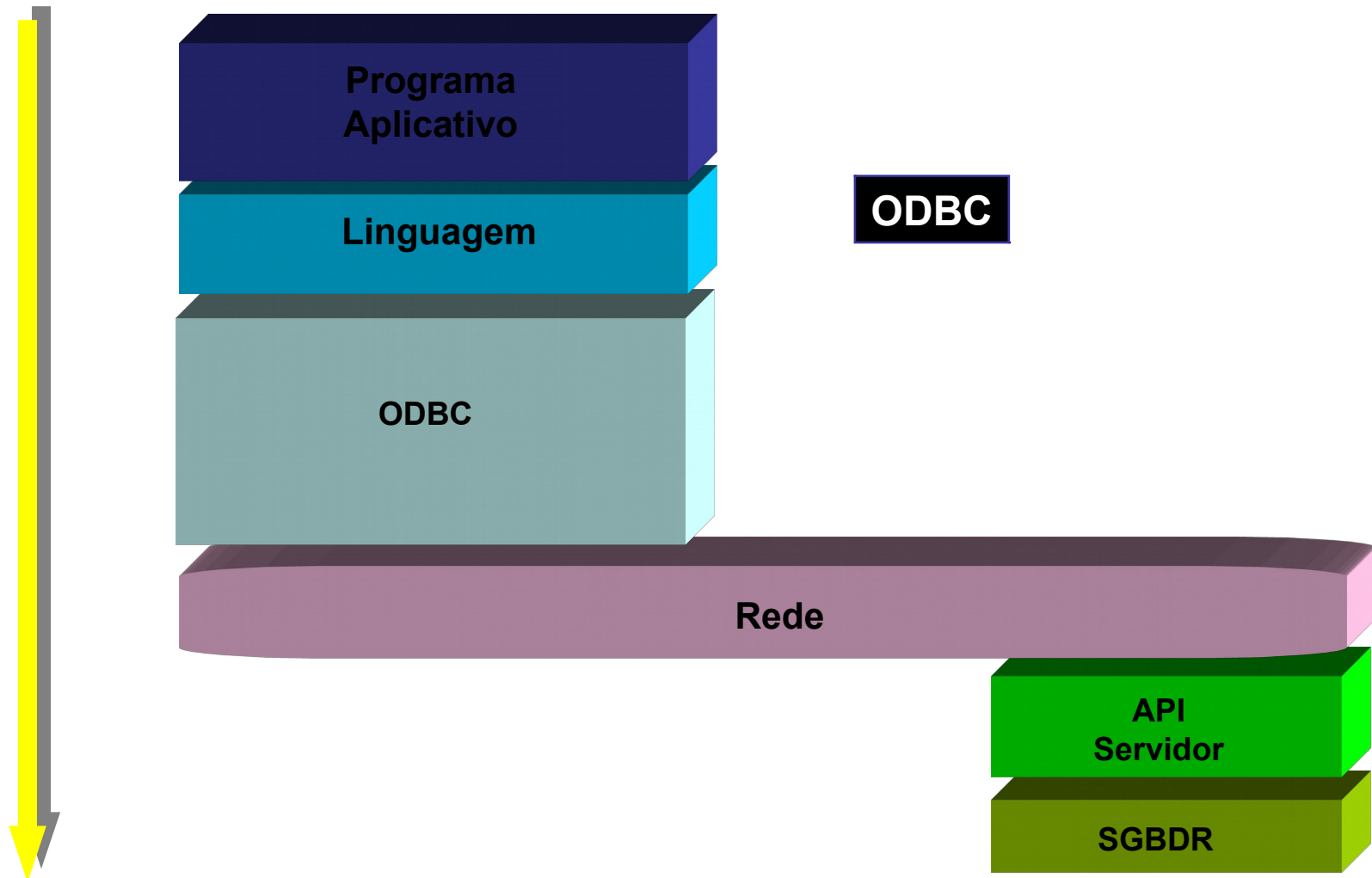
**'Get a connection for communicating with SQL Server**

**Sqlconn% = SqlOpen%( Login%, "server" )**

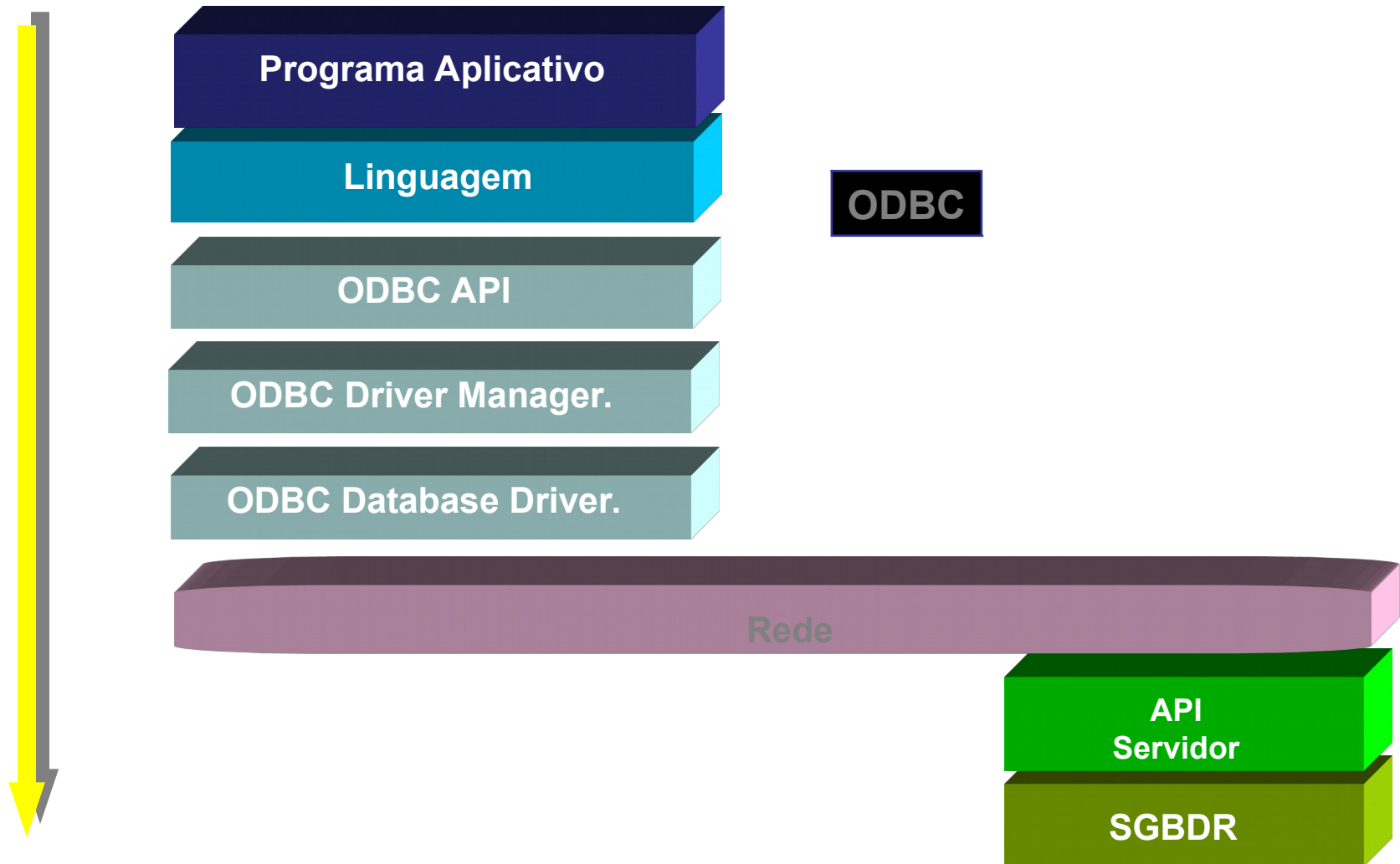
*Exemplo: MS Visual Basic utilizando MS SQL Server*



# Bancos de Dados Cliente/Servidor



# Bancos de Dados Cliente/Servidor



# Bancos de Dados Cliente/Servidor

## ODBC (Open Database Connectivity)

- Driver Manager
  - Carrega drivers específicos do Banco de Dados
  - Processa chamadas de inicialização da ODBC
  - Converte chamadas de funções ODBC feitas pela aplicação para o driver específico
  - Provê parâmetros e sequência de validação para chamadas ODBC
- Database Driver
  - Estabelece conexão ao Data Source
  - Submete requisições ao Data Source
  - Faz tradução de formatos de dados
  - Retorna resultados à aplicação

# Bancos de Dados Cliente/Servidor

## ODBC (Open Database Connectivity)

- Possibilita o acesso a diferentes SGBDs a partir de uma única aplicação.
- Implementa o padrão SQL ANSI (1989 e 1992).
- A performance aumentou significativamente, ampliando a utilização do padrão ODBC.
- Base para acesso de aplicativos 32 bits para gerenciadores de alguns fabricantes.
- Necessidade de utilização de comandos SQL específicos para utilização de recursos fornecidos pelo SGBD mas não implementados no driver ODBC disponível.
- Simplifica o desenvolvimento de aplicações

# Bancos de Dados Cliente/Servidor

## ODBC (Open Database Connectivity)

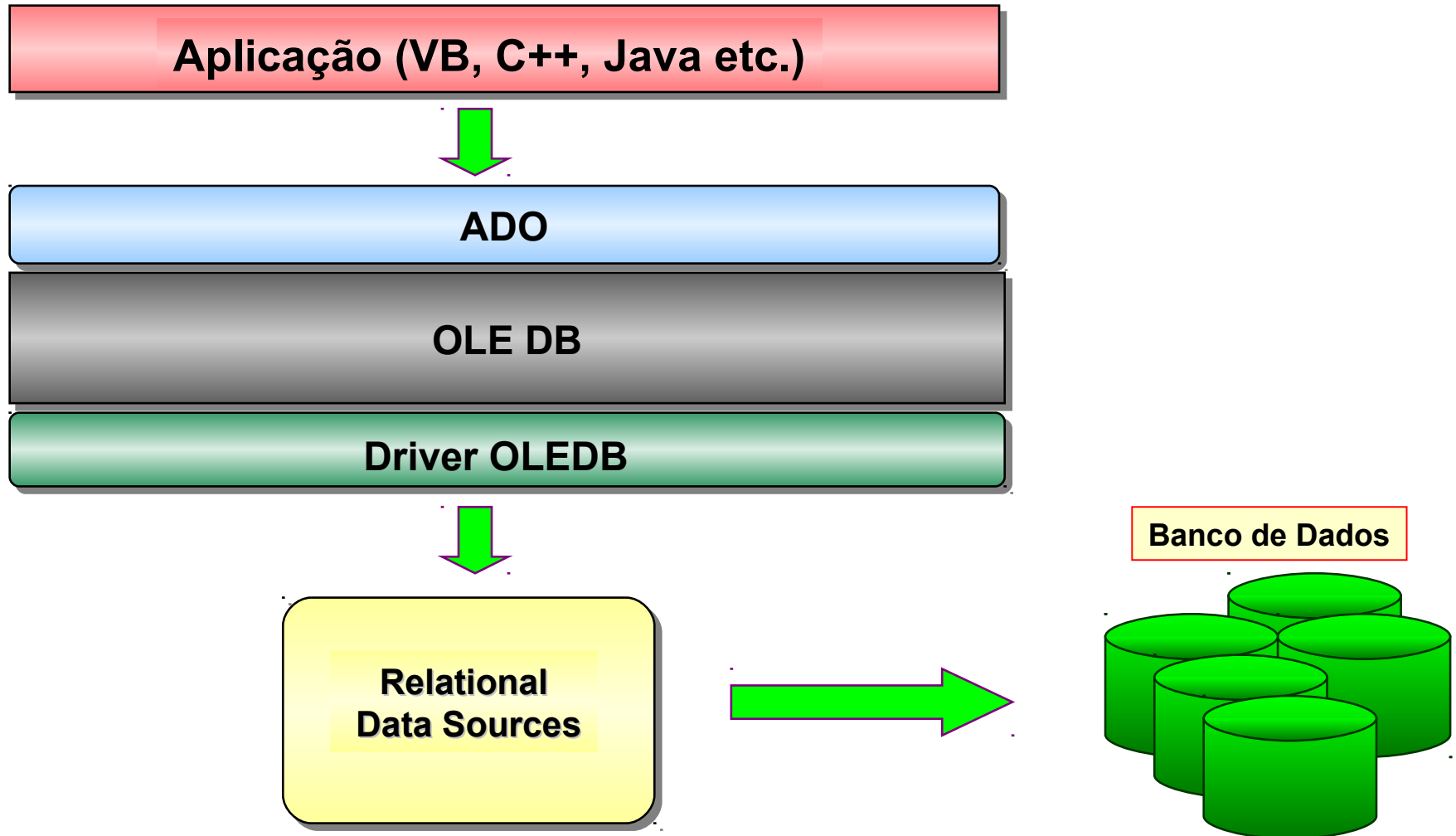
```
/* Connect to a data source */
```

```
retcode = SQLConnect (hdbc, "EmpData, SQL_NTS, "JohnS", SQL_NTS, "Sesame", SQL_NTS);
```

Exemplo: MS Visual C utilizando ODBC

# Bancos de Dados Cliente/Servidor

OLEDB (OLE Automation Interface)



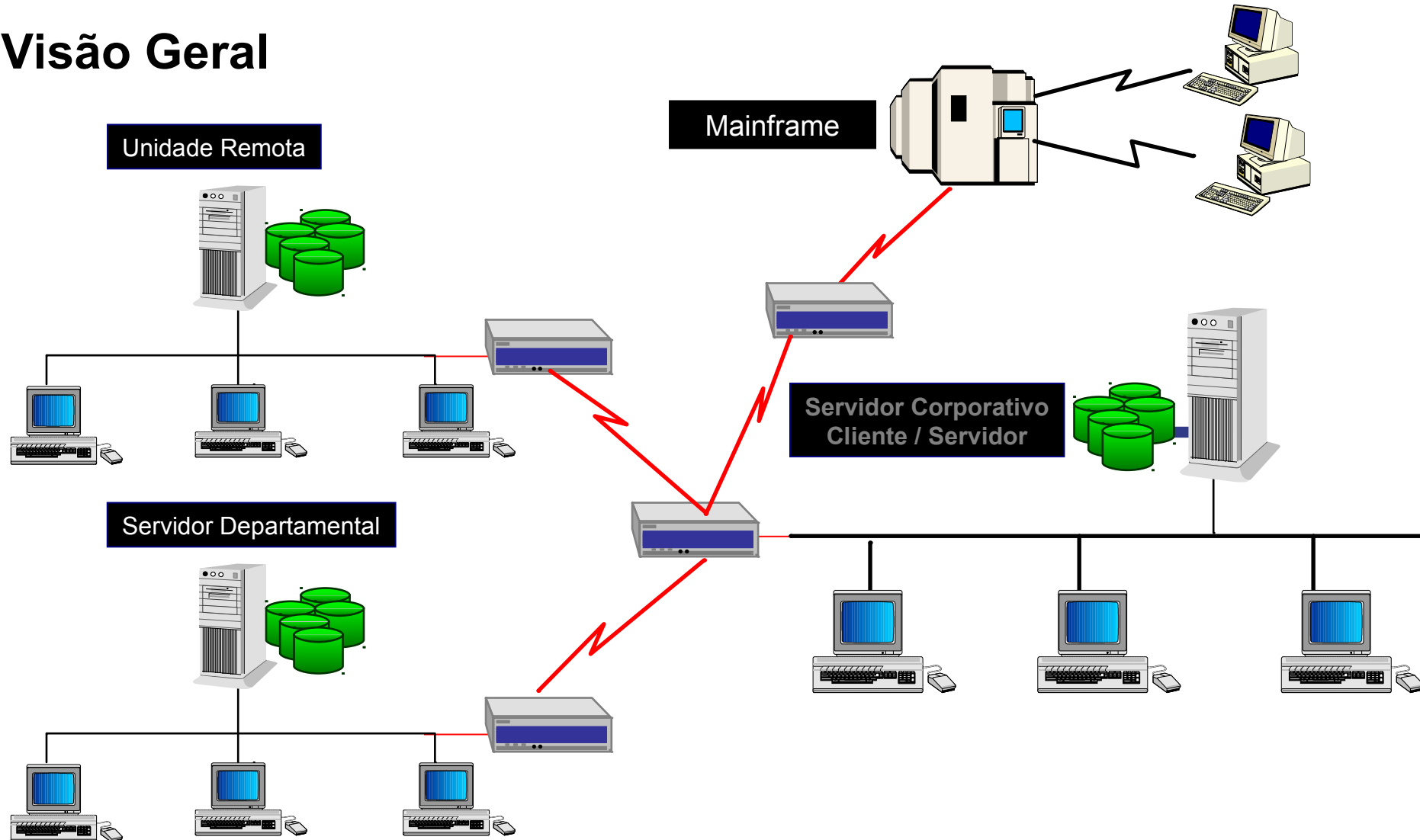
# Bancos de Dados Cliente/Servidor

OLEDB é uma API de baixo nível da Microsoft para acesso a diferentes fontes de dados

- Evita a duplicação de código
- Permite a construção de aplicativos em três camadas
- Permite a independência entre a aplicação e a manipulação de dados
- Permite a utilização de linguagens diferentes para aplicação e tratamento de dados

# Bancos de Dados Distribuídos

## Visão Geral



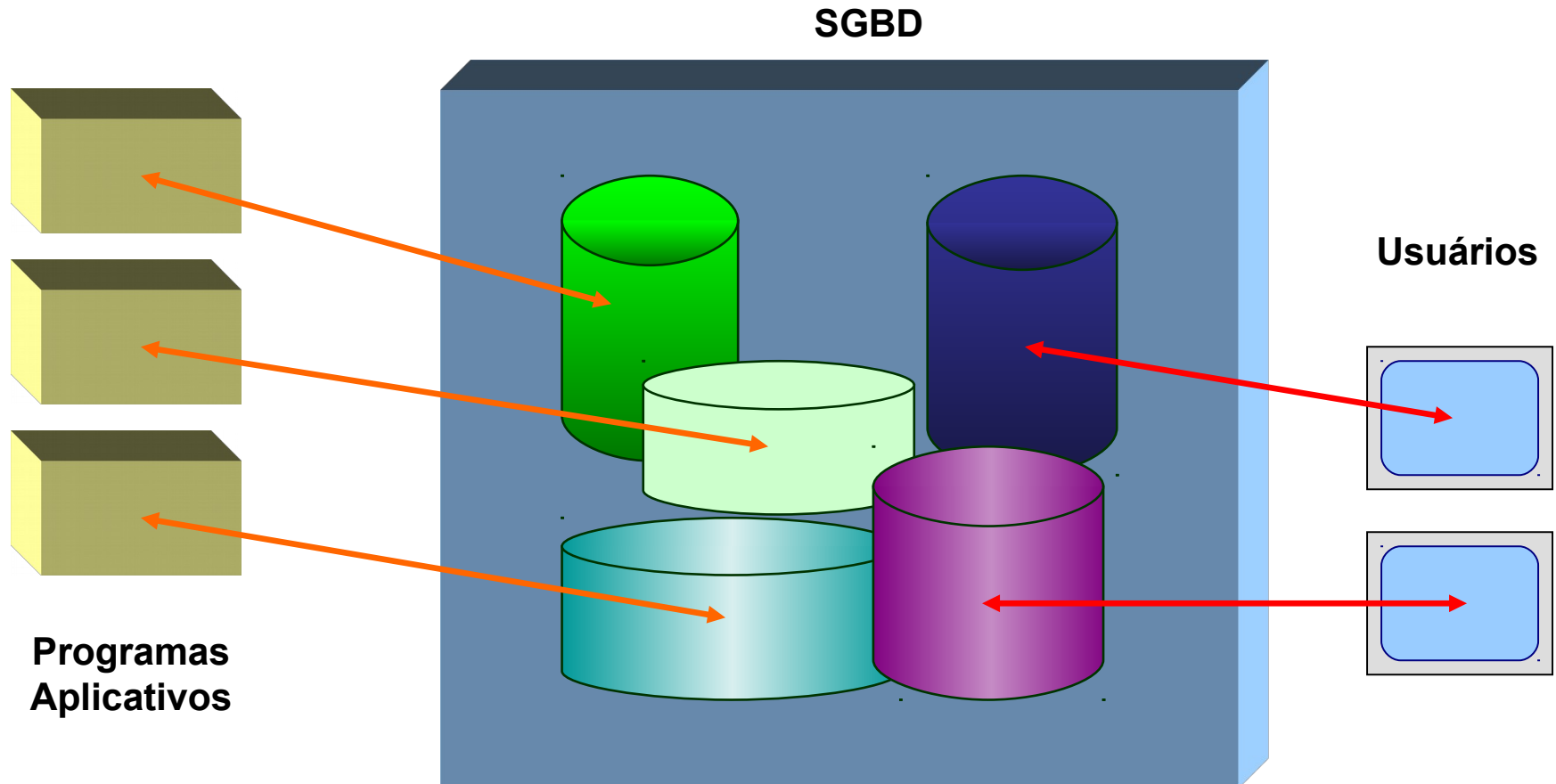


# Bancos de Dados: Conceitos

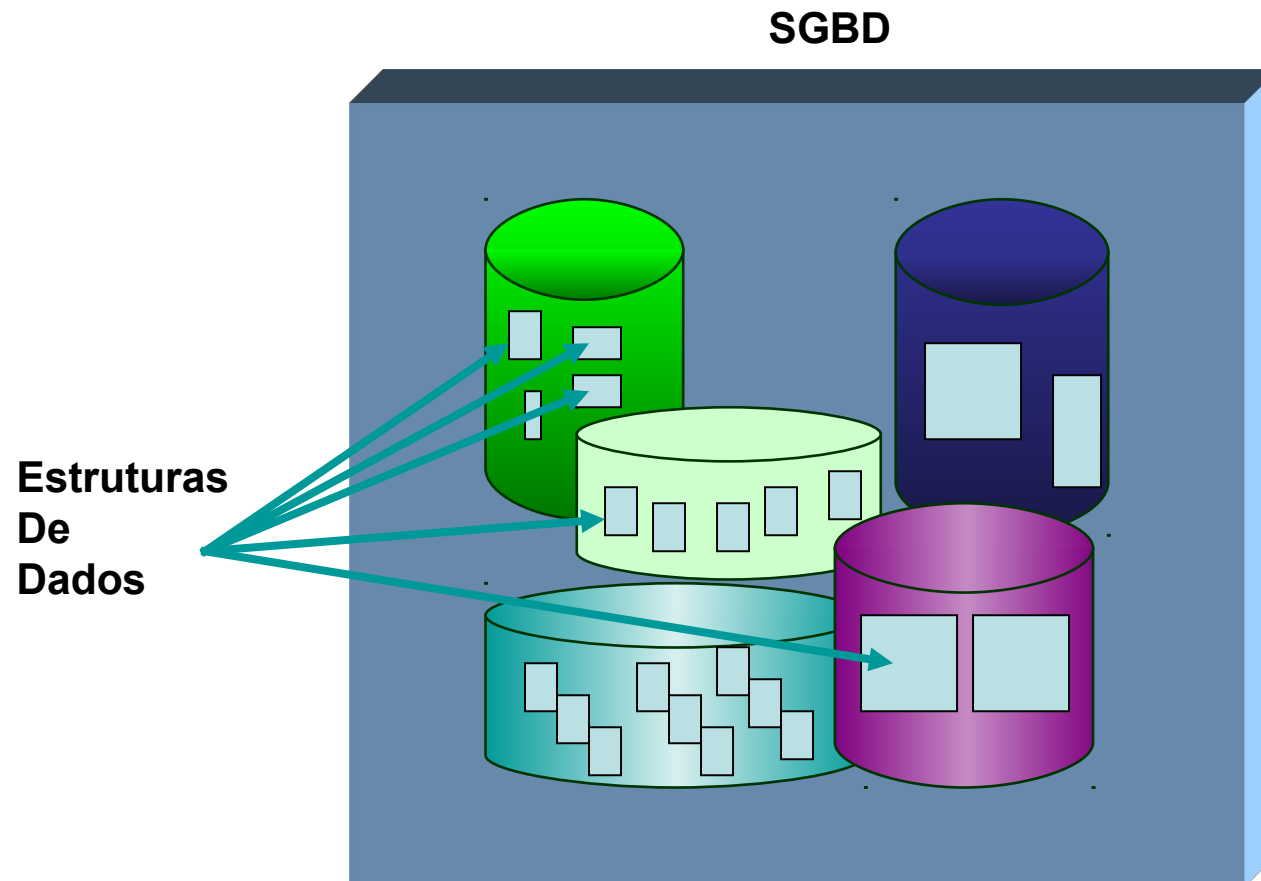
## SGBD – Sistemas Gerenciadores de Bancos de Dados

- Sistema computadorizado de guardar registros, cujo objetivo geral é manter informação e torná-la disponível sob demanda.
- Informação pode ser qualquer coisa que se considere ter relevância para o indivíduo ou organização a que o sistema deve servir, tudo que seja necessário para ajudar no processo geral de tocar os negócios desse indivíduo ou organização.

# Bancos de Dados: Conceitos



# Bancos de Dados: Conceitos



# Bancos de Dados: Conceitos

## Arquitetura de Implementação

- Hierárquico
- Rede
- Relacional

# Bancos de Dados Hierárquicos

**Peça**

**Número**

**Nome**

**Uso**

**Fornecedor**

**Fornecimento**

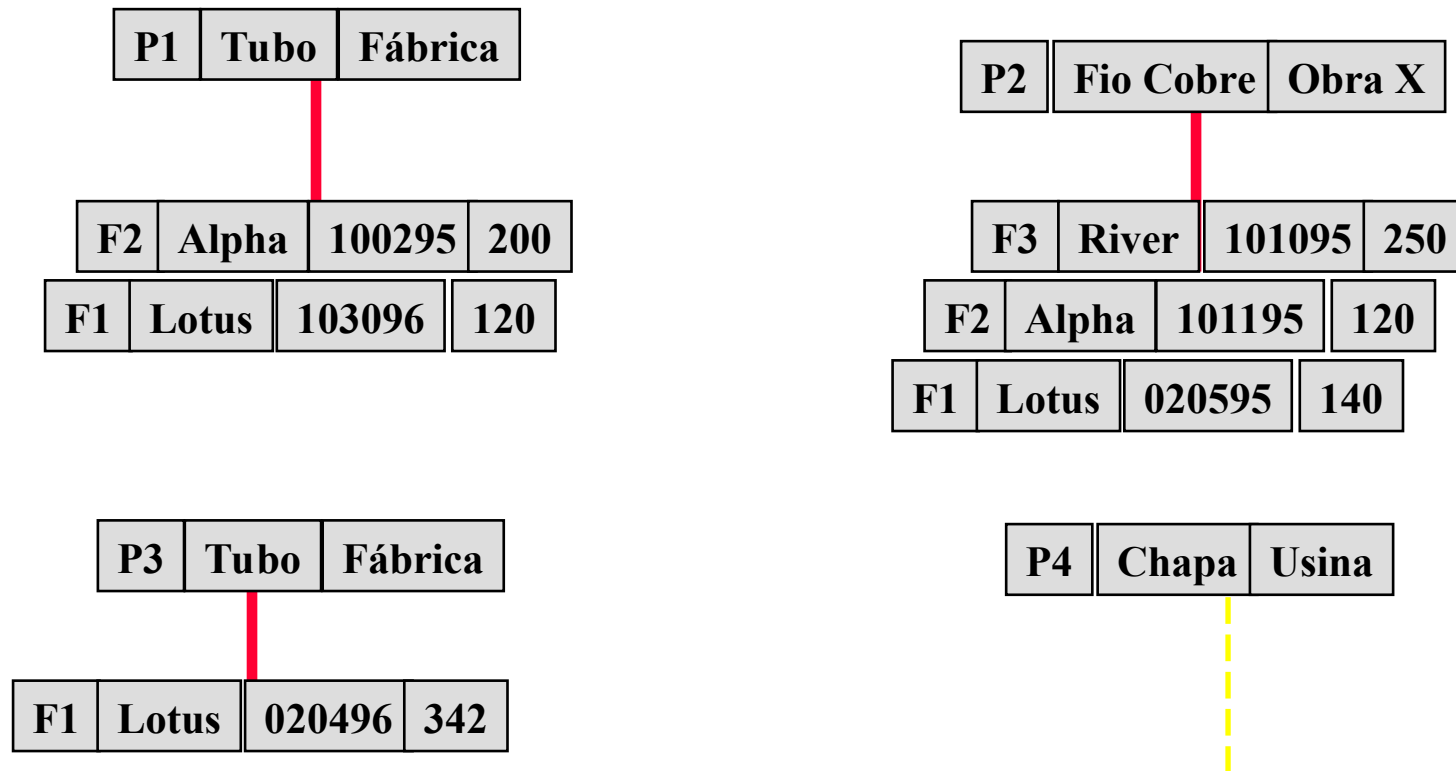
**Num.**

**Nome**

**Data Fornec.**

**Qte.**

# Bancos de Dados Hierárquicos



# Bancos de Dados Hierárquicos

- Estrutura em “árvore” :
  - RAIZ (peças)
  - REGISTROS DEPENDENTES (fornecedores)
- “Arquivo” complexo: diversos tipos de registros (dois neste caso, um para PEÇAS e outro para FORNECEDORES)
- Assimetria na recuperação dos dados:
  - qual o FORNECEDOR de uma PEÇA ?
  - que PEÇA é fornecida por um FORNECEDOR ?
- Ideal para estruturas 1:N (hierarquia de subordinação natural) sendo inadequado para os casos de N:N

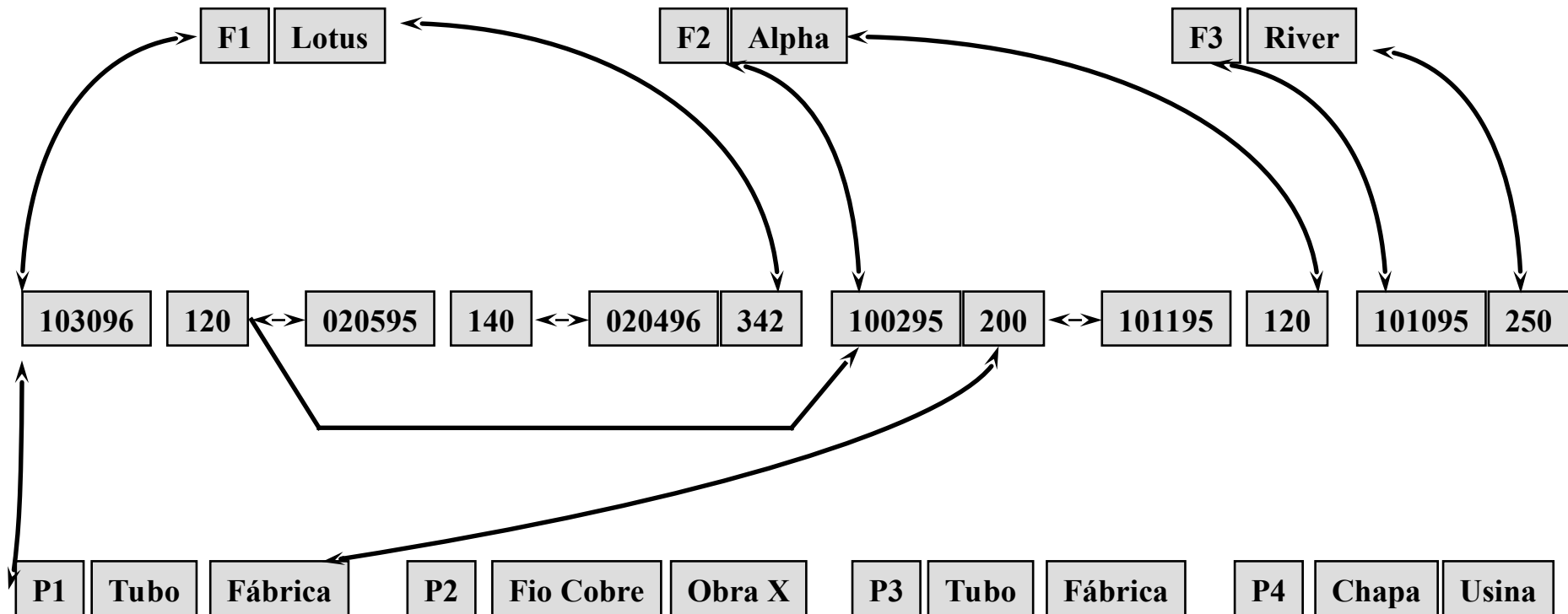
# Bancos de Dados Hierárquicos

## Anomalias:

- **INSERÇÃO:**
  - Não é possível sem a introdução de uma PEÇA fictícia , inserir dados de um FORNECEDOR se o mesmo não efetuar nenhum fornecimento.
- **REMOÇÃO:**
  - O fornecimento está associado ao FORNECEDOR. Se tivermos necessidade de remover um fornecimento e o FORNECEDOR tiver feito somente um fornecimento removeremos então todos os dados do FORNECEDOR
- **ATUALIZAÇÃO:**
  - Se houver necessidade de alterar, por exemplo, o endereço do FORNECEDOR F1, teremos possibilidade de introduzir inconsistências em razão da redundância destes dados.



# Bancos de Dados em Rede



# Bancos de Dados em Rede

- Os dados são representados por:
  - Registros (peças e fornecedores)
  - Interligações (fornecimento)
- A Interligação é um Conector, ou seja, a implementação de uma Associação (N:N)
- Estrutura tem mais simetria que a hierárquica

# Bancos de Dados Relacionais

## Objetos de Dados Relacionais

- Relação – Tabelas
- Tuplas – Linhas
- Cardinalidade – Quantidade de linhas
- Atributos – Colunas
- Grau – Quantidade de colunas
- Chave Primária – Identificador Único
- Domínio – Reservatório de valores, do qual atributos específicos, de relações específicas extraem seus valores reais

# Modelos de Implementação

## SGBD Relacional

- Baseados na teoria matemática das relações
- Evita detalhes de Implementação física
- Mantém abordagem abstrata pura
- Nomenclatura:
  - Relação = Tabela
  - Tupla = Linha
  - Domínios = Valores do item

## SGBD Hierárquico

- Estrutura dada pelo modelo físico
- Rota lógica de acesso exibida com ênfase

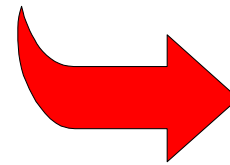
- Rota de acesso limita flexibilidade no projeto de banco
- Caso particular modelo de rede
- Nomenclatura
  - Registros = Segmentos
  - Itens = Campos
- Registros ascendentes e descendentes
- Os elos são 1:N

## SGBD em Rede

- Mais flexível no projeto que modelo hierárquico
- Suporta relação N:N
- Estruturação dados é complexa

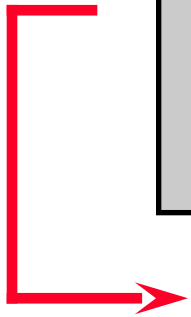
# Objetos de Dados Relacionais

## Representação das Tabelas (Relações)

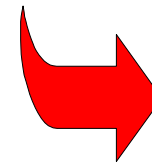


**Matricial**

Matrícula	Nome	Endereço	Data Nasc.
001	Maria	R. 5, 110	01.02.90
002	João	R. A, 200	02.07.70
003	José	Av. W3, 5	07.10.75



**Linhas (Tuplas)**



**Colunas  
(Atributos)**

# Objetos de Dados Relacionais

## Representação das Tabelas (Relações)

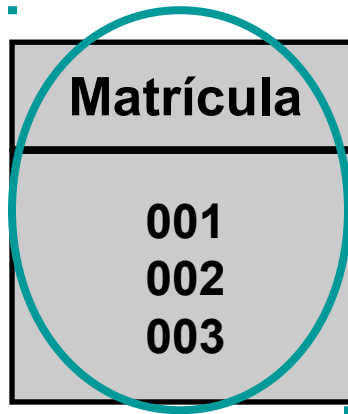
Matrícula	Nome	Endereço	Data Nasc.
001	Maria	R. 5, 110	01.02.90
002	João	R. A, 200	02.07.70
003	José	Av. W3, 5	07.10.75

Cardinalidade

Grau

# Objetos de Dados Relacionais

## Chave Primária

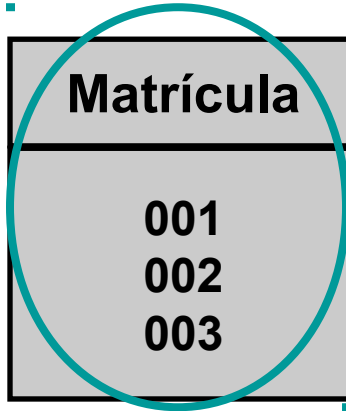


Matrícula	Nome	Endereço	Data Nasc.
001	Maria	R. 5, 110	01.02.90
002	João	R. A, 200	02.07.70
003	José	Av. W3, 5	07.10.75

Identificador único de uma Tupla (Linha)

# Objetos de Dados Relacionais

## Domínio



Matrícula	Nome	Endereço	Data Nasc.
001	Maria	R. 5, 110	01.02.90
002	João	R. A, 200	02.07.70
003	José	Av. W3, 5	07.10.75

**Conjunto de todos os números de matrícula possíveis. Qualquer número de matrícula que porventura existir na Tabela (Relação), estará restrito a ser um subconjunto deste conjunto.**



# Objetos de Dados Relacionais

## Propriedades da matriz de relação

- Cada linha representa uma tupla
- A ordem das linhas não é importante
- As linhas são distintas uma das outras
- A ordem das colunas não importa desde que sejam rotuladas convenientemente
- Um elemento da coluna pode ser vazio ou conter um único valor

## Objetos de Dados Relacionais

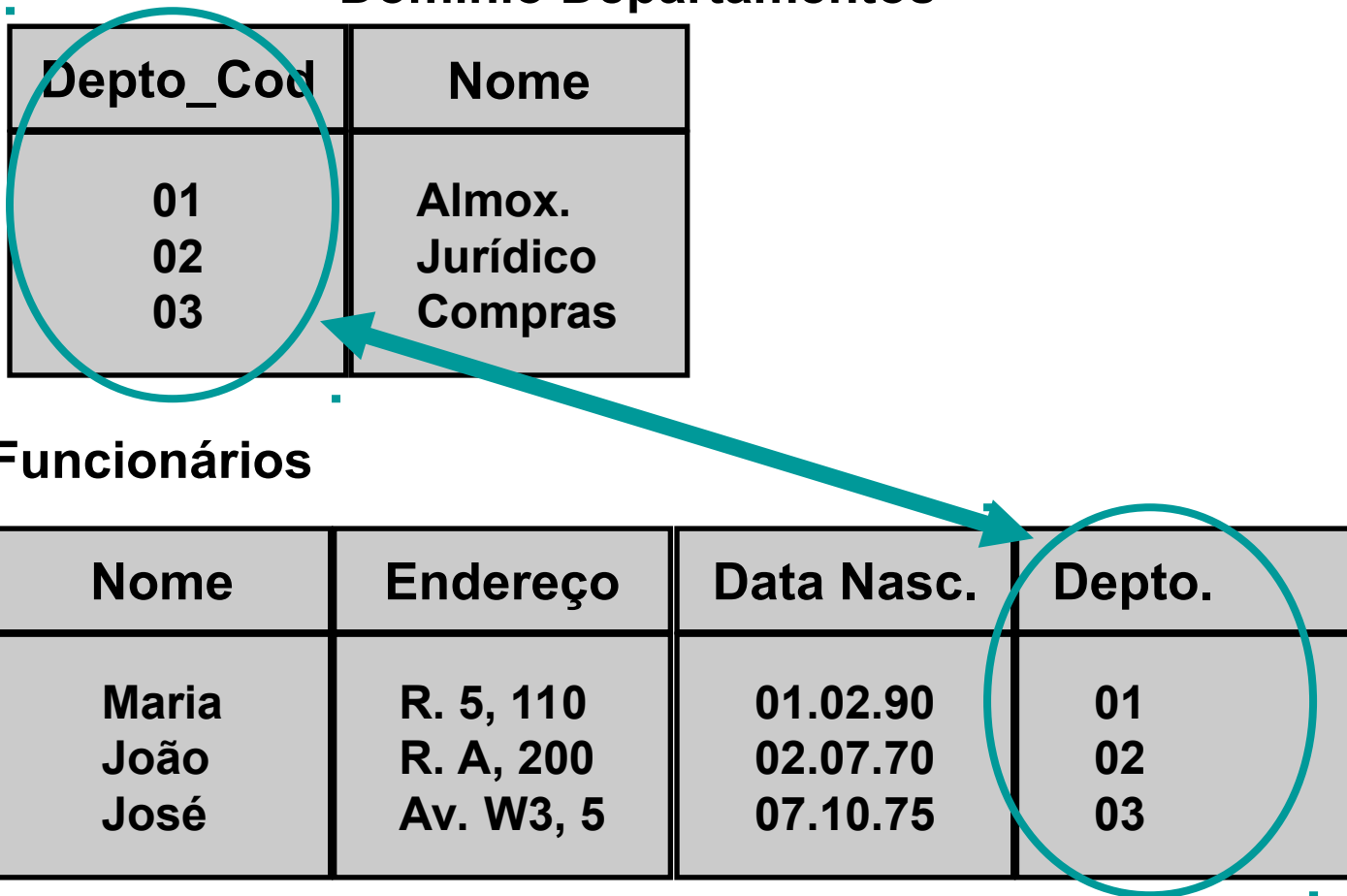
# Propriedades da matriz de relação

- Os valores que uma coluna pode ter, pertencem a um determinado domínio
- Duas colunas distintas podem ser definidas no mesmo domínio

**MATRIZ DA RELAÇÃO**  **TABELA RELACIONAL**

# Objetos de Dados Relacionais

## Domínio Departamentos



Depto_Cod	Nome
01	Almox.
02	Jurídico
03	Compras

## Domínio Funcionários

Matrícula	Nome	Endereço	Data Nasc.	Depto.
001	Maria	R. 5, 110	01.02.90	01
002	João	R. A, 200	02.07.70	02
003	José	Av. W3, 5	07.10.75	03

# Integridade de Dados

## Chaves Candidatas

Se  $R$  é uma relação (Tabela) e  $K$  é um subconjunto dos valores de  $R$ , podemos dizer que se  $K$  é chave candidata, deverá então possuir as seguintes características:

- Propriedade da Unicidade
  - Duas Tuplas (Linhas) de  $R$  nunca tem o mesmo valor para  $K$
- Propriedade da Irredutibilidade
  - Chaves primárias não podem ser nulas
  - Chaves primárias não podem ser alteradas

Estas propriedades dizem respeito principalmente ao conjunto de atributos e são a base para a formação das chaves primárias.

# Integridade de Dados

## Chaves Primárias

- Identifica unicamente uma tupla
  - Chave Primária X Chave Candidata
    - Quando existe mais de uma chave candidata em uma Relação (Tabela), uma delas deve ser eleita chave primária

# Integridade de Dados

## Restrições de Integridade

### Restrições de mundo real

- Fornecedores possuem códigos com quatro posições numéricas
- Status de fornecedor assume valores de 1 até 100
- Cidades de fornecedores devem existir em uma lista de cidades
- Todas as peças vermelhas devem estar em Belo Horizonte
- Etc.

# Integridade de Dados

## Restrições de Integridade

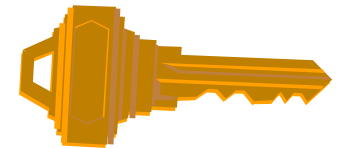
Asseguram que as mudanças feitas no Banco de Dados pelos usuários autorizados não irão resultar em perda da consistência dos dados

# Integridade de Dados

## Integridade de Entidade

O conteúdo de um campo chave não pode ser nulo

Member				
member_no	lastname	firstname	image	
1001	Anderson	Andrew	A	~~~
1002	Barr	Andrew	R	~~~
1003	Barr	Bill	NULL	~~~
1004	Anderson	Bill	B	~~~
1005	Anderson	Sally	A	~~~
1006	Henson	Jack	NULL	~~~





# Integridade de Dados

## Integridade de Entidade

O conteúdo de um campo chave não pode ser nulo

<i>Member</i>				
<i>member_no</i>	<i>lastname</i>	<i>firstname</i>	<i>middle_i</i>	<i>photo</i>
1001	Anderson	Andrew	A	~~~
1002	Barr	Andrew	R	~~~
1003	Barr	Bill	NULL	~~~
1004	Anderson	Bill	B	~~~
1005	Anderson	Sally	A	~~~
1006	Henson	Jack	NULL	~~~

Ex.: Matrículas podem assumir valores entre 1000 e 10000

# Integridade de Dados

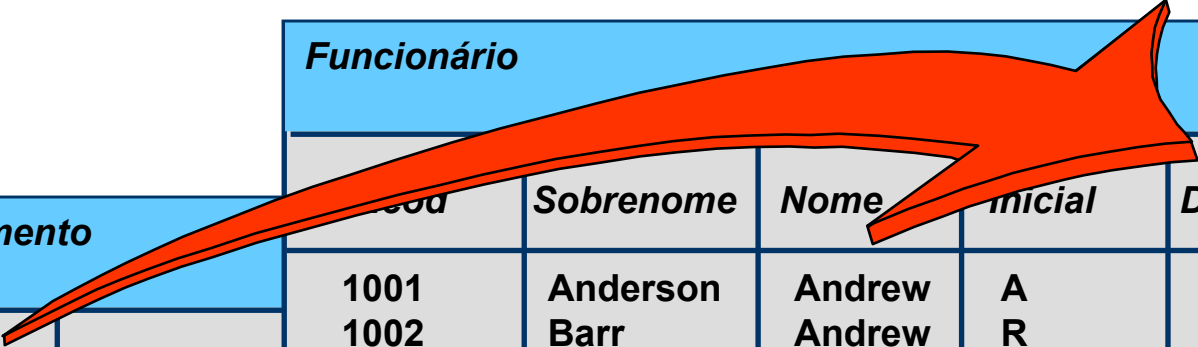
## Integridade Referencial

Define a validade de um atributo em relação a outro atributo

- Chaves Estrangeiras
  - É aquela chave existente em uma tabela que teve sua origem em outra chave de outra tabela (Chave Primária ou Chave Candidata)

# Integridade de Dados

## Integridade Referencial



Departamento		Funcionário				
Depcod	Departamento	Depcod	Sobrenome	Nome	Inicial	Depcod
01	Presidência	1001	Anderson	Andrew	A	03
02	Compras	1002	Barr	Andrew	R	03
03	Almoxarifado	1003	Barr	Bill	NULL	02
04	Oficina	1004	Anderson	Bill	B	01
		1005	Anderson	Sally	A	02
		1006	Henson	Jack	NULL	04
			Sally	A	~~~	
			Jack	NULL	~~~	

Ex.: Funcionários pertencem a departamentos válidos

# Álgebra / Cálculo Relacional

## Álgebra Relacional

Definida em operações sobre conjuntos

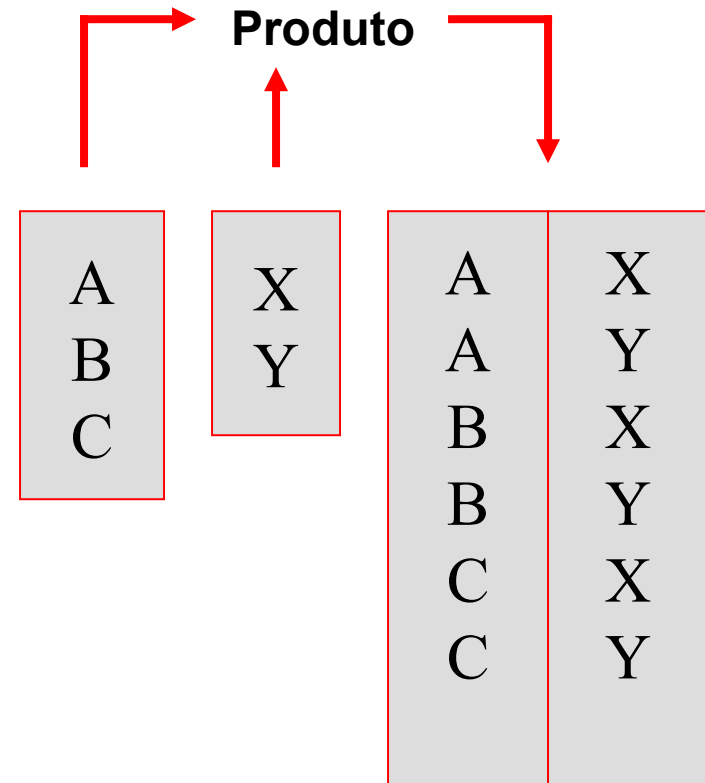
- Dois grupos de quatro operadores cada
  - Operações Tradicionais
    - União
    - Interseção
    - Diferença
    - Produto Cartesiano
  - Operações Especiais
    - Restringir
    - Projetar
    - Junção
    - Dividir

# Álgebra / Cálculo Relacional

**Restringir**

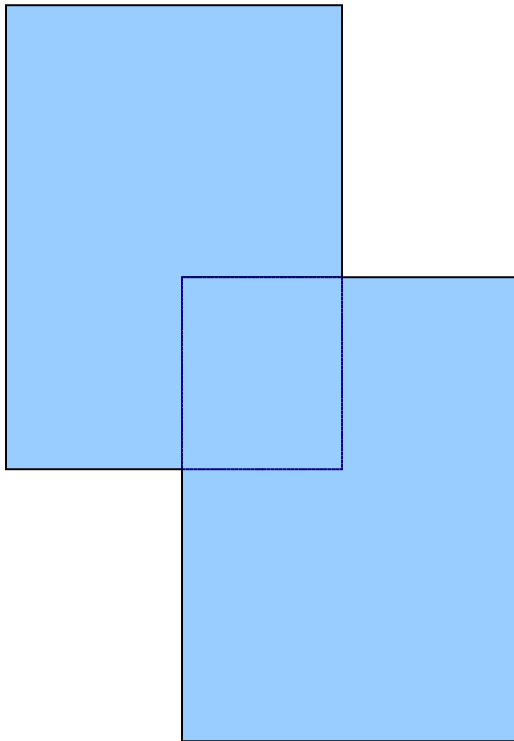

**Projetar**


**Produto**

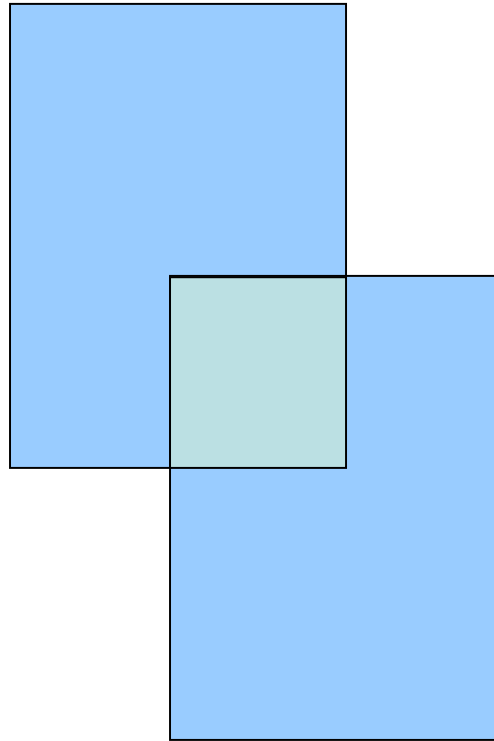


# Álgebra / Cálculo Relacional

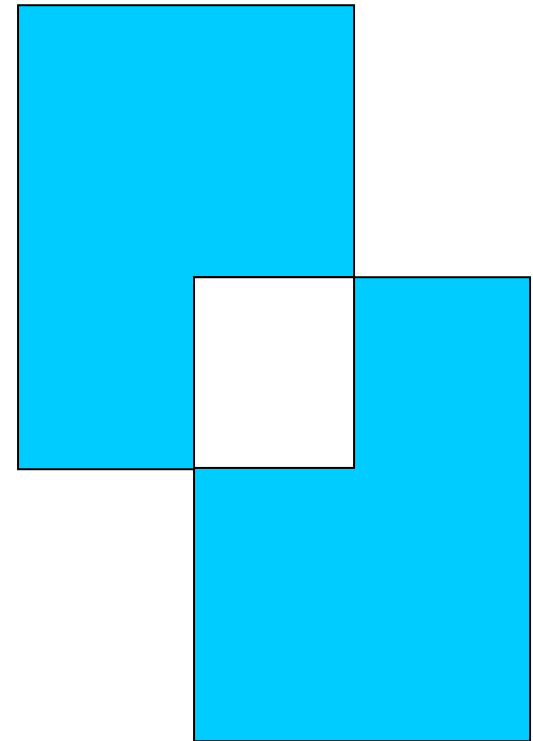
**União**



**Interseção**



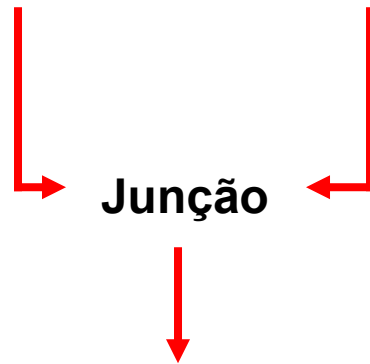
**Diferença**



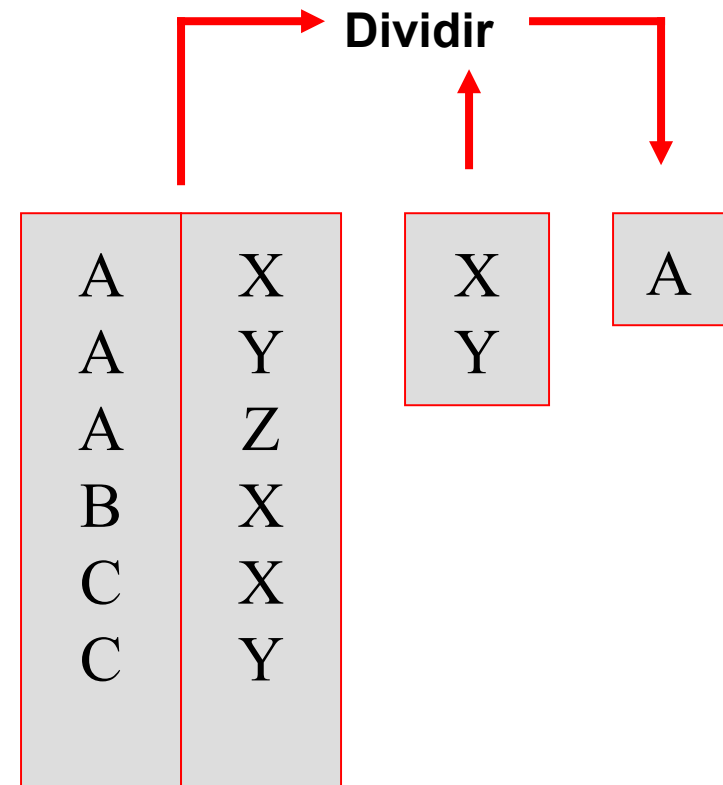
# Álgebra / Cálculo Relacional

A1	B1
A2	B1
A3	B2

B1	C1
B2	C2
B3	C3



A1	B1	C1
A2	B1	C1
A3	B2	C2



# Álgebra / Cálculo Relacional

## Cálculo Relacional

Equivalente a álgebra relacional, o cálculo relacional pode ser considerado descritivo, enquanto a álgebra é considerada prescritiva.

O cálculo relacional existe em duas formas:

- Cálculo de Tuplas
- Cálculo de Domínios

O cálculo relacional nos mostra de forma semântica, tal como uma linguagem esquemática, como serão efetuadas as operações relacionais.

O Cálculo Relacional é a base para linguagens relacionais como o SQL – Structured Query Language



# Linguagem SQL

SQL – Structured Query Language

- Forma padronizada de se especificar conjuntos de dados
- Forma padronizada de se manipular e recuperar dados em um banco de dados
- Utilizada em todas as funções do banco de dados, inclusive manuseio de dados e administração de B.D's
- Pode ser utilizada na forma “Embedded SQL” via um programa aplicativo

# Linguagem SQL

- DDL – Data Definition Language
  - Utilizada para definir as estruturas de dados e os objetos de bancos de dados como tabelas, Views, Triggers etc.
- DML – Data Manipulation Language
  - Utilizada para manipular os dados em Bancos de dados, promovendo operações como inserção, atualização, exclusão e consulta de dados.
- DCL – Data Control Language
  - Utilizada para efetuar operações de concessão e revogação de permissões sobre os objetos de Bancos de Dados

# Linguagem SQL

## 1 - Criação de Tabelas (DDL)

```
CREATE TABLE publishers  
(  
  pub_id      CHAR (4)          NOT NULL,  
  pub_name    VARCHAR (40)      NULL,  
  city        VARCHAR (20)      NULL ,  
  state       CHAR (2)          NULL  
)
```

## 2 - Colocação de dados em Tabelas (DML)

```
INSERT INTO publishers  
VALUES  
( "0736", "New Moon Books", "Boston", "MA" )
```

# Linguagem SQL

## 3 – Concessão de Permissões (DCL)

```
GRANT SELECT,  
        INSERT,  
        UPDATE,  
        DELETE  
ON authors TO sergio
```

# Projeto de Bancos de Dados

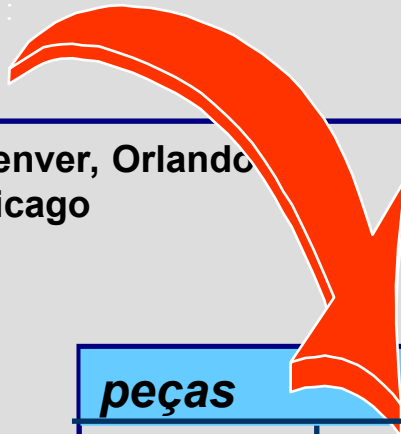
## Normalização

### 1ª Forma Normal

Uma tabela está na 1ª forma normal quando possui apenas um valor em cada célula.

# Projeto de Bancos de Dados

<i>peças</i>	
<i>peça_nº</i>	<i>Depósito</i>
T232 H995	Chicago, Denver, Orlando Denver, Chicago



<i>peças</i>				
<i>peça_nº</i>	<i>Depósito</i>	<i>Qtd. Est.</i>	<i>Telefone</i>	<i>Nº Emb.</i>
T232	Chicago	467	222-9876	47
T232	Denver	31	654-8900	13
T232	Orlando	45	554-9900	15

1ª Forma Normal

# Projeto de Bancos de Dados

<i>peças</i>				
<i>peça_nº</i>	<i>Depósito</i>	<i>Qtd. Est.</i>	<i>Telefone</i>	<i>Nº Emb.</i>
T232	Chicago	467	222-9876	47
T232	Denver	31	654-8900	13
T232	Orlando	45	554-9900	15
T233	Chicago	115	222-9876	17

Problemas da 1ª Forma Normal :

- A base de dados ocupa mais espaço, uma vez que a mesma informação se repete
- A atualização de informações torna-se mais trabalhosa, pois deverá ser feita em diversos lugares

# Projeto de Bancos de Dados

## Normalização

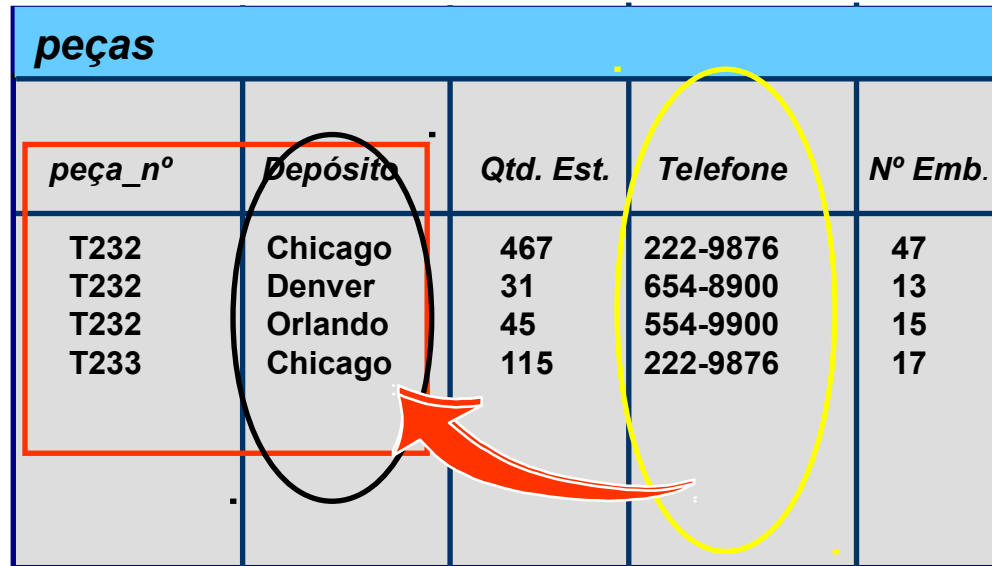
### 2ª Forma Normal

- Uma tabela está na 2ª forma normal quando está na 1ª forma normal.
- Quando todos os outros atributos de uma tabela dependerem apenas da chave primária como um todo e não de apenas parte da chave (Chave primária composta).



# Projeto de Bancos de Dados

peças				
peça_nº	Depósito	Qtd. Est.	Telefone	Nº Emb.
T232	Chicago	467	222-9876	47
T232	Denver	31	654-8900	13
T232	Orlando	45	554-9900	15
T233	Chicago	115	222-9876	17



2ª Forma Normal ?

# Projeto de Bancos de Dados

<i>peças</i>			
<i>peça_nº</i>	<i>Depósito</i>	<i>Qtd. Est.</i>	<i>nº Emb.</i>
T232	Chicago	467	47
T232	Denver	31	13
T232	Orlando	45	15
T233	Chicago	115	17

<i>Depósito</i>	
<i>Nome</i>	<i>Telefone</i>
Chicago Denver	222-9876 654-8900

2ª Forma Normal

# Projeto de Bancos de Dados

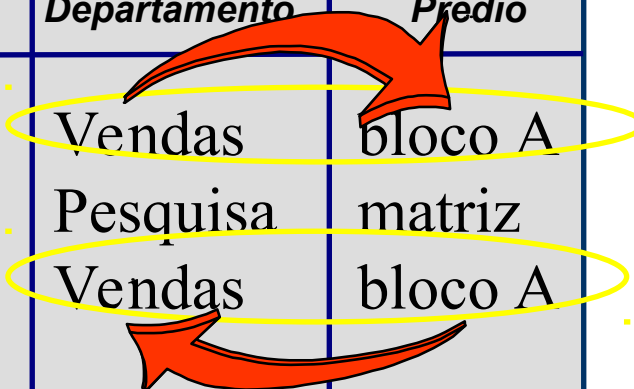
## Normalização

### 3ª Forma Normal

- Uma tabela está na 3ª forma normal quando está na 2ª forma normal e conseqüentemente na 1ª forma normal.
- Quando todos os outros atributos que não fazem parte da chave, não possuem relacionamentos entre si.

# Projeto de Bancos de Dados

Local de trabalho		
Empregado_nº	Departamento	Prédio
26622	Vendas	bloco A
41156	Pesquisa	matriz
33987	Vendas	bloco A



3ª Forma Normal ?

# Projeto de Bancos de Dados

Dep. de empregados	
<i>Empregado_nº</i>	<i>Departamento</i>
26622	Vendas
41156	Pesquisa
33987	Vendas

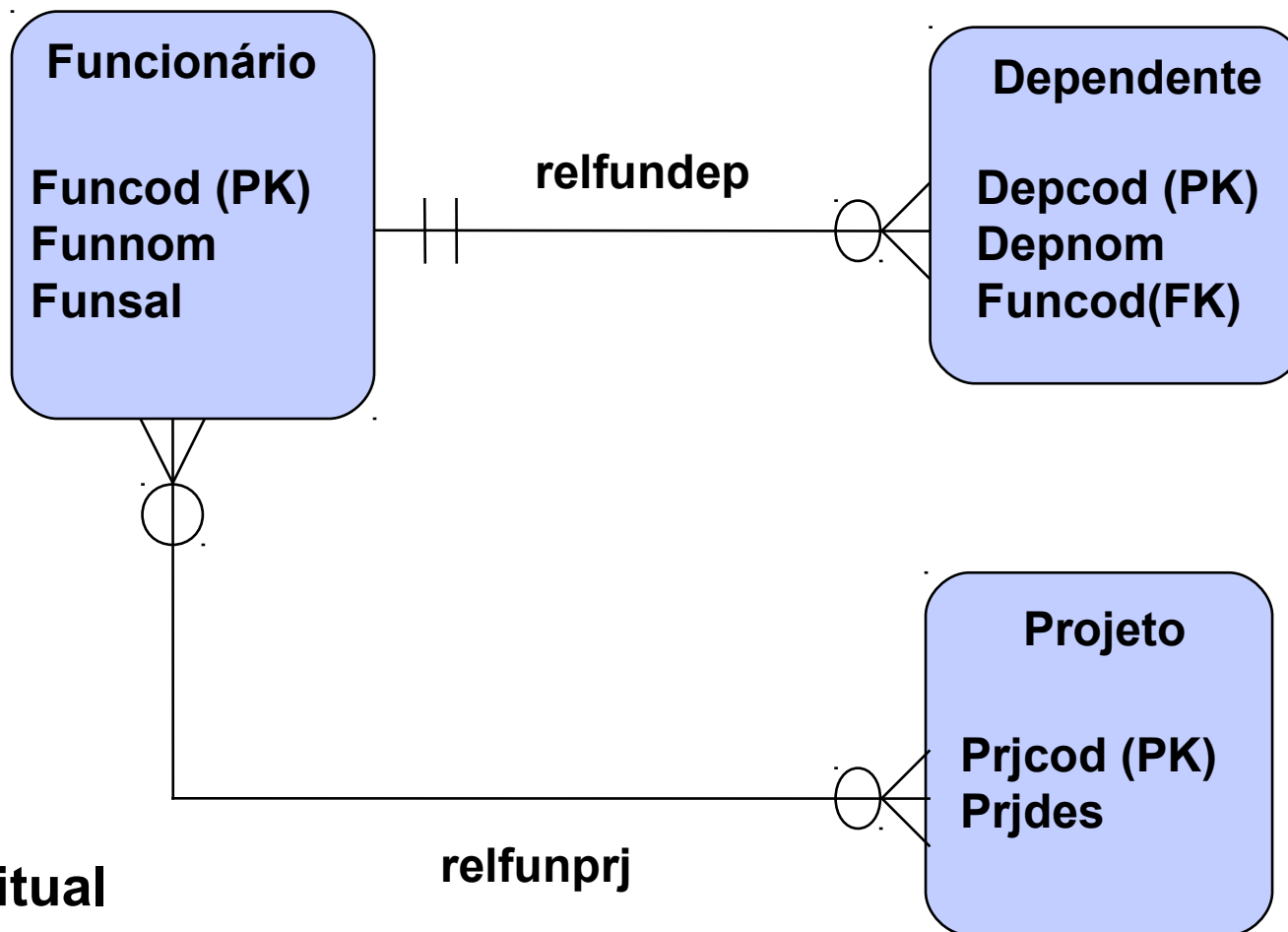
Prédio	
<i>Departamento</i>	<i>Prédio</i>
Vendas	bloco A
Pesquisa	matriz

3ª Forma Normal

# Projeto de Bancos de Dados

## Modelo de Entidade e Relacionamento

D.E.R. – Diagrama de Entidade e Relacionamento

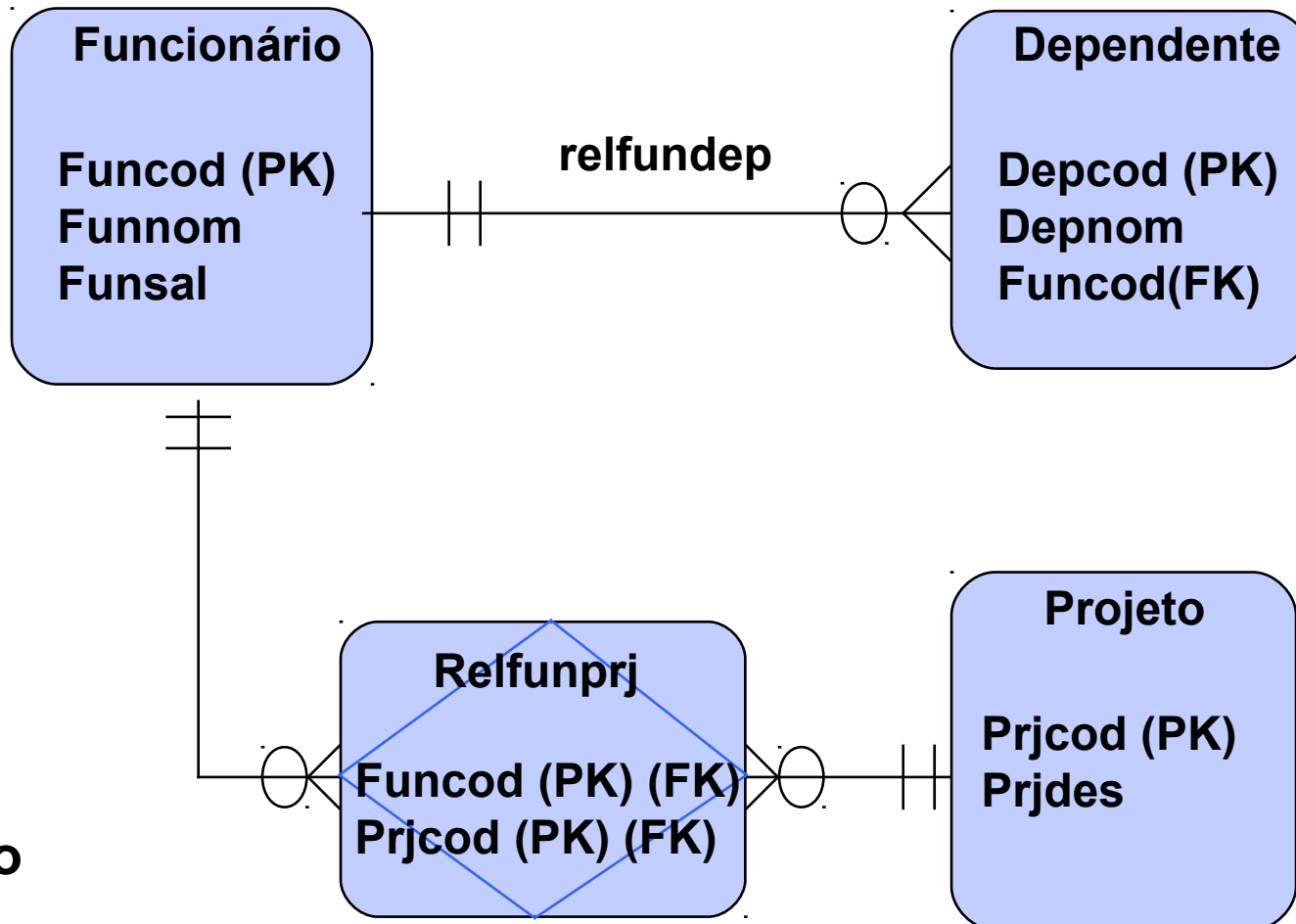


**Conceitual**

# Projeto de Bancos de Dados

## Modelo de Entidade e Relacionamento

D.E.R. – Diagrama de Entidade e Relacionamento



Lógico

# Projeto de Bancos de Dados

## Modelo de Entidade e Relacionamento

### D.E.R. – Diagrama de Entidade e Relacionamento

#### Relação de atributos por entidade

##### Entidade : Dependente

**Nome:** DEPCOD **Tipo:** Data Element **Nulidade:** NOT NULL

**Tipo de Dado:** INT **Tamanho:** 5

**Código do Funcionário**

**Nome:** DEPNOM **Tipo:** Data Element **Nulidade:** NOT NULL

**Tipo de Dado:** CHAR **Tamanho:** 50

**Nome do Dependente**

**Nome:** FUNCOD **Tipo:** Data Element **Nulidade:** NOT NULL

**Tipo de Dado:** INT **Tamanho:** 50

**Código do Funcionário**

##### Entidade : Funcionário

...

## Lógico



# Projeto de Bancos de Dados

## Modelo de Entidade e Relacionamento

```
/* SQL Product = SQL Server V7 */
/* Environment Case of column names = Map to lower case */
/* Environment Case of other names = Retain case */
/* Environment Prefixes = Prefix Column names */
/* Environment Indexes = CLUSTER based on property */

.....
CREATE TABLE dependente (
            depcod                int            NOT NULL,
            depnom                char(50)       NOT NULL,
            funcod                int            NOT NULL
            )

.....
ALTER TABLE dependente ADD CONSTRAINT PK_dependente
        PRIMARY KEY CLUSTERED (depcod)

.....
ALTER TABLE DEPENDENTE ADD CONSTRAINT FK_relfundep
        FOREIGN KEY (funcod)
        REFERENCES funcionario (funcod)
```

Físico

# **Projeto de Bancos de Dados**

## **Proteção de Dados**

## **Recuperação**

Em Bancos de Dados, recuperação significa a capacidade de recuperar um banco de dados a um estado íntegro, depois de alguma falha que o tornou incorreto ou pelo menos suspeito.

# **Projeto de Bancos de Dados**

## **Proteção de Dados**

## **Recuperação**

Existem várias formas de garantir este estado de integridade.

Um dos conceitos básicos que viabilizam a recuperação é o conceito de transação.

# Projeto de Bancos de Dados

## Proteção de Dados

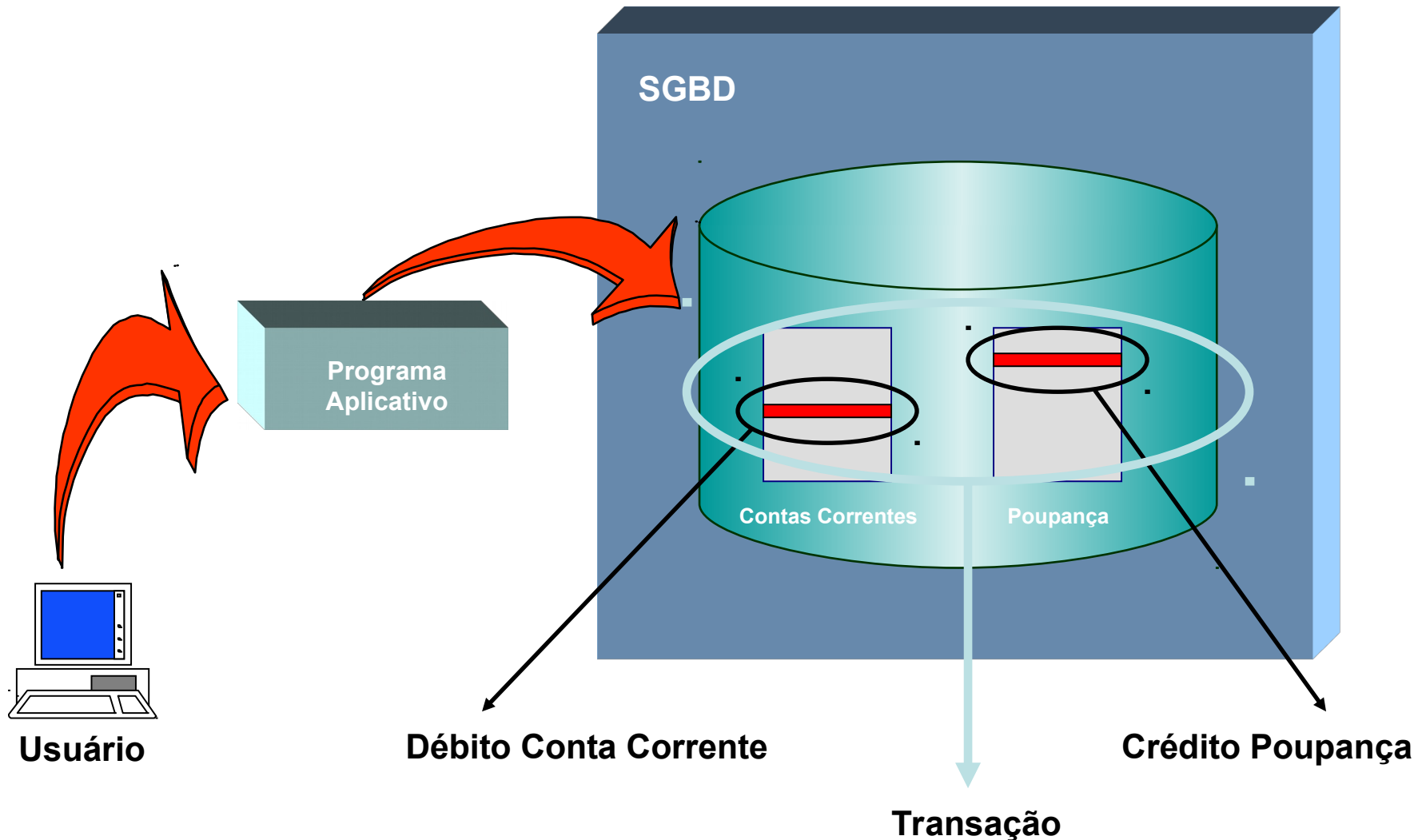
### Transação

Transação é uma unidade lógica de trabalho.

Para melhor entendermos o conceito vamos analisar o seguinte exemplo:

Uma transação bancária de débito em conta corrente e crédito em conta poupança

# Projeto de Bancos de Dados



# Projeto de Bancos de Dados

## Proteção de Dados

### Transação

#### Propriedades ACID

- Atomicidade (Tudo ou Nada)
- Consistência (Transforma o Banco de um estado consistente para outro estado consistente)
- Isolamento (Uma transação é isolada de outra transação)
- Durabilidade (Uma transação completa sobrevive, é durável)

<b>Atomic</b> Atômica	Uma transação, mesmo sendo um conjunto de ações, deve ser executada como uma unidade única. Uma transação deve ser executada exatamente uma única vez, sem nenhuma dependência.
<b>Consistent</b> Consistente	Ao término de uma transação, o sistema deve ser deixado em um estado consistente, respeitando as integridades de dados e relacional da base sendo manipulada.
<b>Isolated</b> Isolada	Isto significa que cada transação, não importando quantas transações estão sendo executadas neste momento no banco de dados, deve parecer ser independente de todas as outras transações. Imagine caixas eletrônicos separados, onde, diferentes pessoas querem executar a mesma operação bancária para a mesma conta. Transações processando duas ações concorrentes devem comportar-se como se cada uma delas estivessem operando com acesso exclusivo à base de dados. Na prática, nós sabemos que as coisas não são tão simples assim. Nós trataremos deste tópico nos próximos artigos.
<b>Durable</b> Durável	Uma vez que a transação seja completada, ela deve permanecer completada. Uma vez que o dinheiro tenha sido transferido com sucesso entre as contas, ele deve permanecer transferido, mesmo que a máquina que está rodando o gerenciador de banco de dados pare por falta de energia elétrica.

# Projeto de Bancos de Dados

## Proteção de Dados

### Concorrência - Os três problemas

- Atualização Perdida
  - Duas transações  $tr1$  e  $tr2$  recuperam o mesmo dado em um instante  $t1$ , mas procedem atualizações diferentes em instantes diferentes,  $t2$  e  $t3$ , usando os dados do instante  $t1$  como base. Assim sendo os dados de atualização submetidos por último sobrescrevem os dados atualizados pela primeira transação.



# Projeto de Bancos de Dados

## Proteção de Dados

### Concorrência - Os três problemas

- Atualização não efetivada (Dirty Read)
  - A transação tr2 recupera dados atualizados e ainda não efetivados pela transação tr1, de forma que se a transação tr1 não for completada com sucesso, a transação tr2 está com dados inválidos (Phantom Values).

# Projeto de Bancos de Dados

## Proteção de Dados

### Concorrência - Os três problemas

- Análise Inconsistente
  - Durante o processo de recuperação de dados feito pela transação tr2, a transação tr1 altera os valores recuperados, provocando uma inconsistência, caso a transação tr2 grave seu resultado de volta no Banco de Dados.

# Projeto de Bancos de Dados

## Proteção de Dados

### Concorrência – Níveis de Isolamento

- Read Committed
- Repeatable Read
- Serializable

Definição do Nível de Isolamento ANSI / ISO	Dirty Read	Unrepeatable Read	Phantom
<b>Read Uncommitted</b>	<i>Possible</i>	<i>Possible</i>	<i>Possible</i>
<b>Read Committed</b>	<i>Not Possible</i>	<i>Possible</i>	<i>Possible</i>
<b>Repeatable Read</b>	<i>Not Possible</i>	<i>Not Possible</i>	<i>Possible</i>
<b>Serializable</b>	<i>Not Possible</i>	<i>Not Possible</i>	<i>Not Possible</i>

# Projeto de Bancos de Dados

## Segurança

- Login\_id
  - Garante acesso ao Servidor
- Username
  - Garante acesso ao Banco de Dados

# Projeto de Bancos de Dados

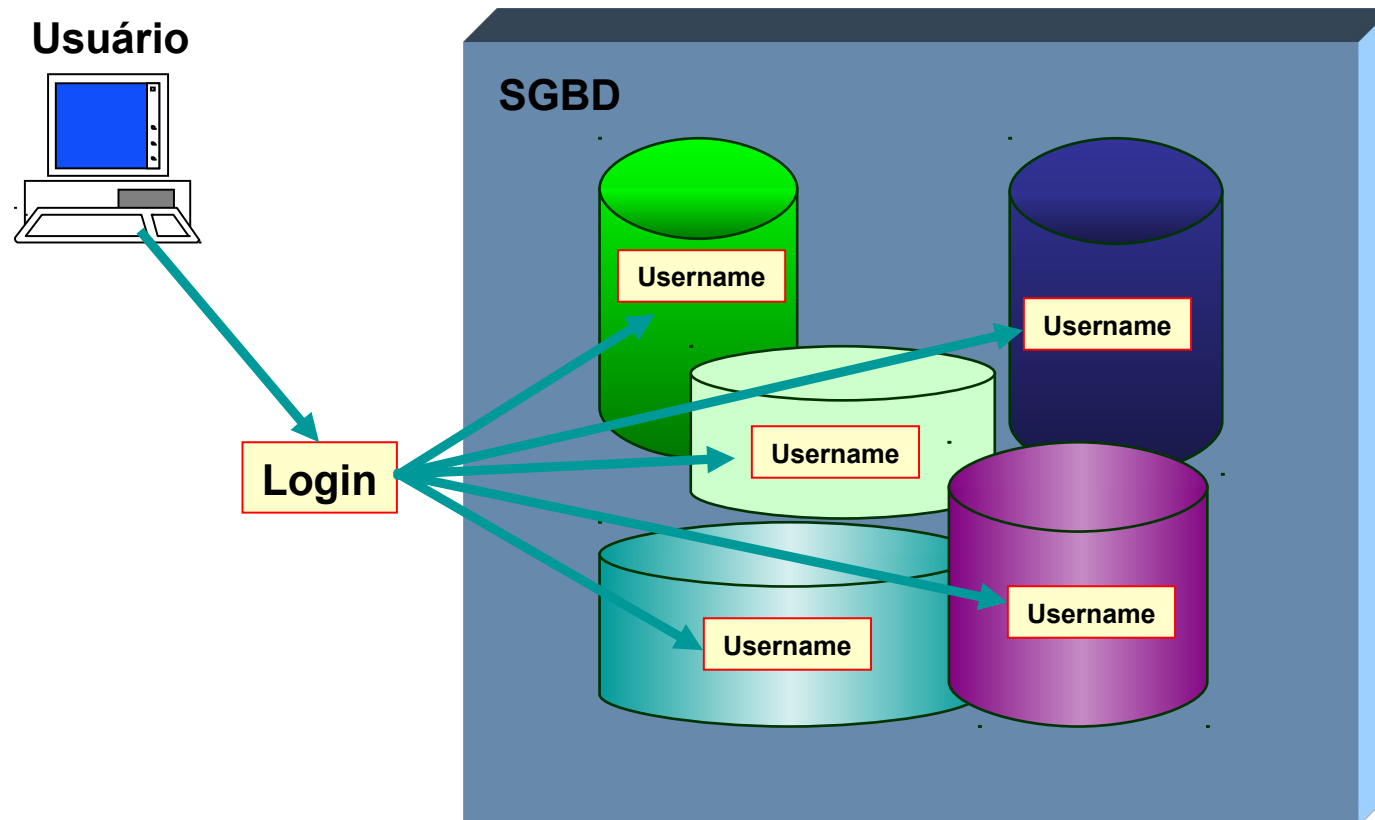
## Segurança

### Hierarquia de Permissões

<b>Administrador</b>
<b>Proprietário do Banco de dados</b>
<b>Proprietário de Objetos</b>
<b>Usuário</b>

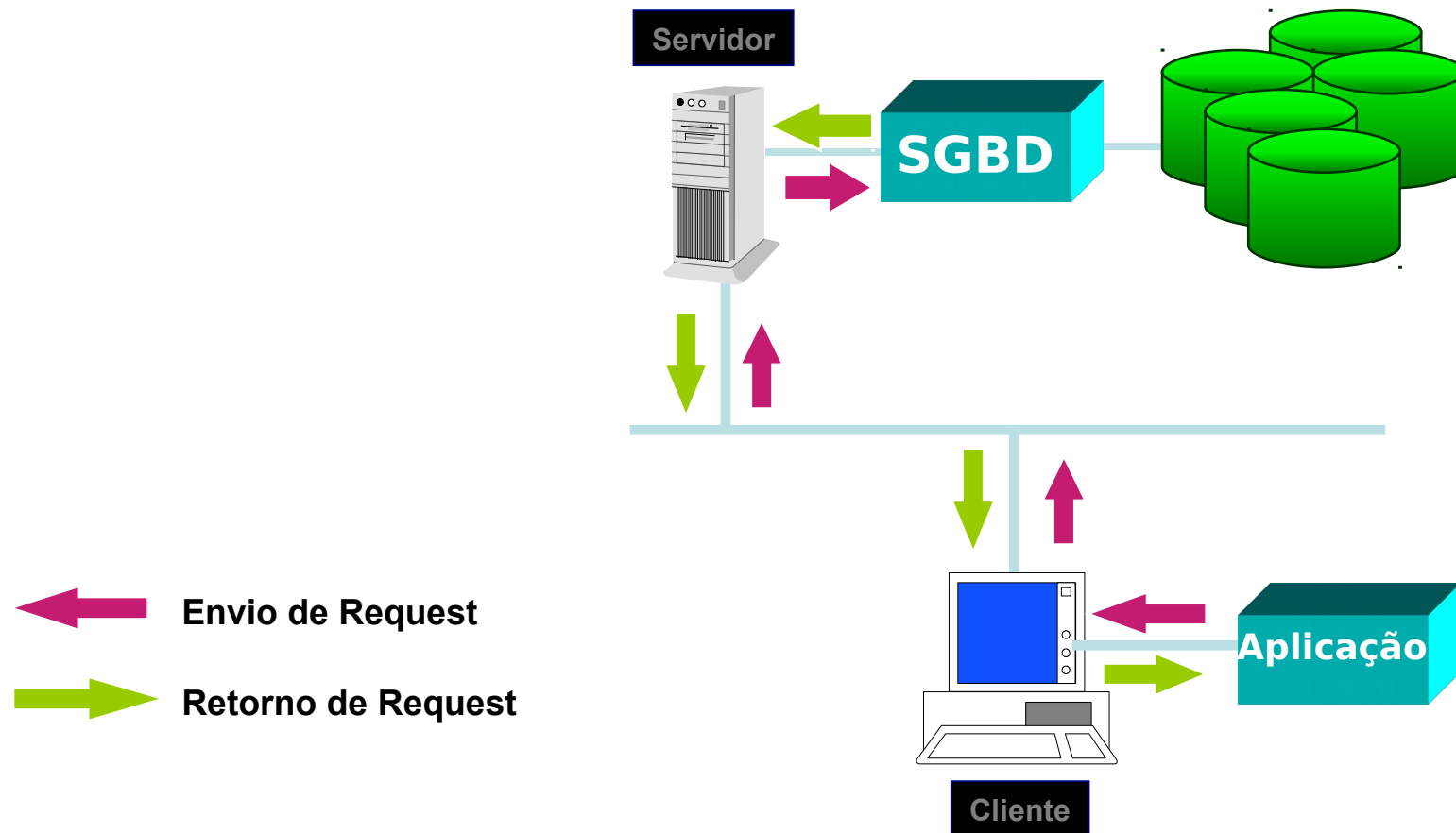
# Projeto de Bancos de Dados

## Segurança no Banco de Dados



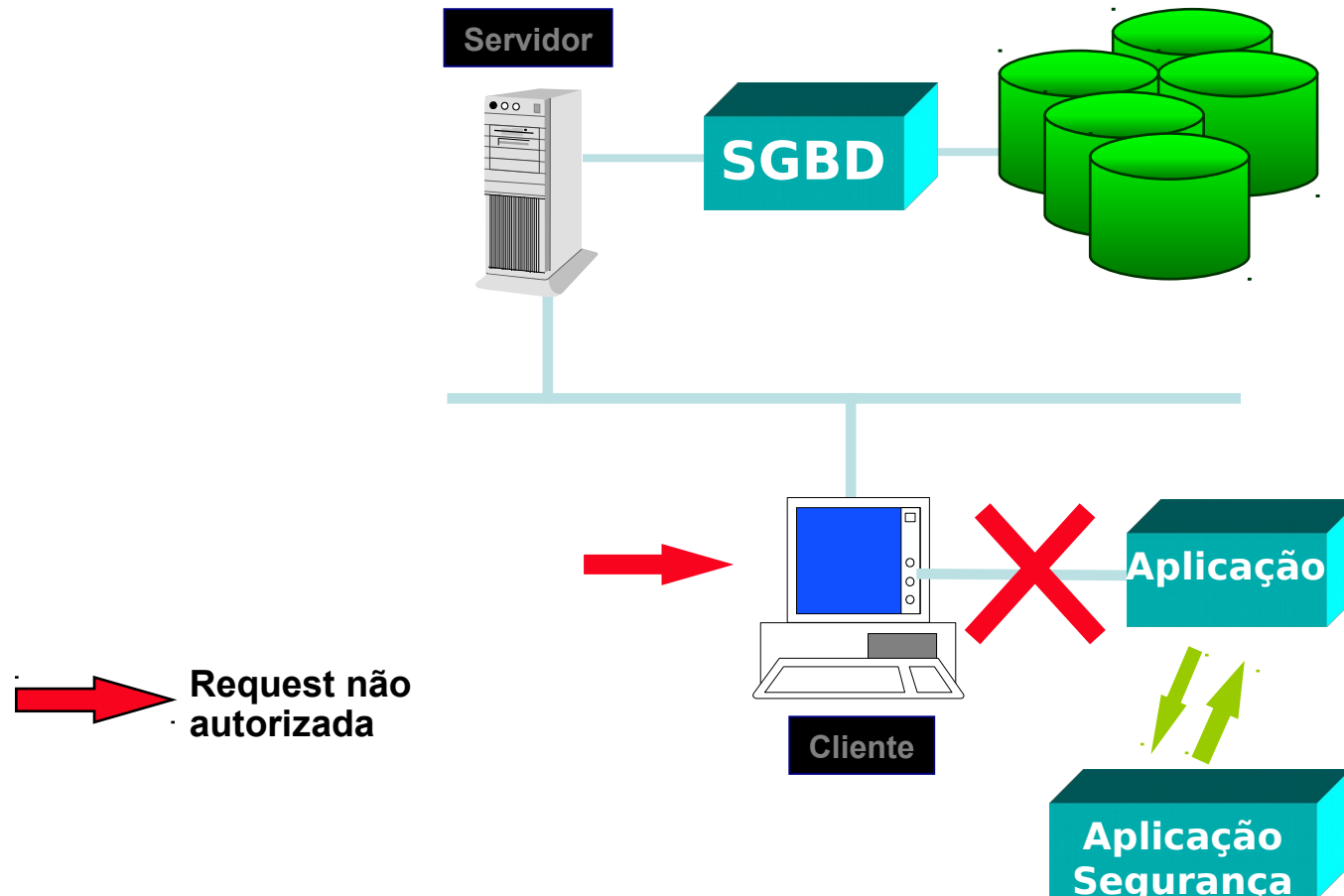
# Projeto de Bancos de Dados

## Segurança - Funcionamento



# Projeto de Bancos de Dados

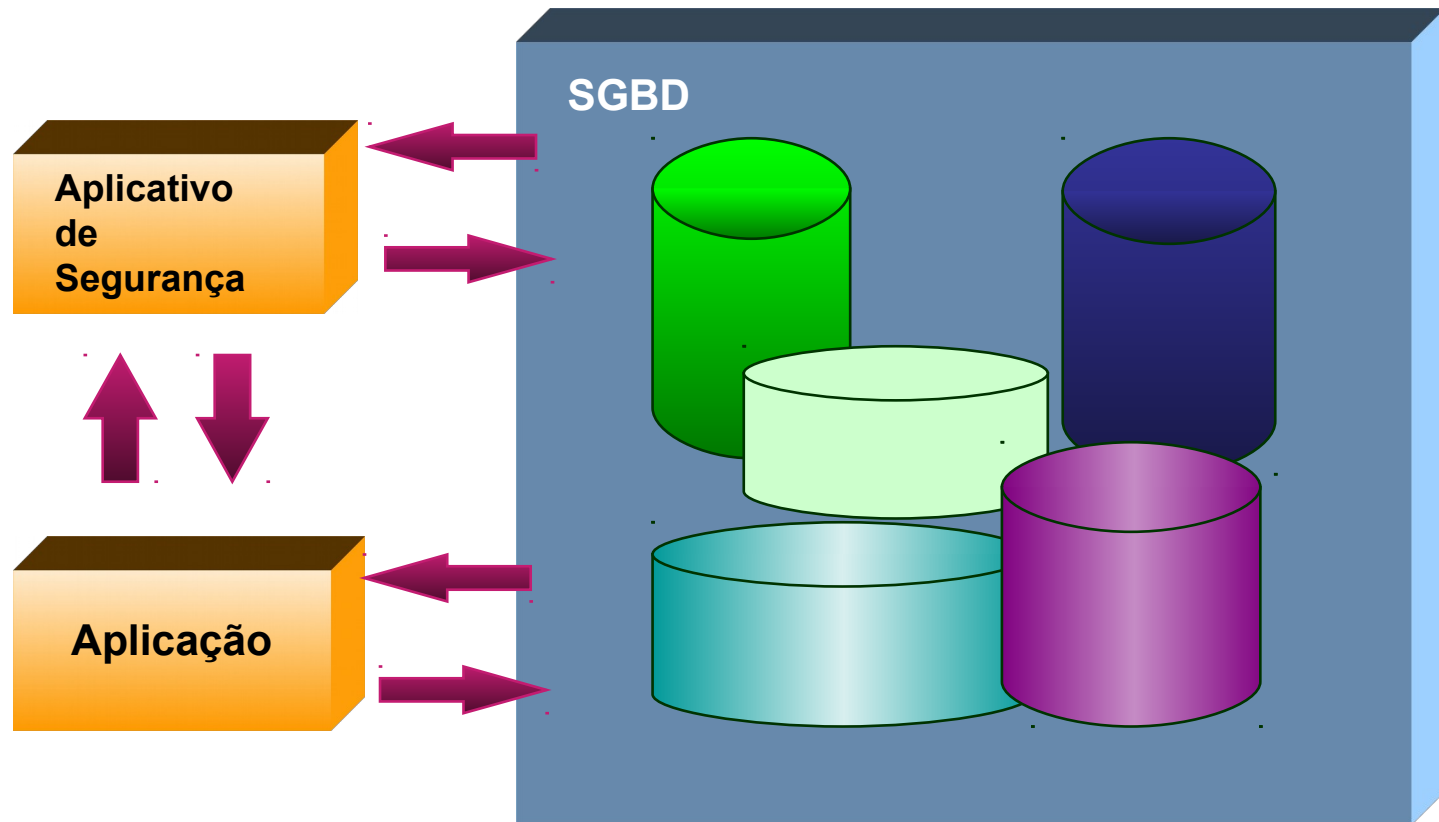
## Segurança em Aplicativos





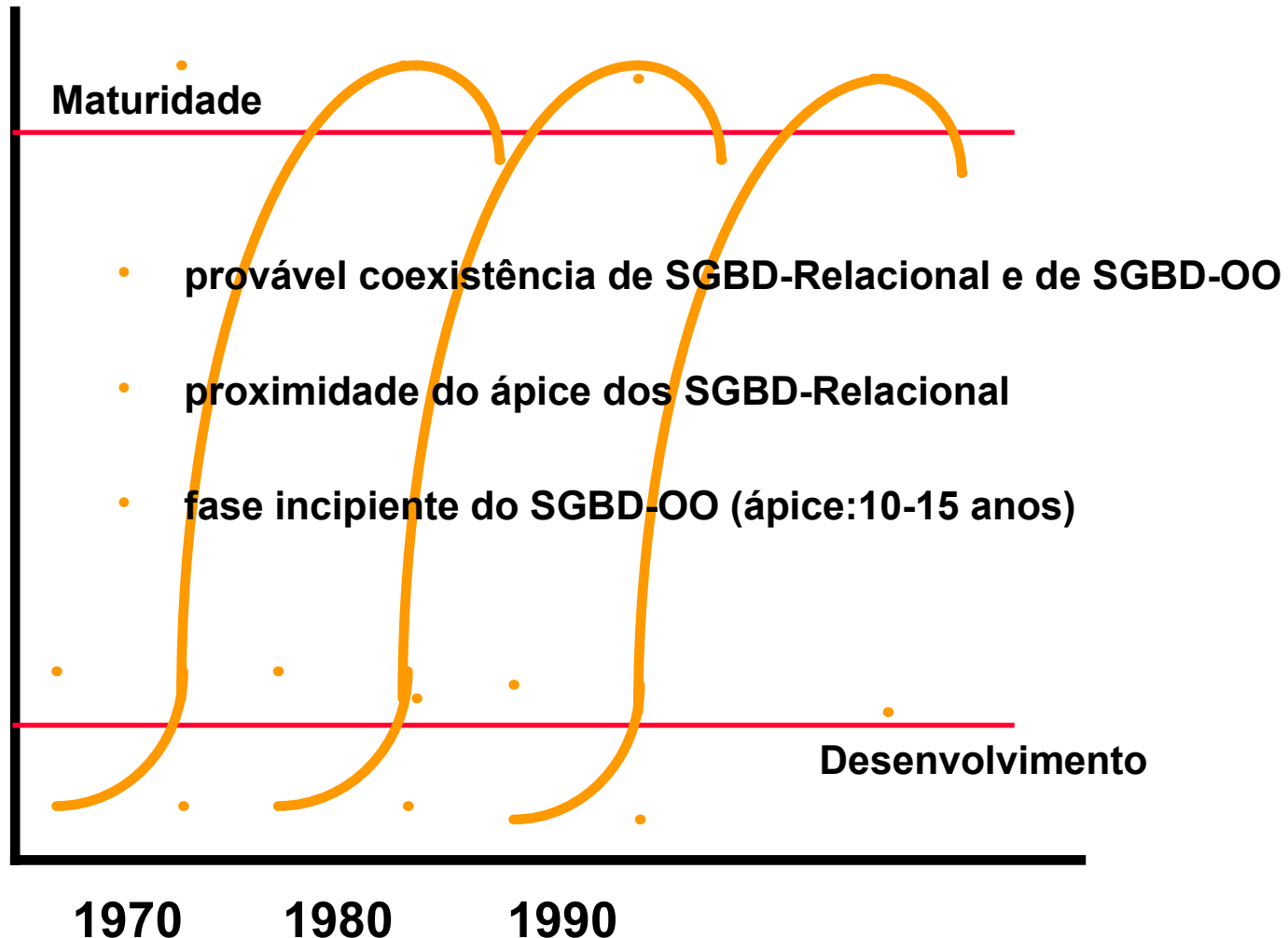
# Projeto de Bancos de Dados

## Segurança em Aplicativos

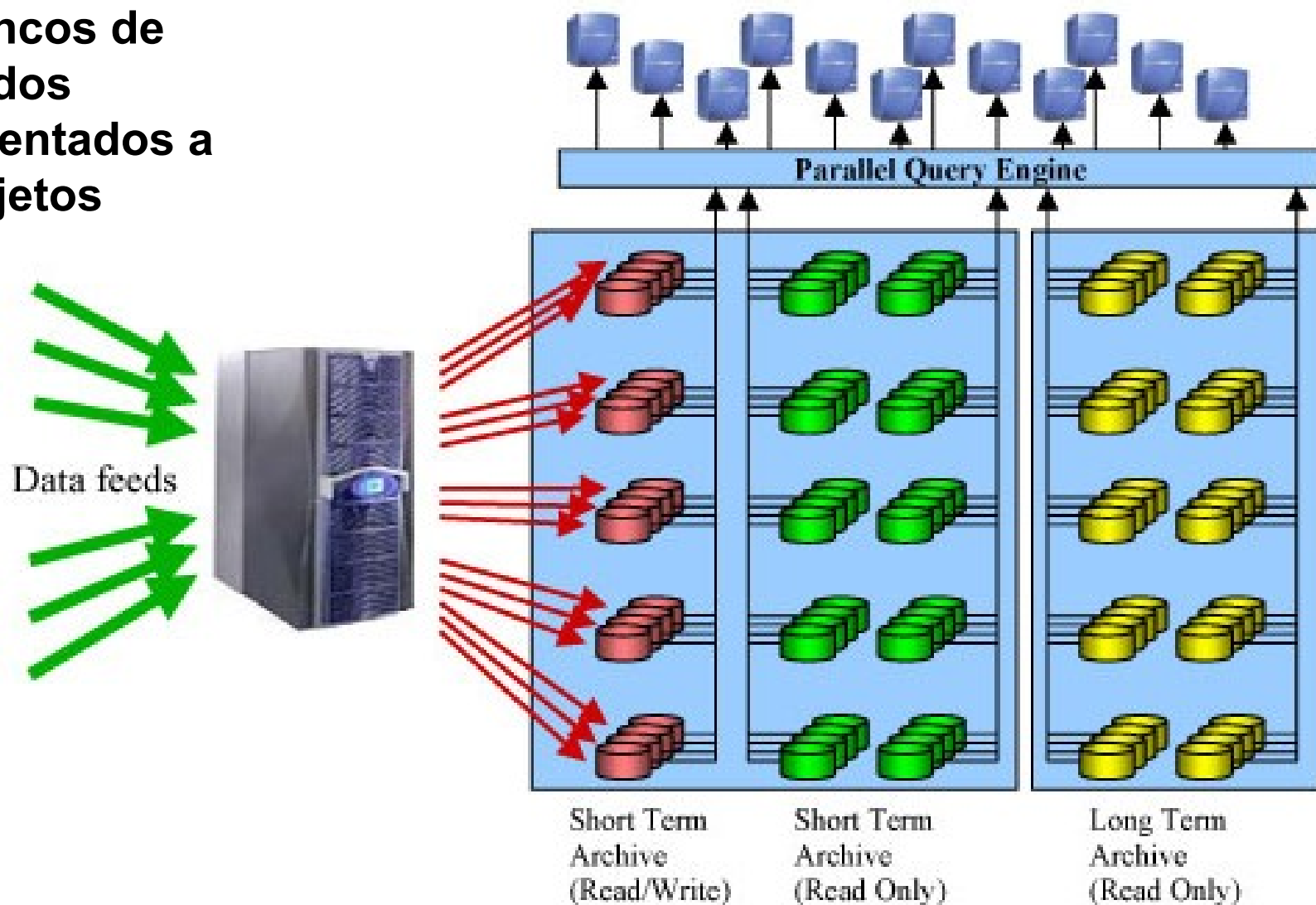


# Projeto de Bancos de Dados

## Bancos de Dados Orientados a Objetos



## Bancos de Dados Orientados a Objetos



# **Projeto de Bancos de Dados**

## **Bancos de Dimensionais**

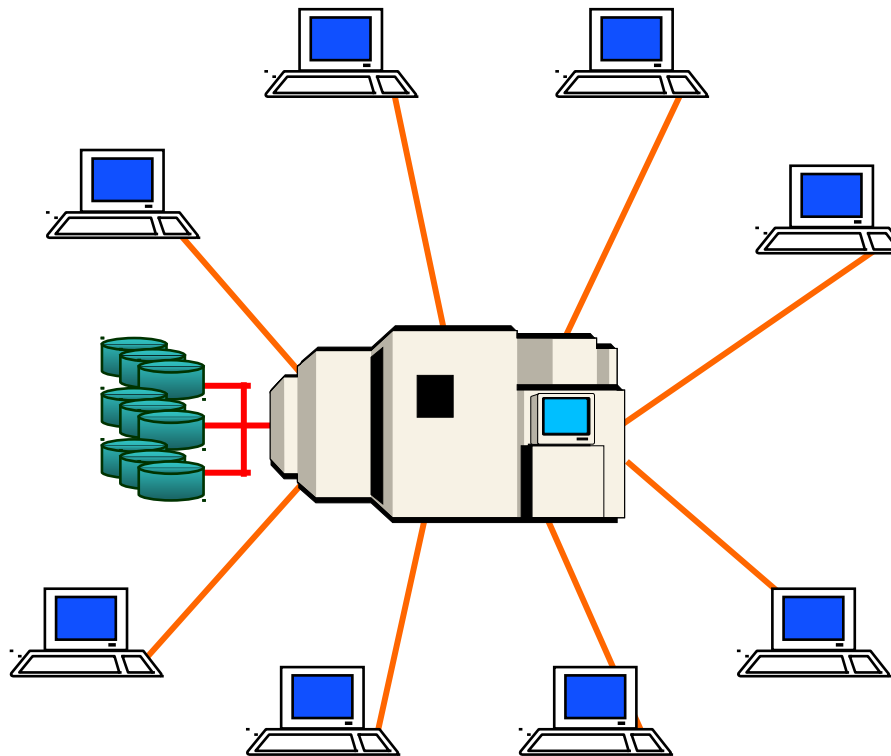
- **Data Warehouse – Estruturas relacionais simples**
- **Gerenciador Dimensional - Cubos**
- **Conjugado com SGBD's Relacionais**

# Conceitos de Bancos de Dados Distribuídos

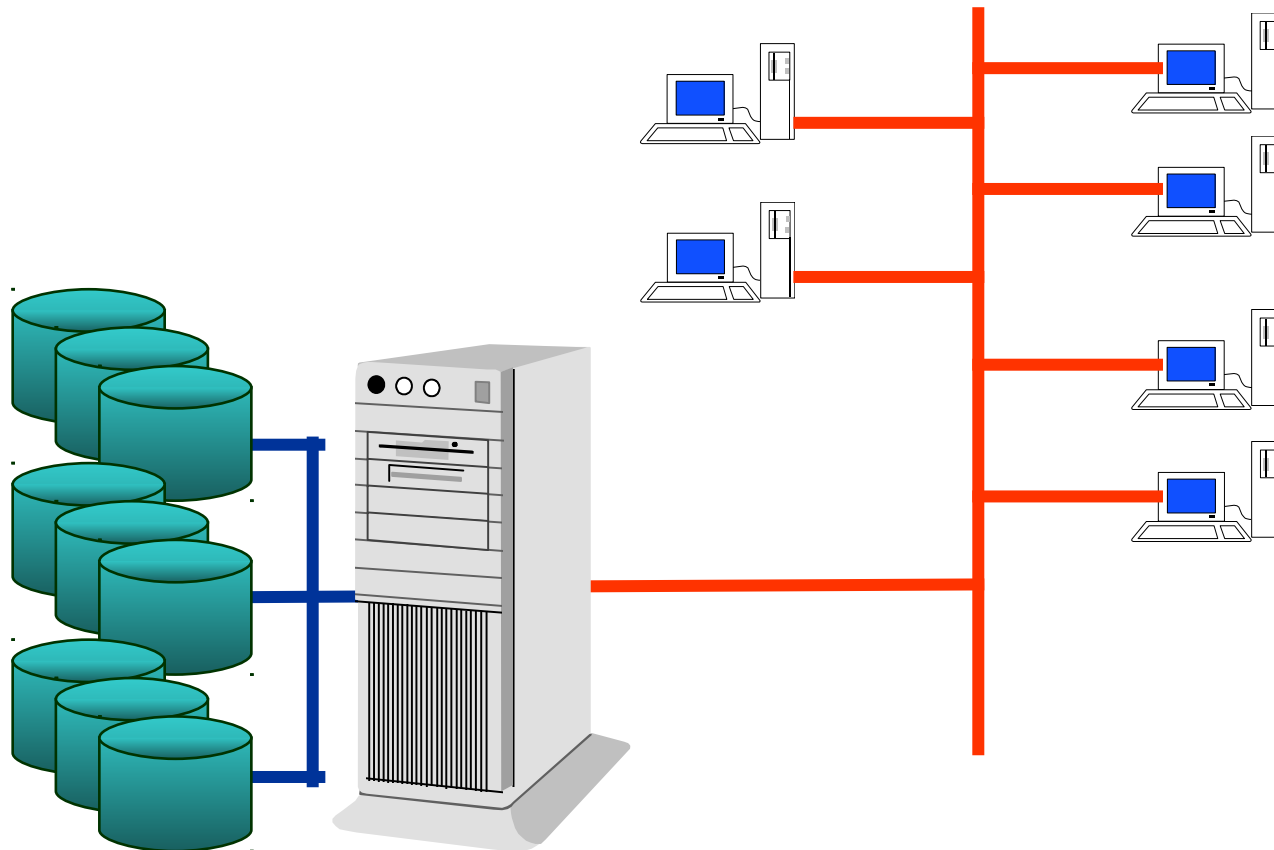
- Centralização de Dados
  - Em Mainframes
  - Em Redes Locais
- Bancos de Dados Distribuídos
- Integração de Bancos de Dados Distribuídos
- Conexão Entre Bancos de Dados Distribuídos

# Conceitos de Bancos de Dados Distribuídos

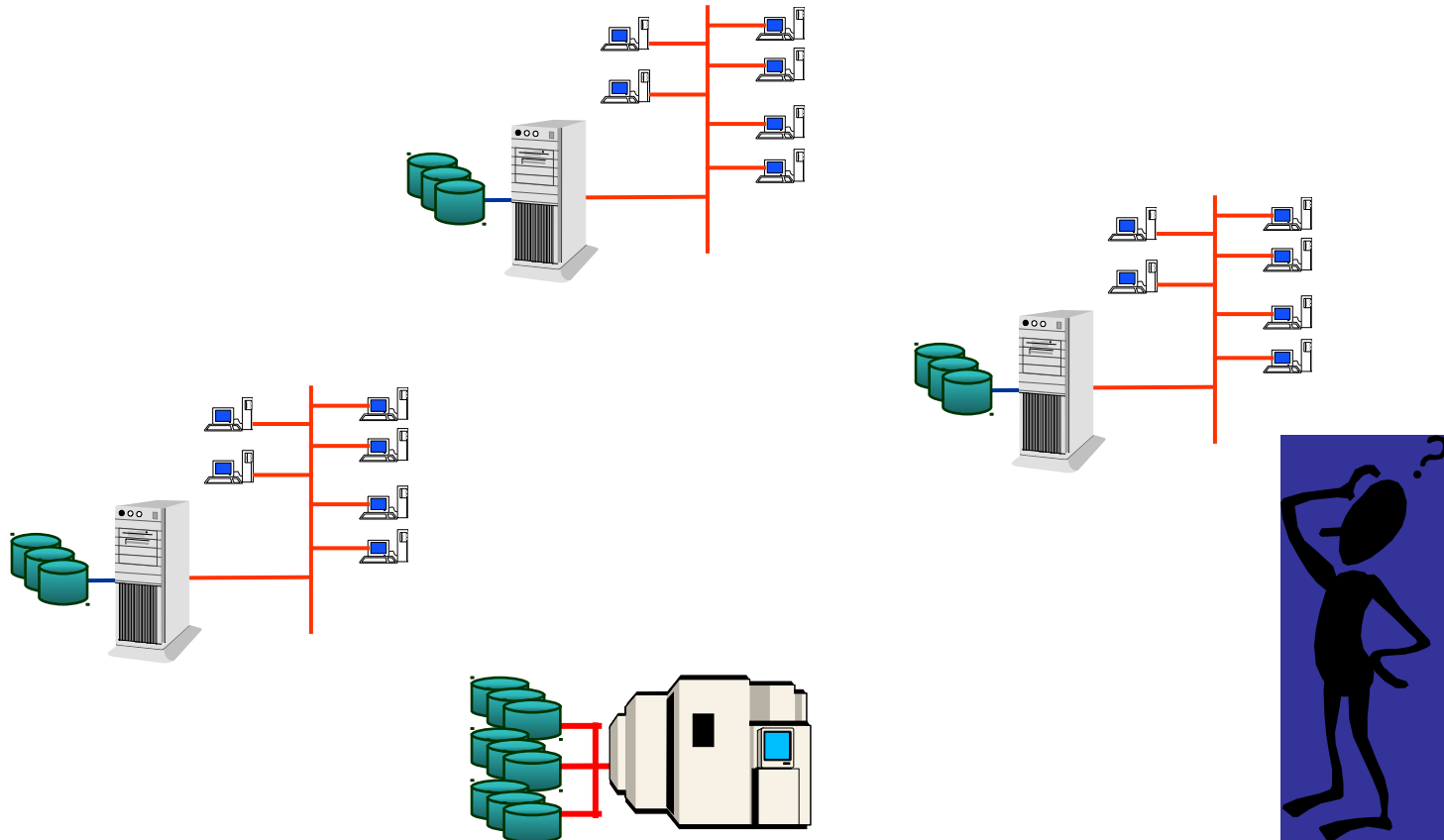
## Centralização de Dados - Mainframe



## Centralização de Dados - Redes Locais

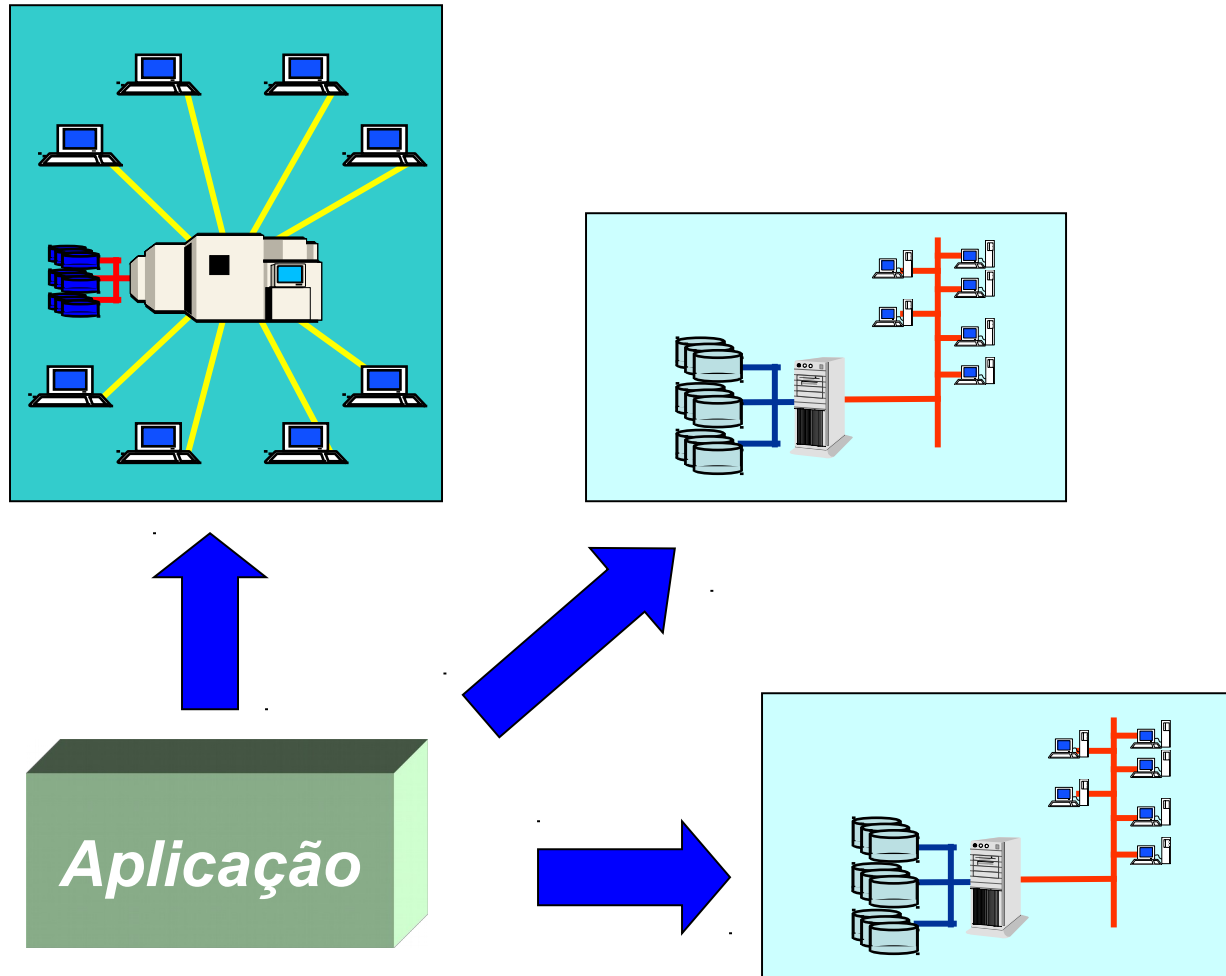


# Integração entre Bancos de Dados Distribuídos

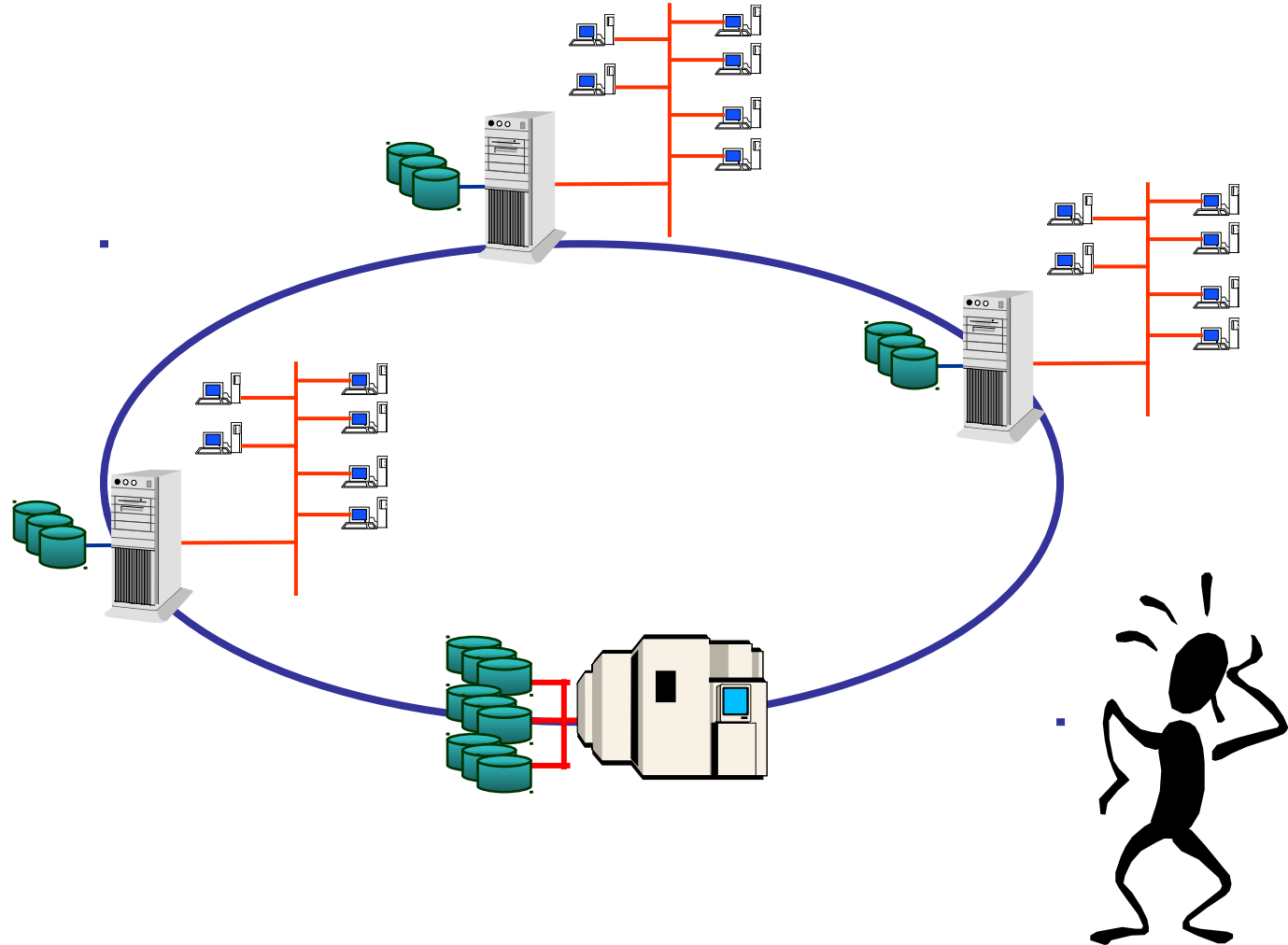




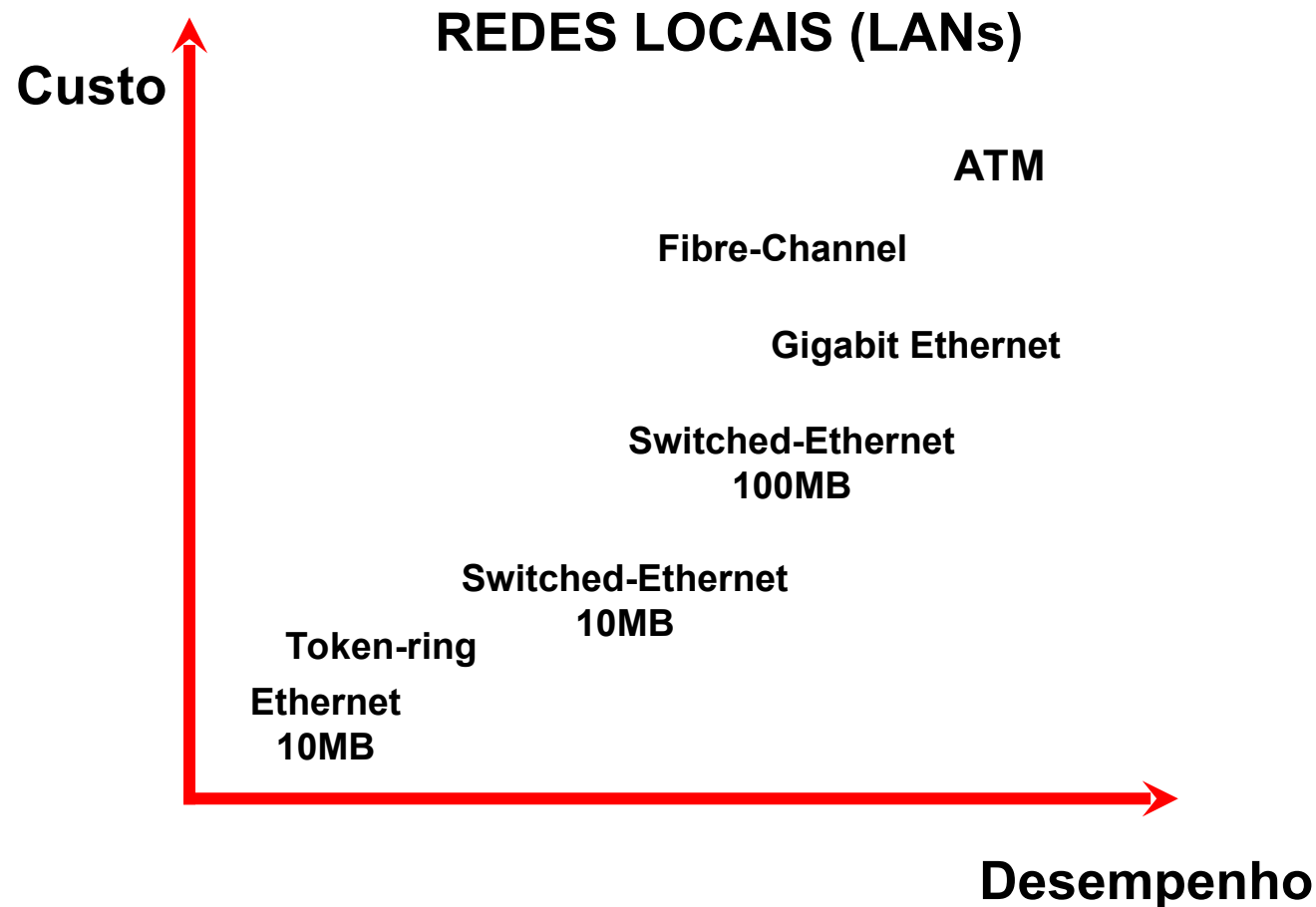
## Integração entre Bancos de Dados Distribuídos



# Integração entre Bancos de Dados Distribuídos

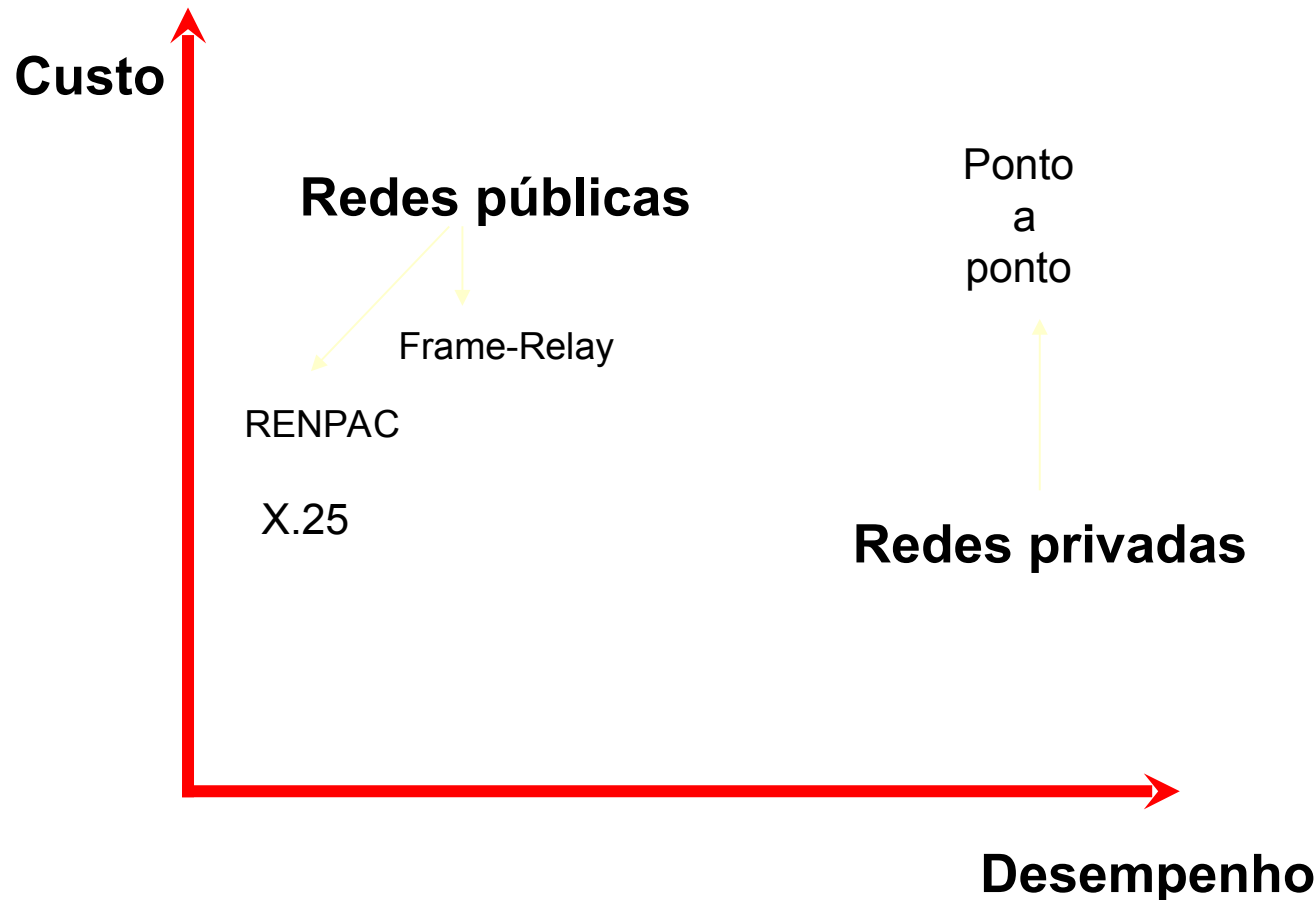


# Infra-estrutura de Conexão



# Infra-estrutura de Conexão

## REDES GEOGRÁFICAS (WANs)

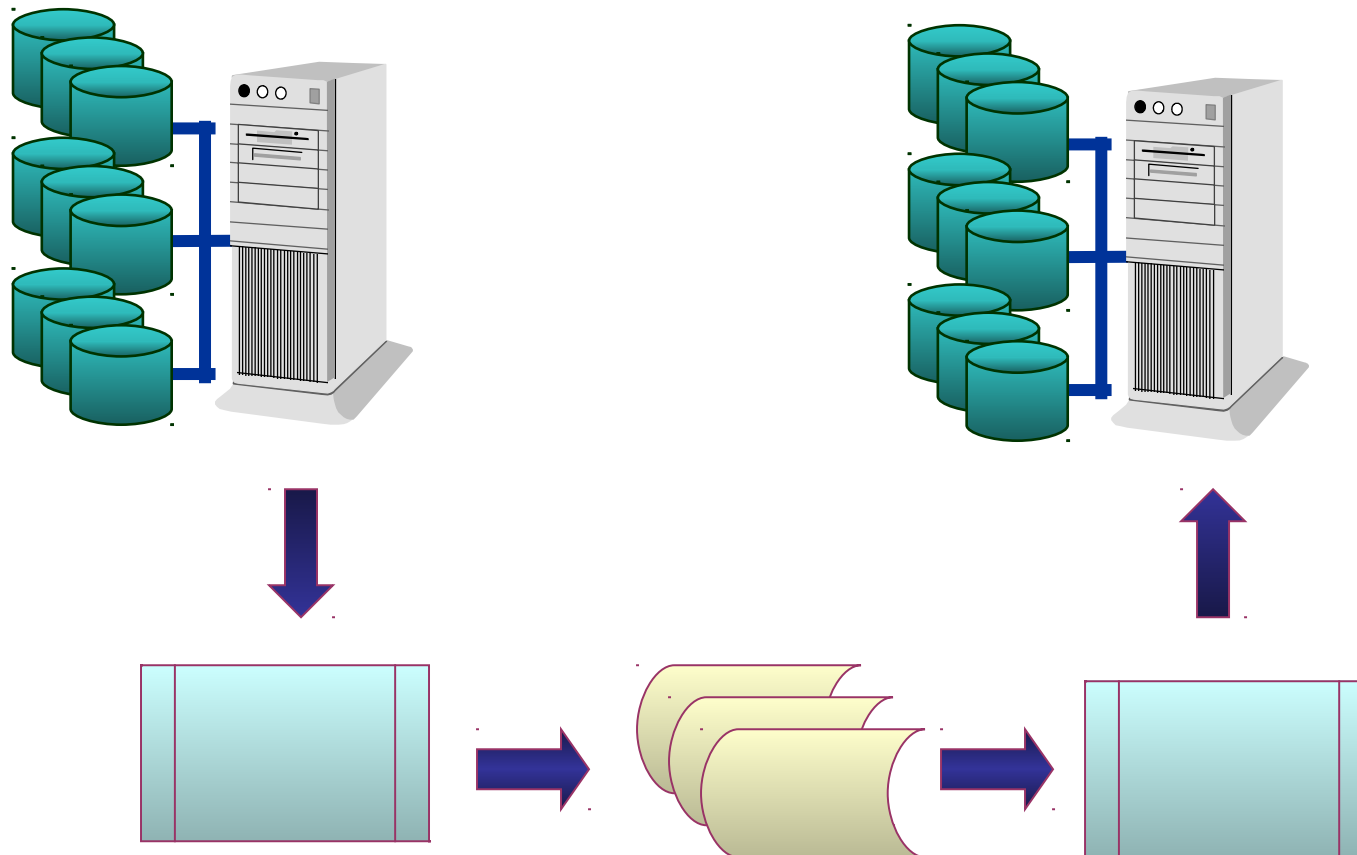


# Distribuição de Dados em SGBD's

- Transferências de Dados
  - Exportação / Importação
  - Backups
- Replicação
  - Publicações, Artigos e Assinantes
  - Processos Síncronos e Assíncronos
  - Replicação Uni-Direcional
    - Síncrona x Assíncrona: Necessidades de Consistência
  - Replicação Bi-Direcional
    - Assíncrona : Colisão de Dados
    - Síncrona : Two Phase Commit
- Fragmentação
  - Particionamento
  - Consistência de Dados

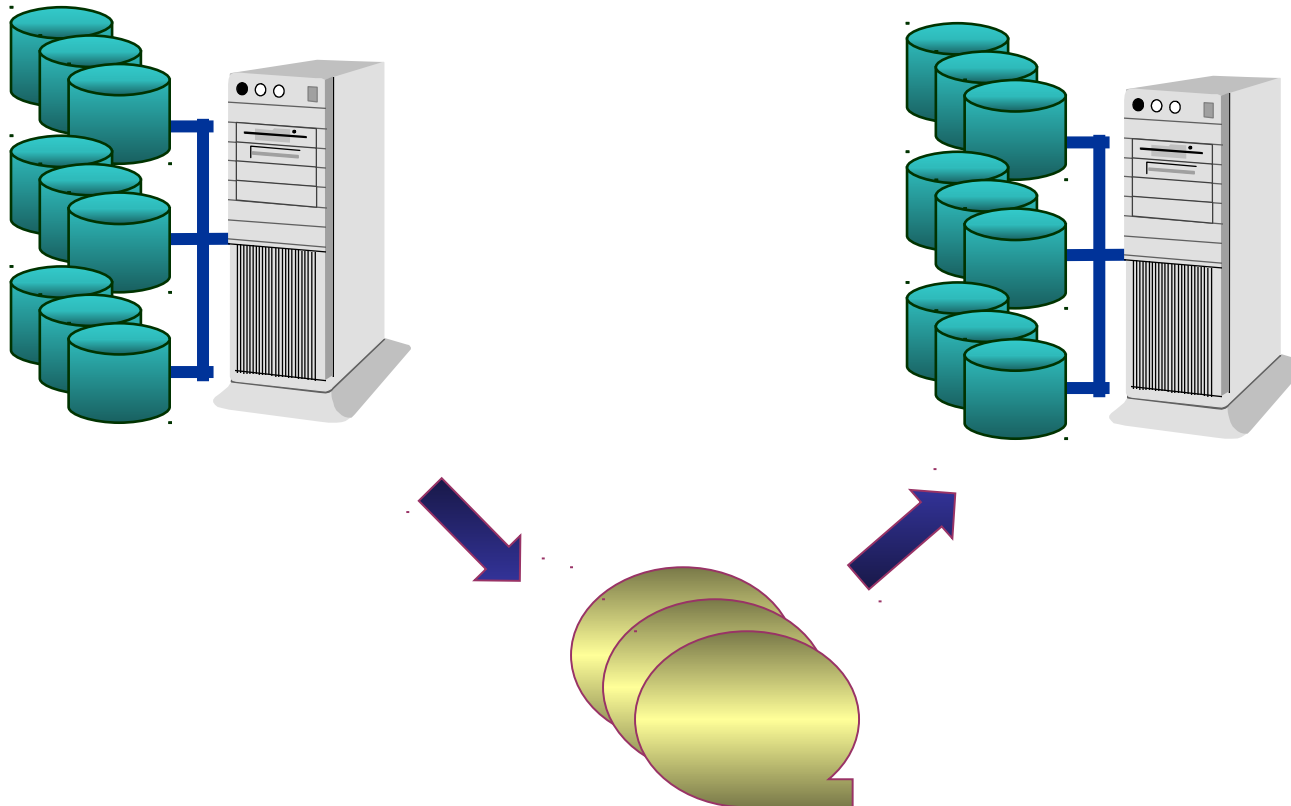
# Transferência de Dados

## Exportação / Importação



# Transferência de Dados

## Backups



# Replicação

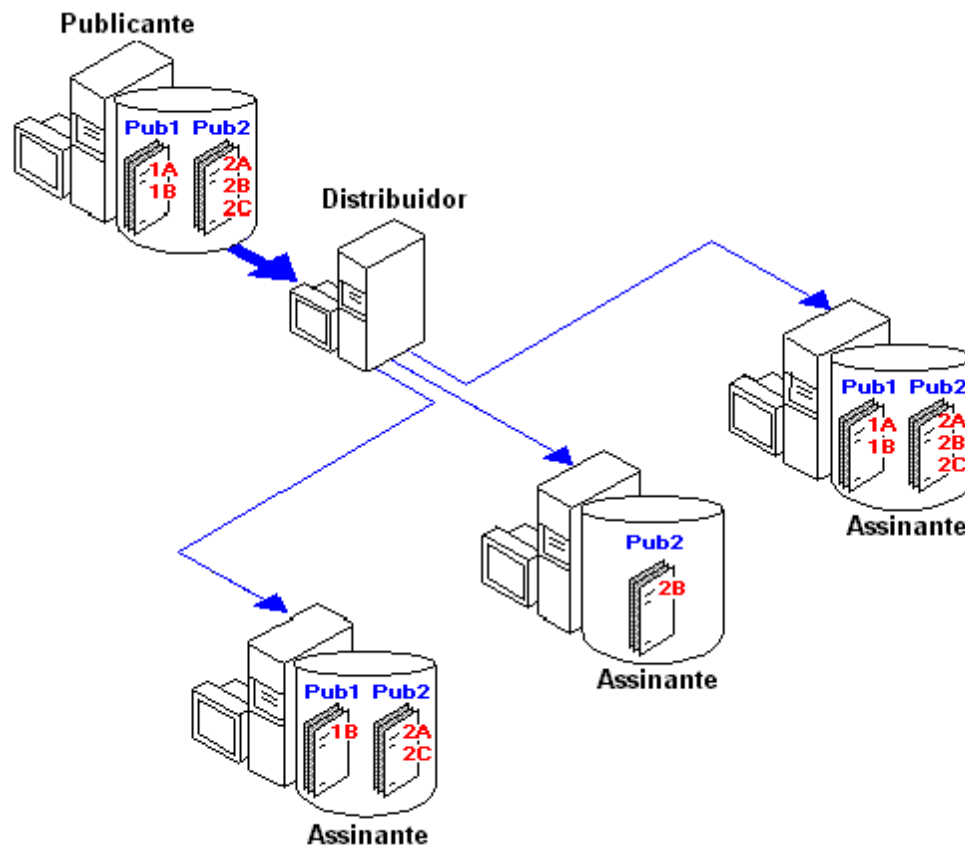
## Publicações, Artigos e Assinaturas

- Publicação
  - Conjunto de um ou mais artigos. Representam as informações disponibilizadas por um servidor para replicação.
- Artigo
  - Unidade lógica da replicação de dados.  
Representa um conjunto lógico de informações.
- Assinante
  - Denominação dada a quem recebe a informação disponibilizada pelo publicante. Destino da réplica de dados.



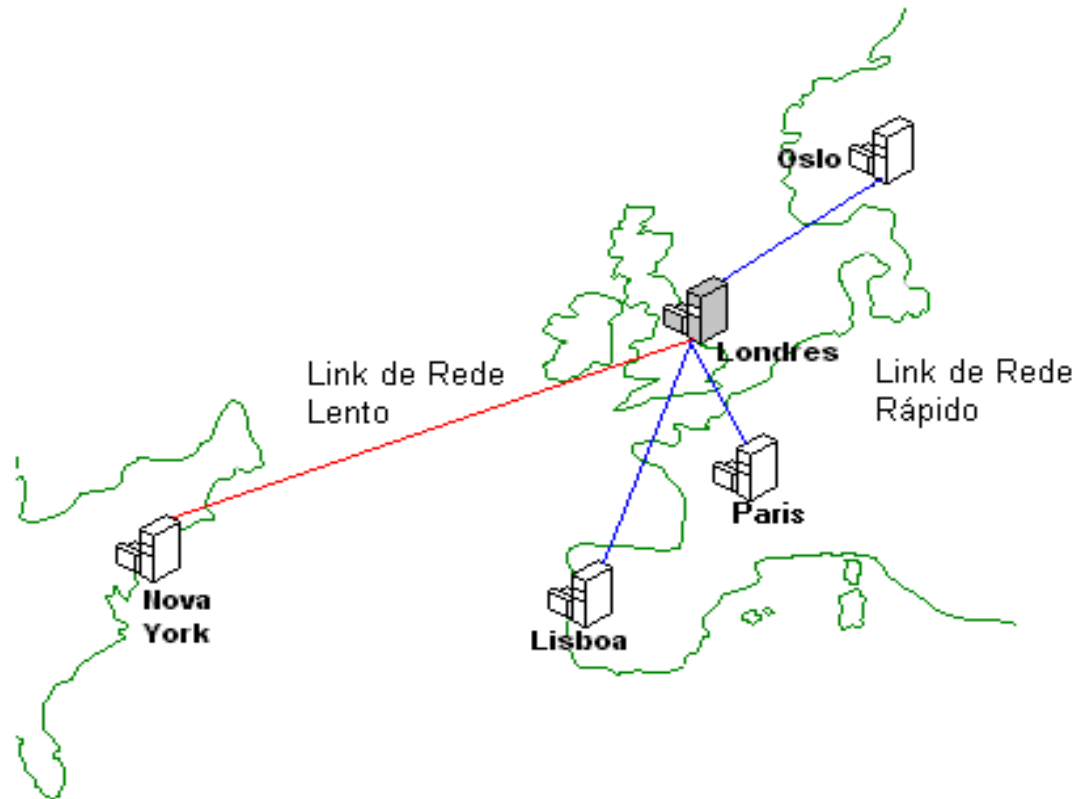
# Replicação

## Publicações, Artigos e Assinaturas



# Replicação

## Publicações, Artigos e Assinaturas



# Replicação

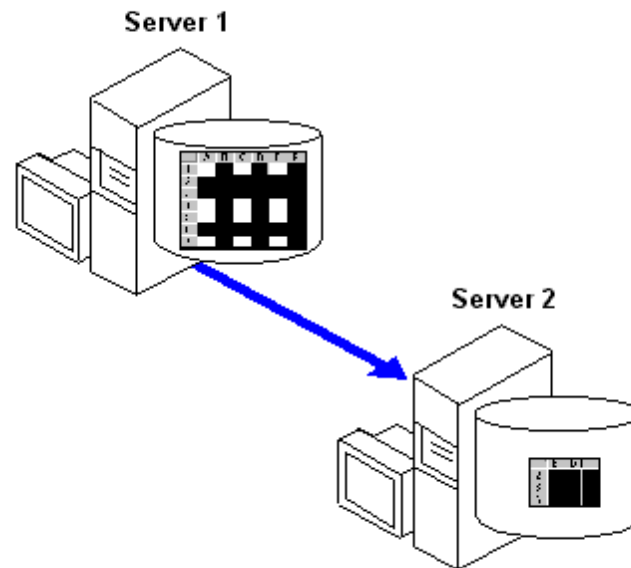
## Publicações, Artigos e Assinaturas

- Particionamento de Replicação
  - Vertical : Colunas
  - Horizontal : Linhas

# Replicação

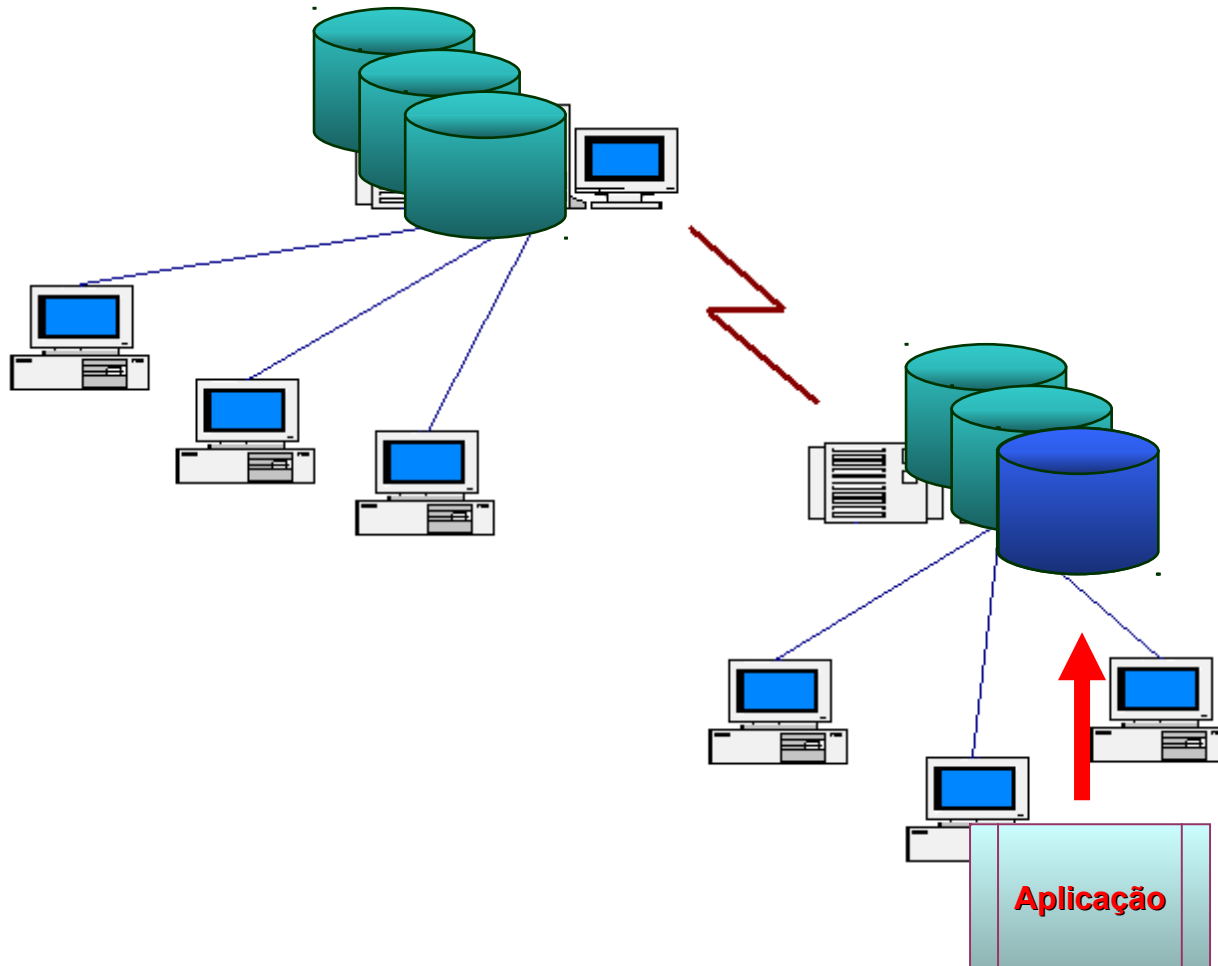
## Publicações, Artigos e Assinaturas

### Replicação Fragmentada



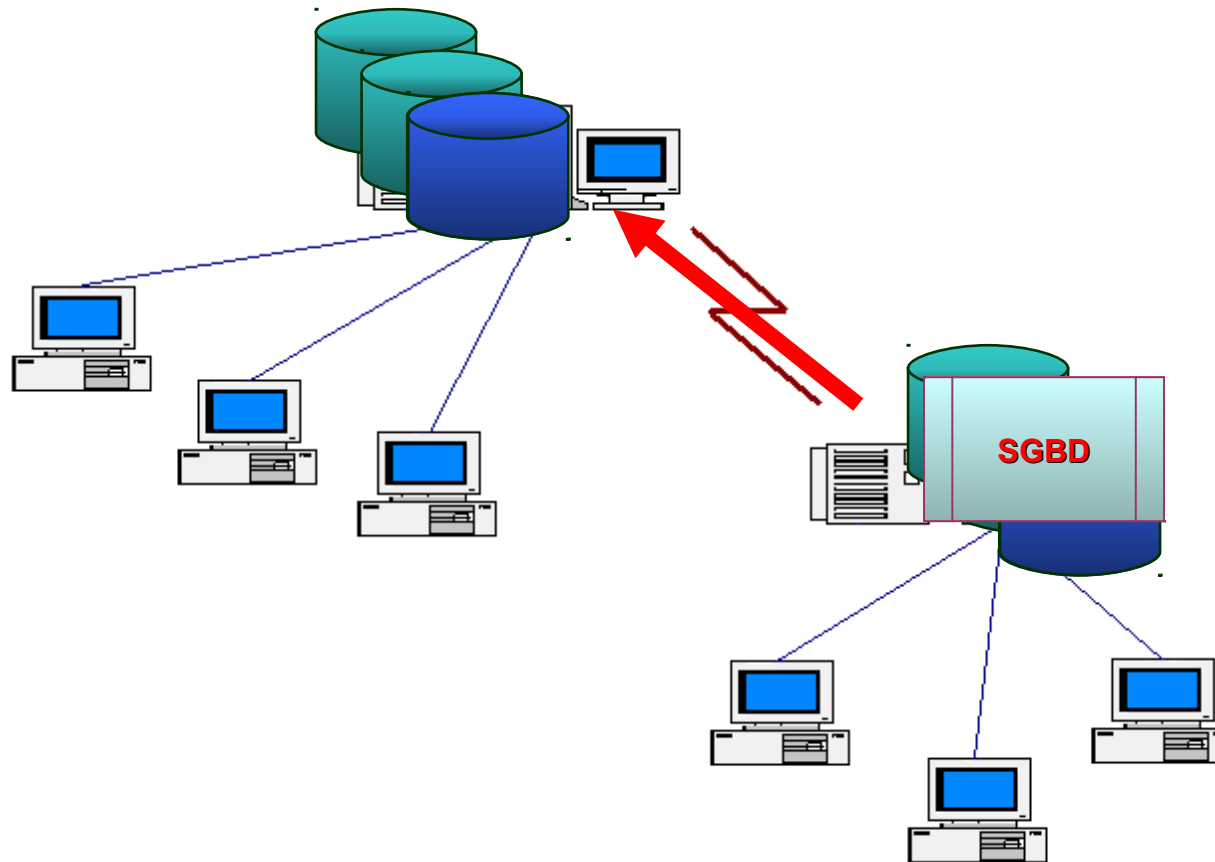
# Replicação

## Processos Assíncronos



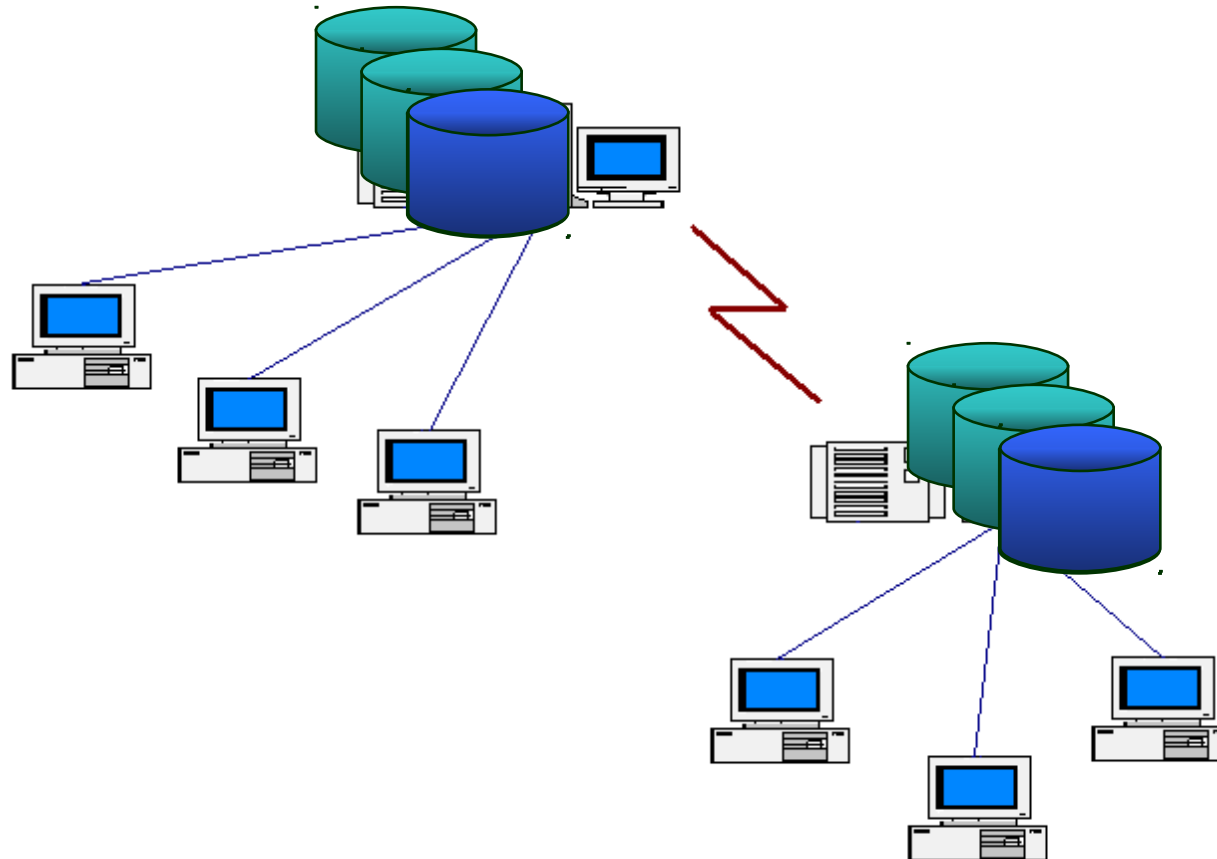
# Replicação

## Processos Assíncronos



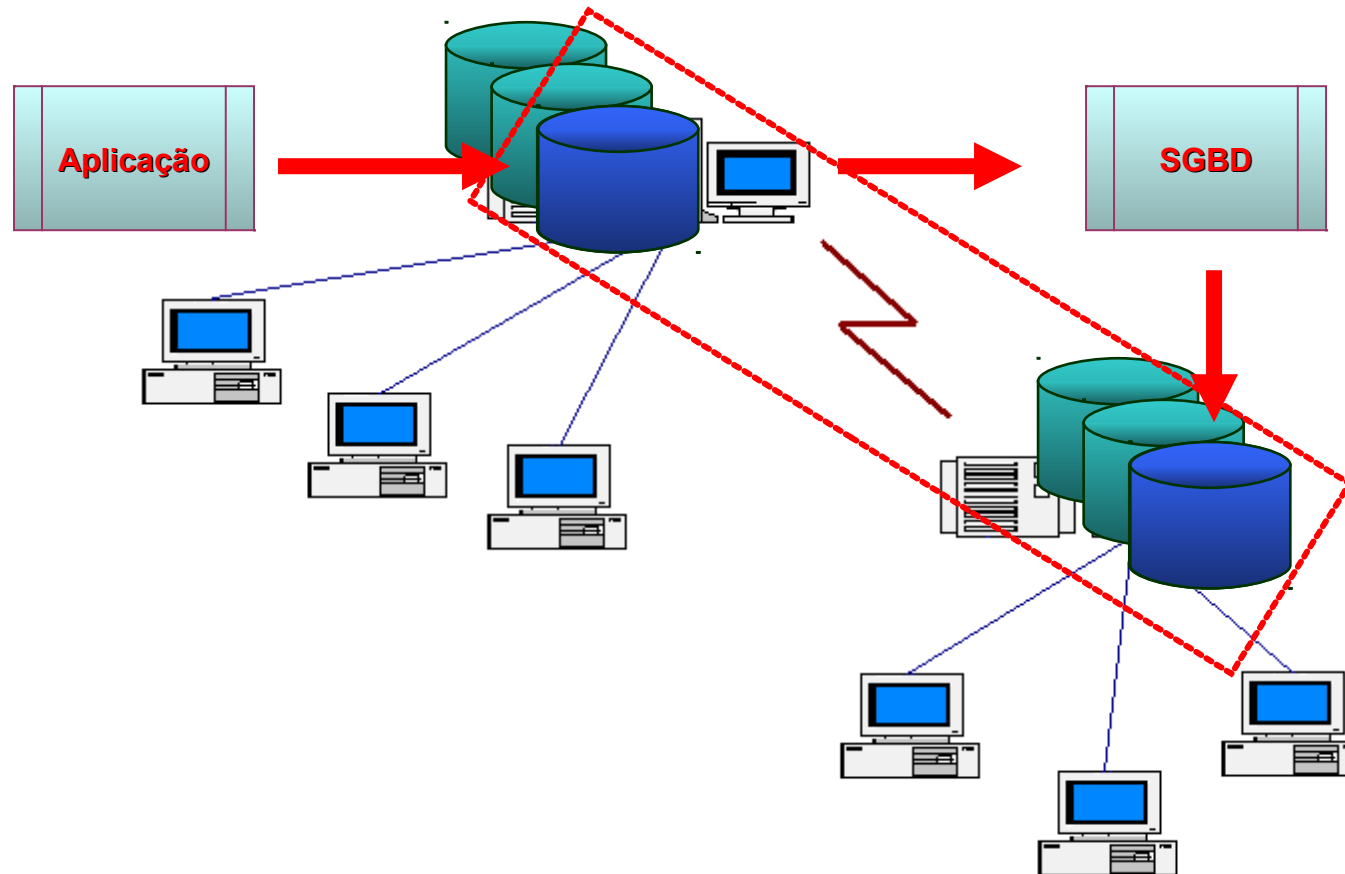
# Replicação

## Processos Assíncronos



# Replicação

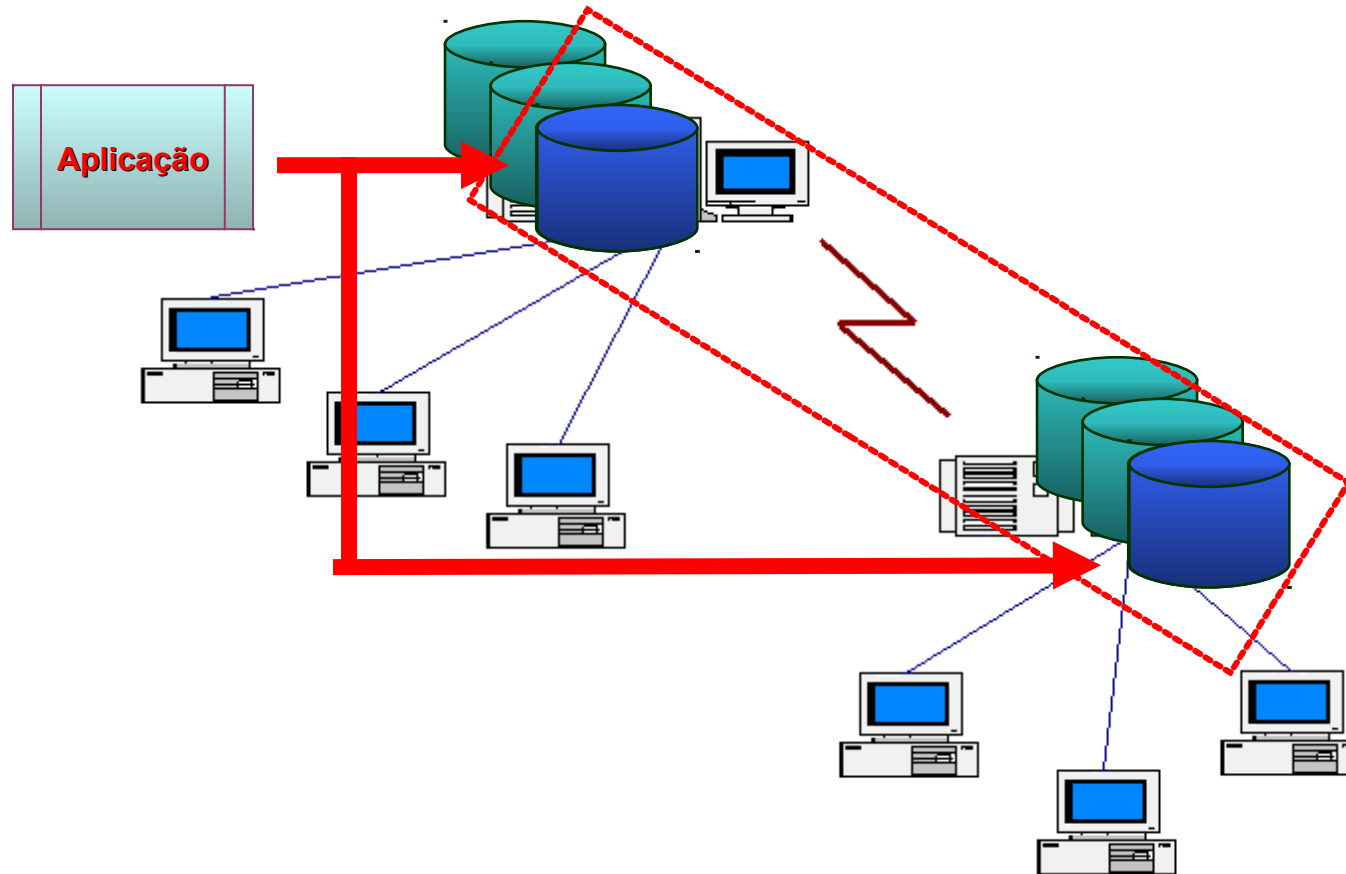
## Processos Síncronos





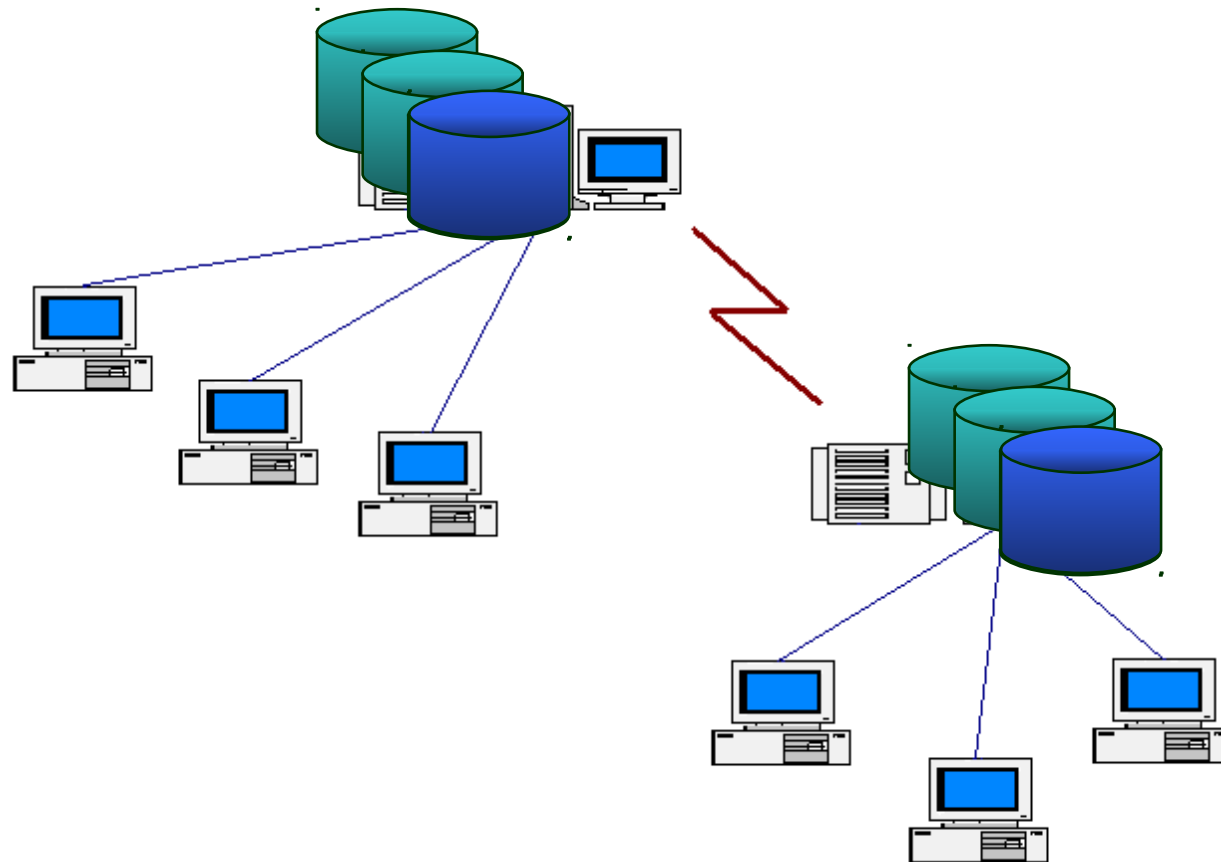
# Replicação

## Processos Síncronos



# Replicação

## Processos Síncronos



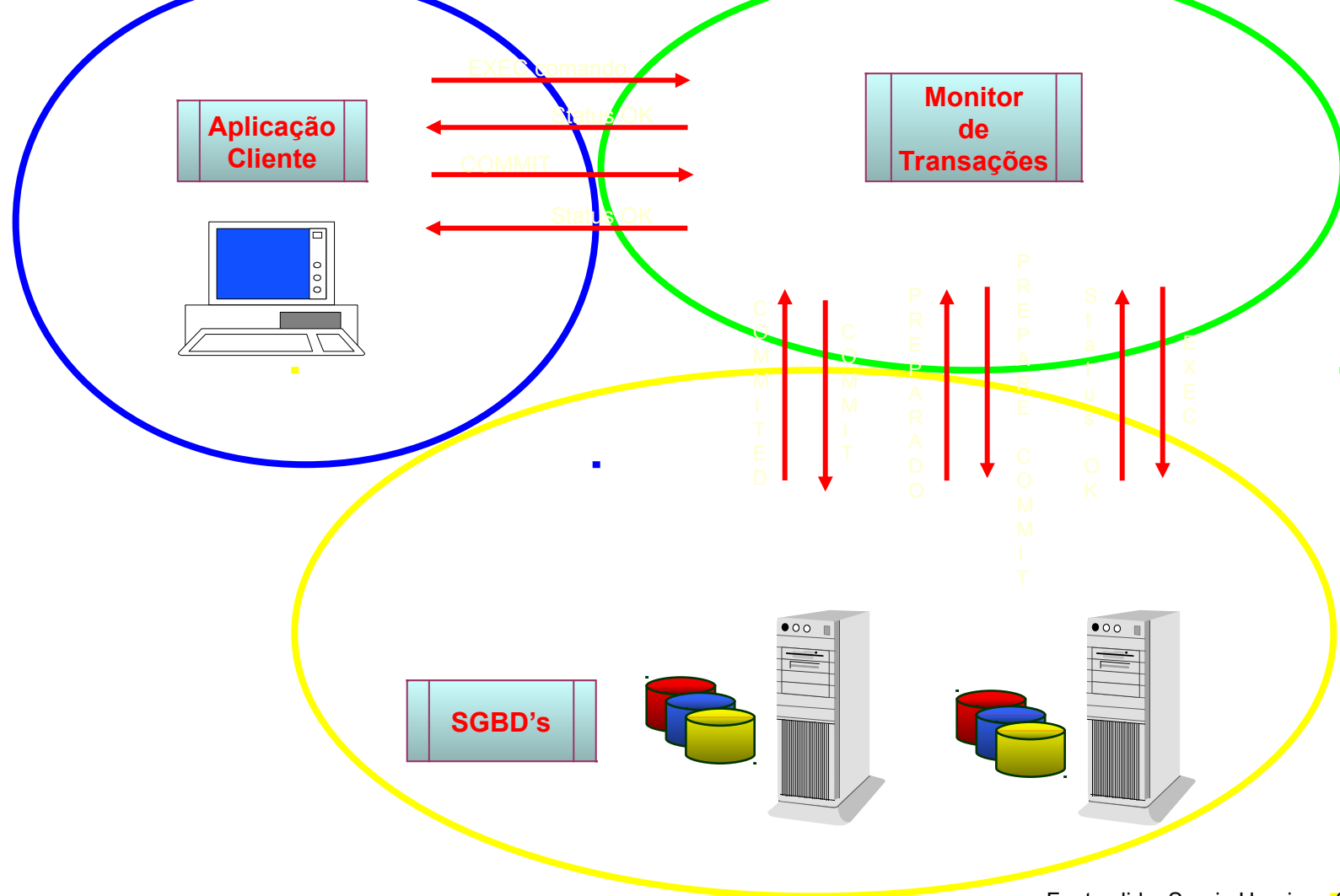
# Replicação

## Processos Síncronos - Two Phase Commit

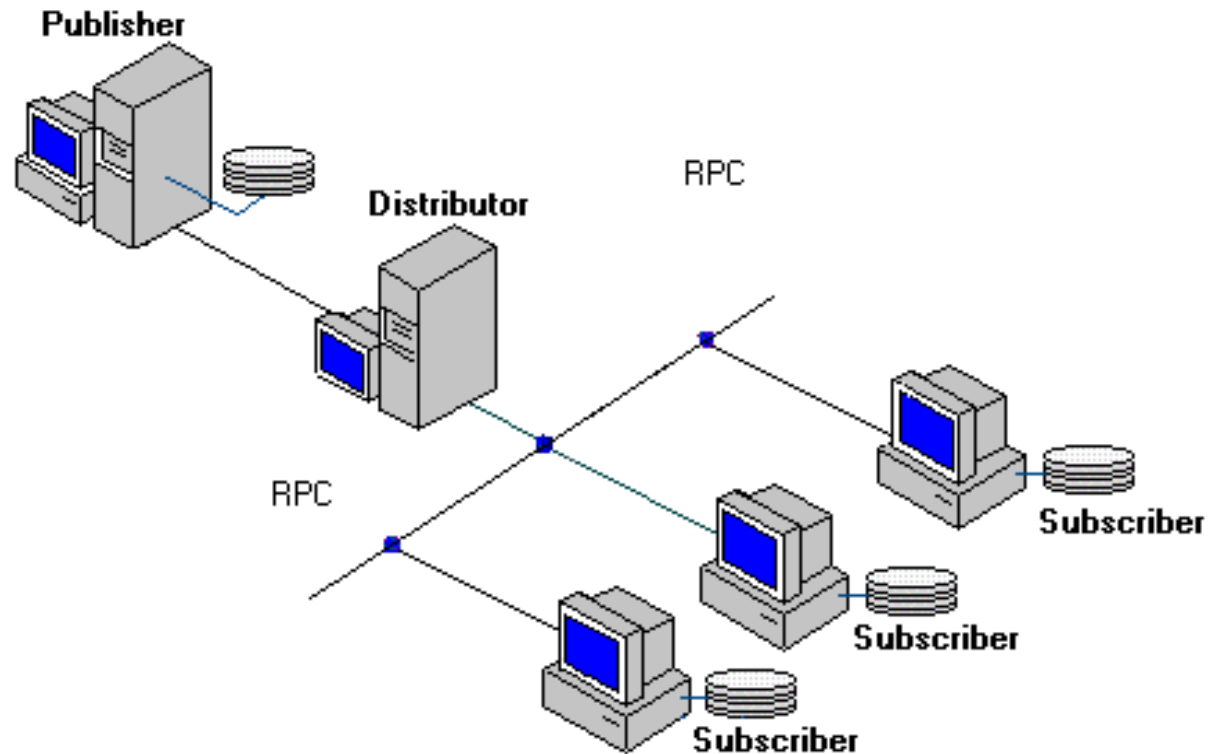
- Controle por Aplicativo
  - Funções de API
- Controle do SGBD
- Monitor de Transações

# Replicação

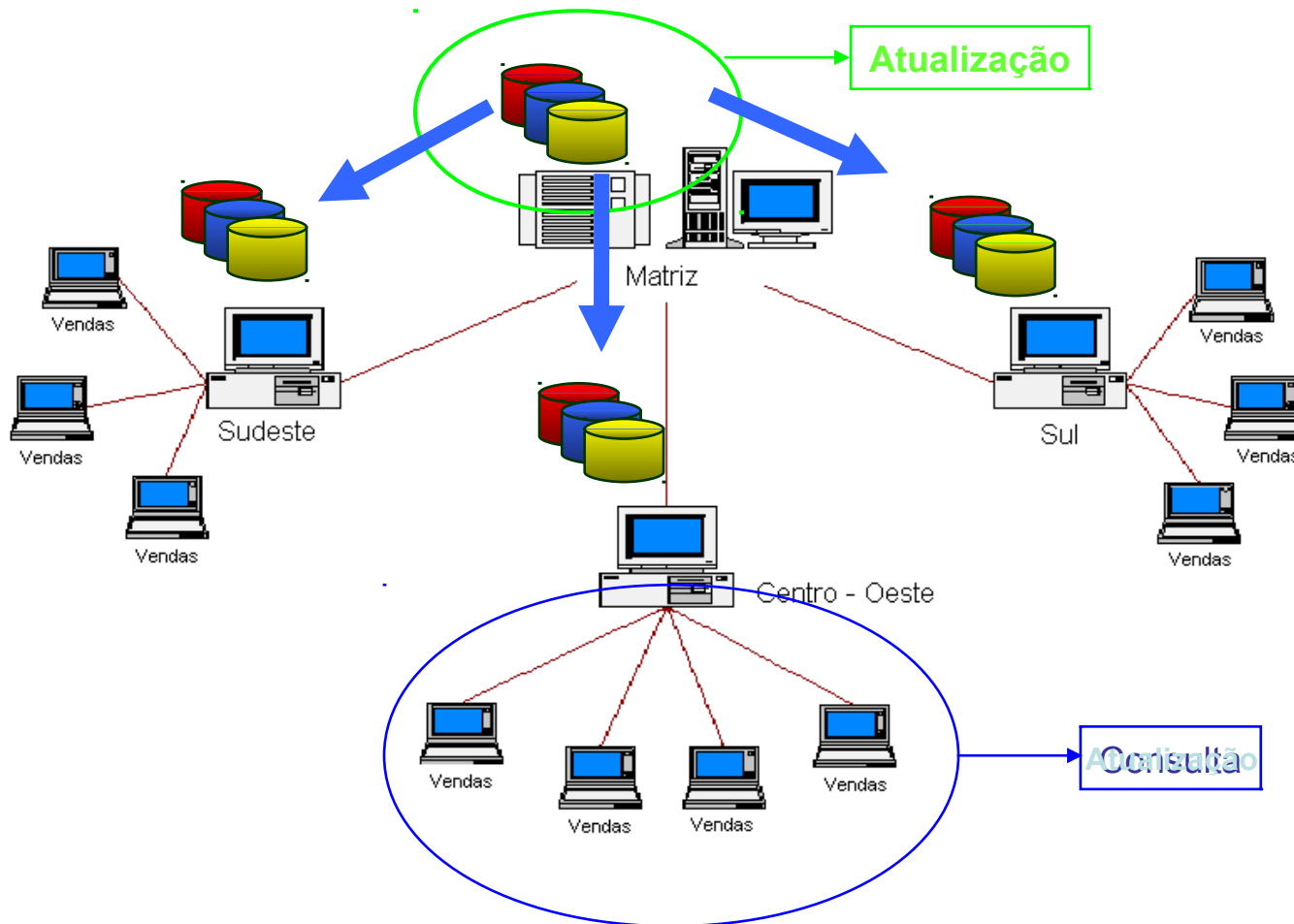
## Processos Síncronos - Two Phase Commit



# Replicação Uni-Direcional



# Replicação Uni-Direcional



# Replicação Uni-Direcional

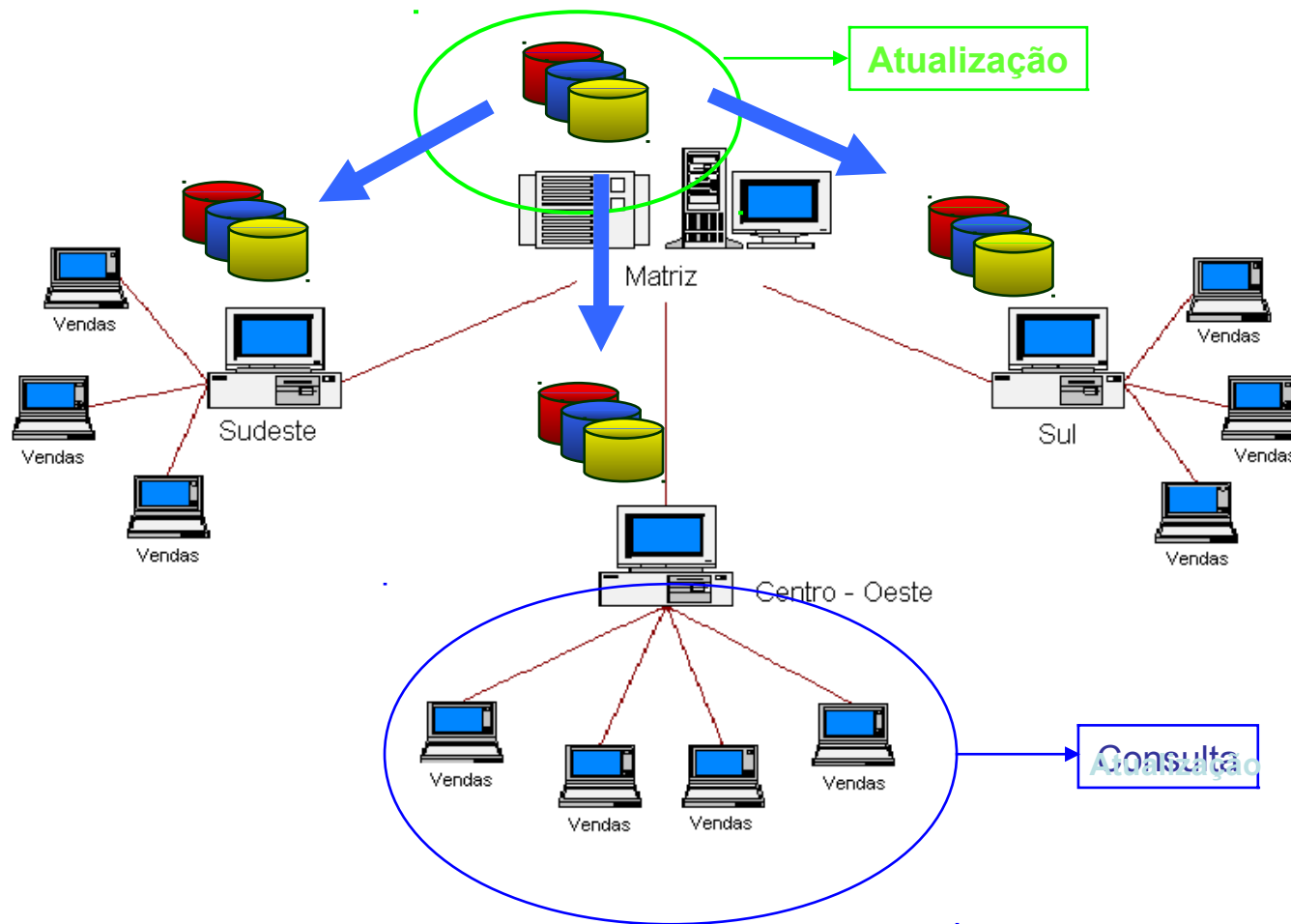
## Síncrona x Assíncrona: Necessidades de Consistência

- Necessidade de dados atualizados em tempo real
- Alta consistência >> Dependência entre servidores

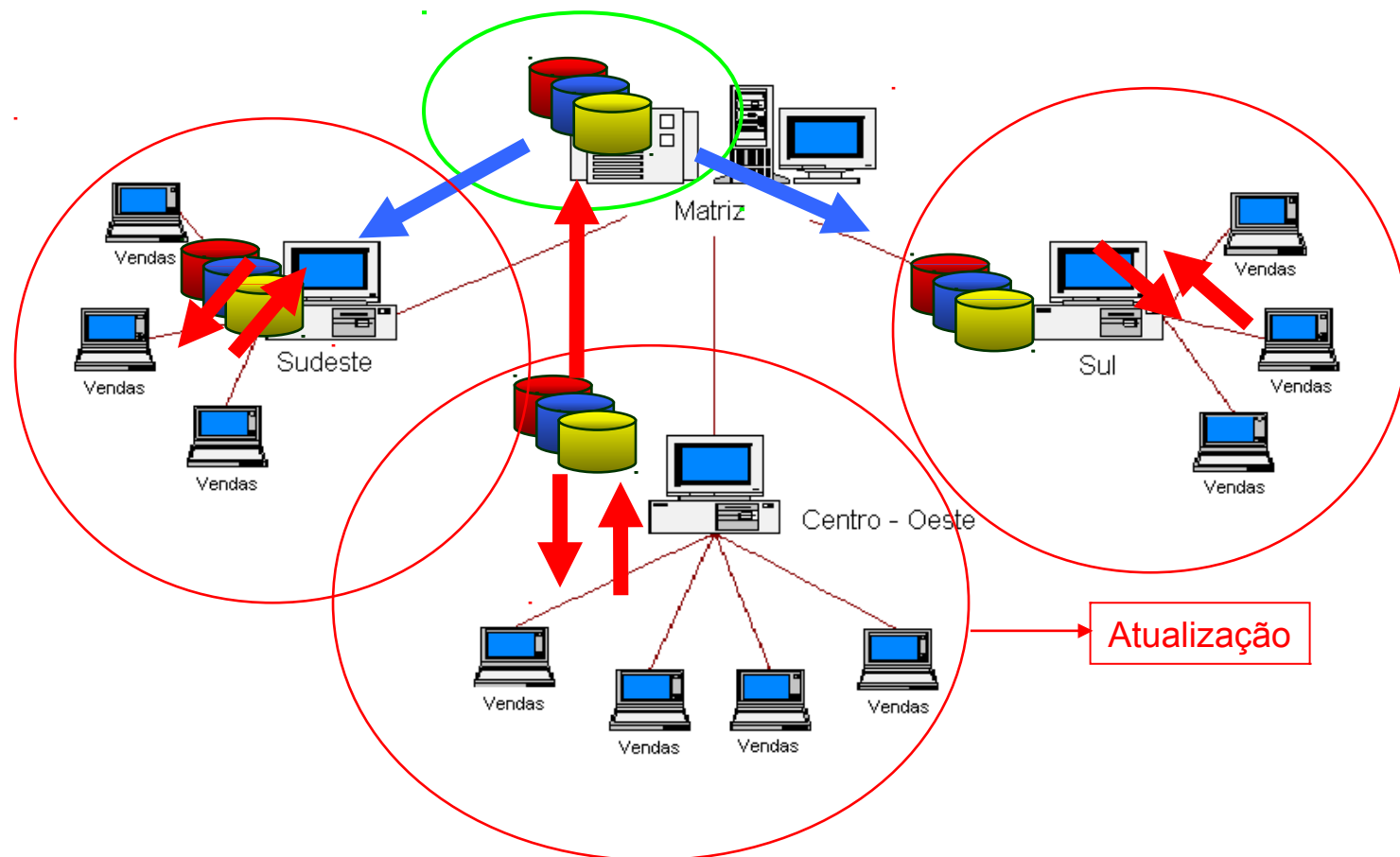




# Replicação Bi-Direcional

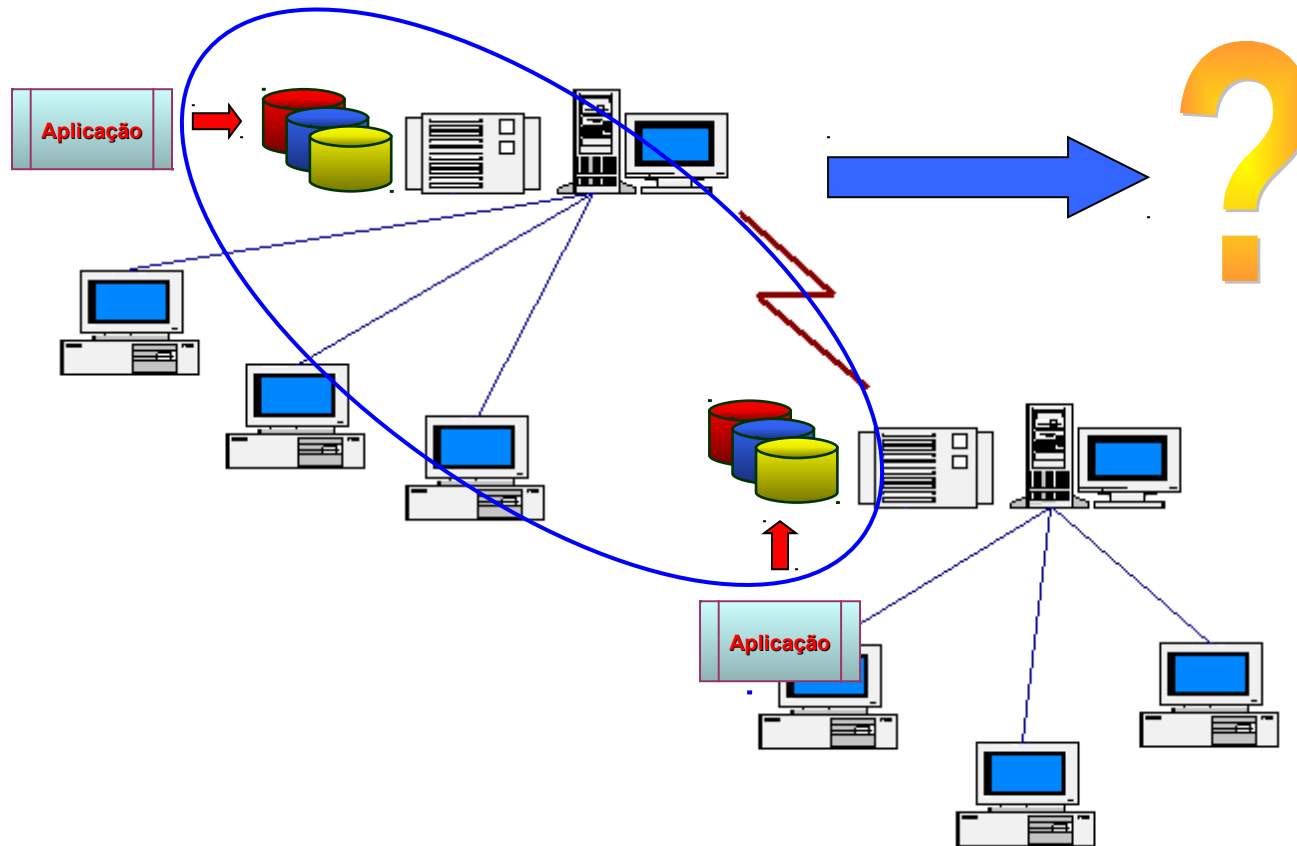


# Replicação Bi-Direcional



# Replicação Bi-Direcional

## Assíncrona: Colisão de Dados



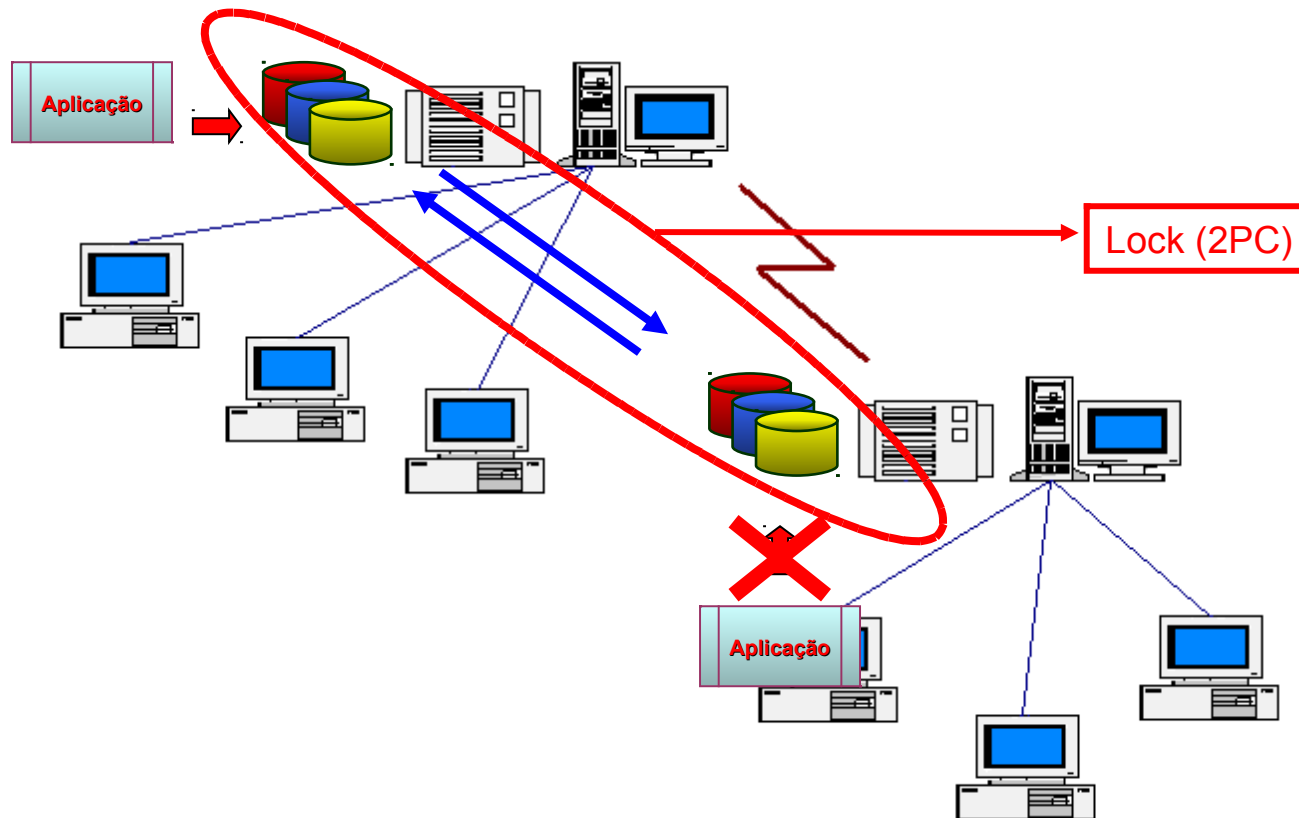
# Replicação Bi-Direcional

## Assíncrona: Colisão de Dados

- Critérios de resolução de conflitos
  - Prioridade
  - Temporalidade
  - Origem de dados (Localização)
- Definição de métodos de resolução
  - Resolução via SGBD
  - Resolução via Aplicação

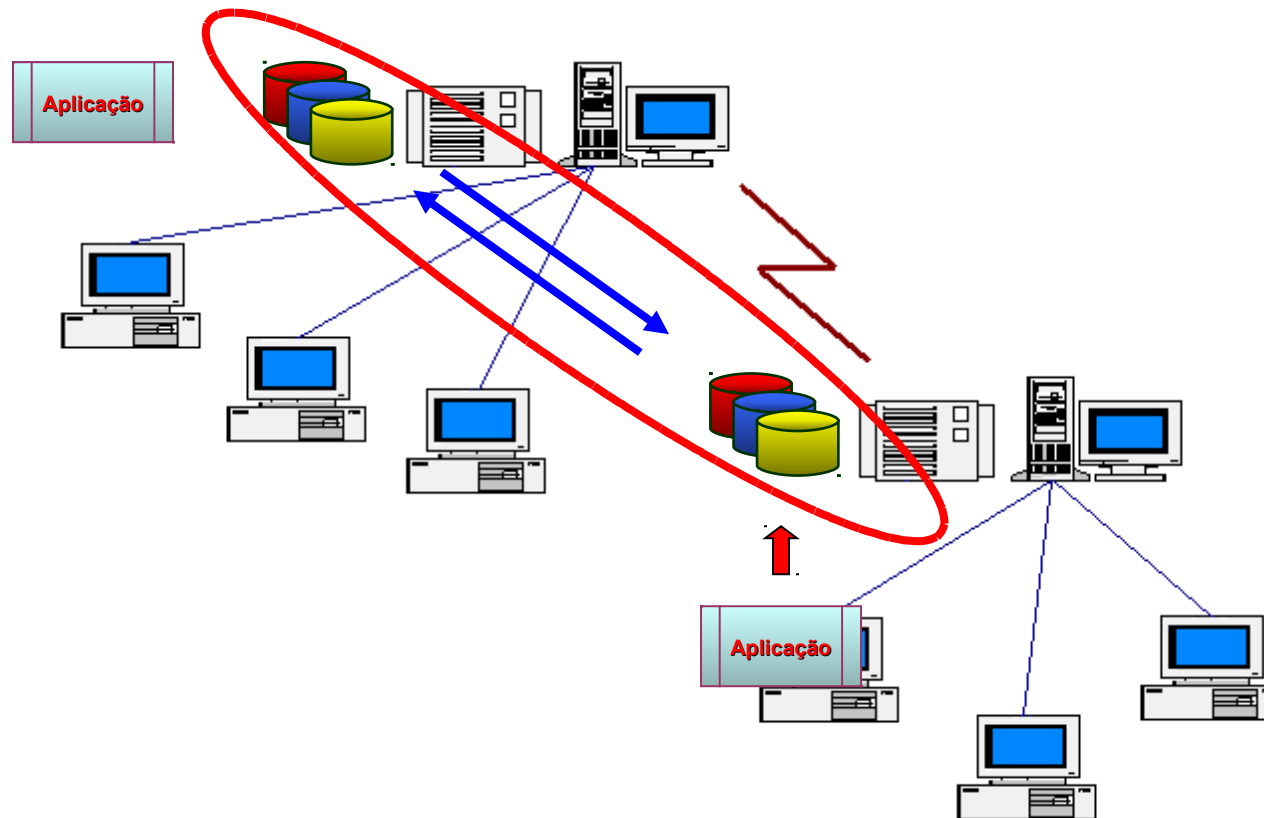
# Replicação Bi-Direcional

Síncrona: Two Phase Commit



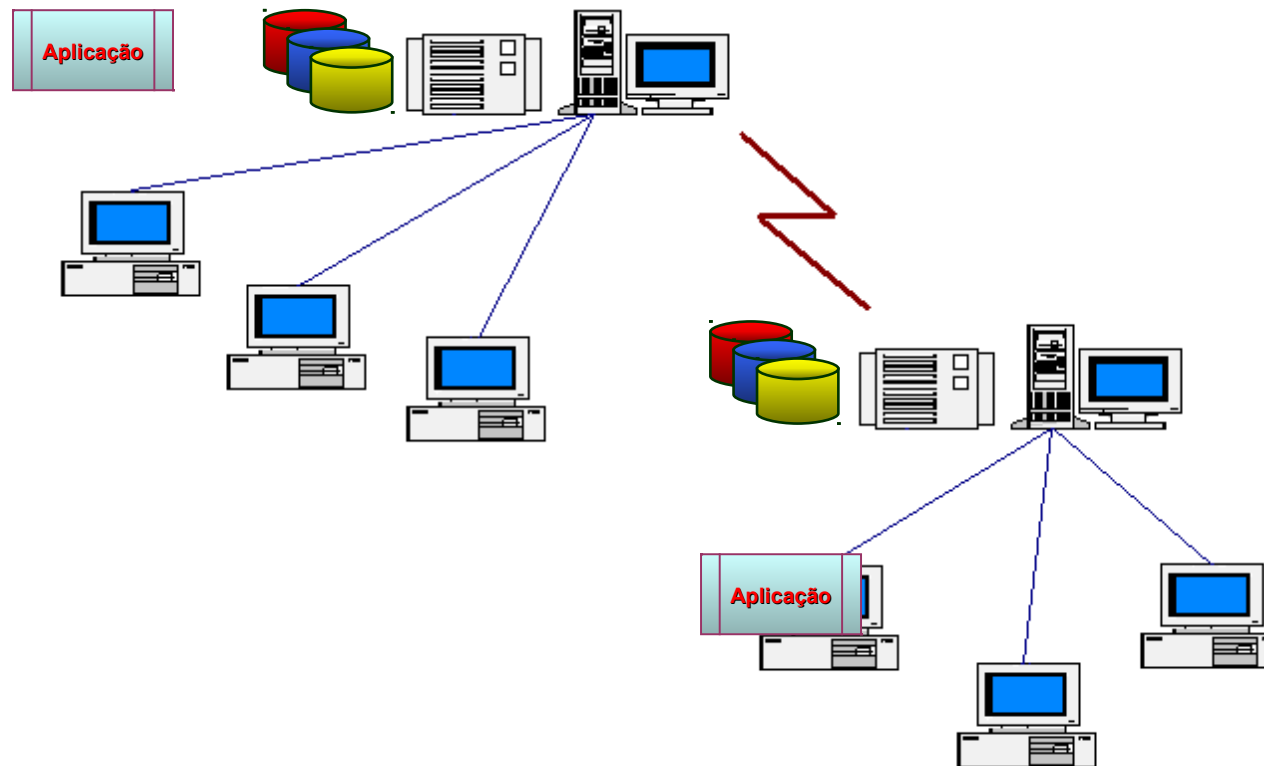
# Replicação Bi-Direcional

Síncrona: Two Phase Commit



# Replicação Bi-Direcional

Síncrona: Two Phase Commit



# Replicação Bi-Direcional

## Síncrona: Two Phase Commit

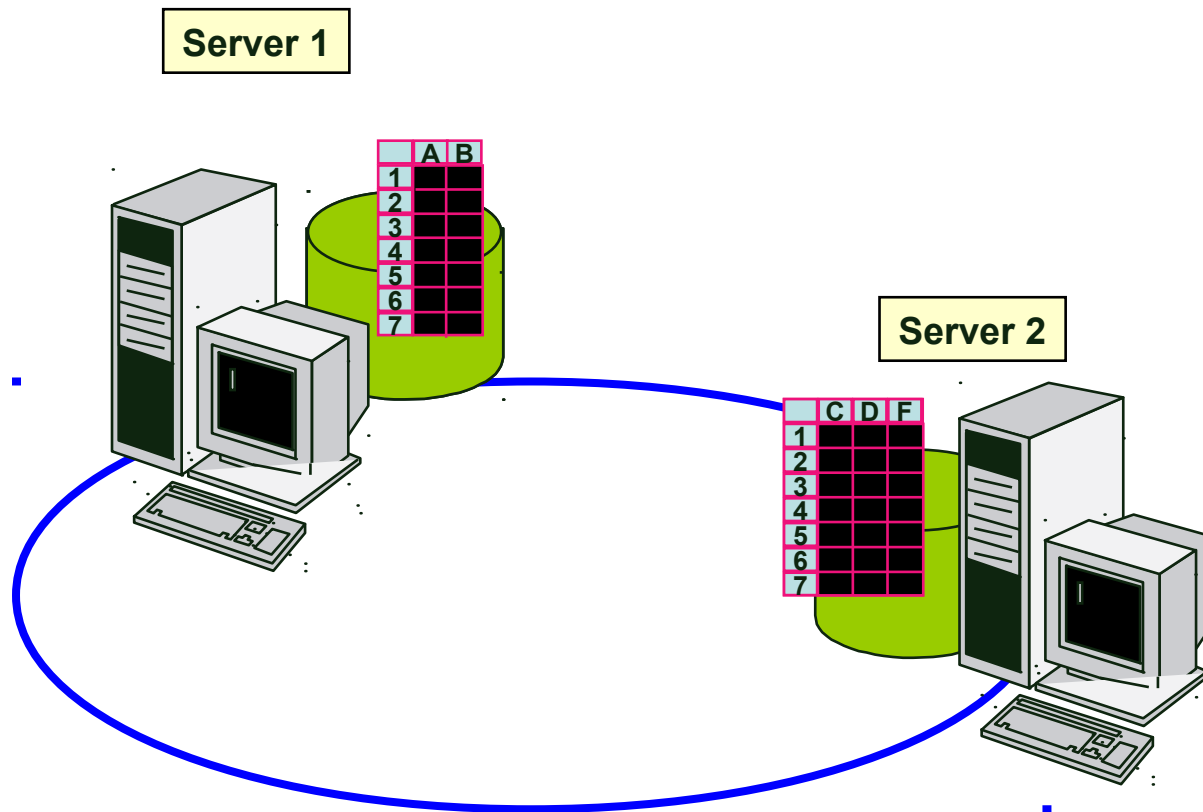
- Necessidade de dados atualizados e síncronos em tempo real
- Necessidade de precisão absoluta dos dados
- Alta consistência >> Dependência entre servidores



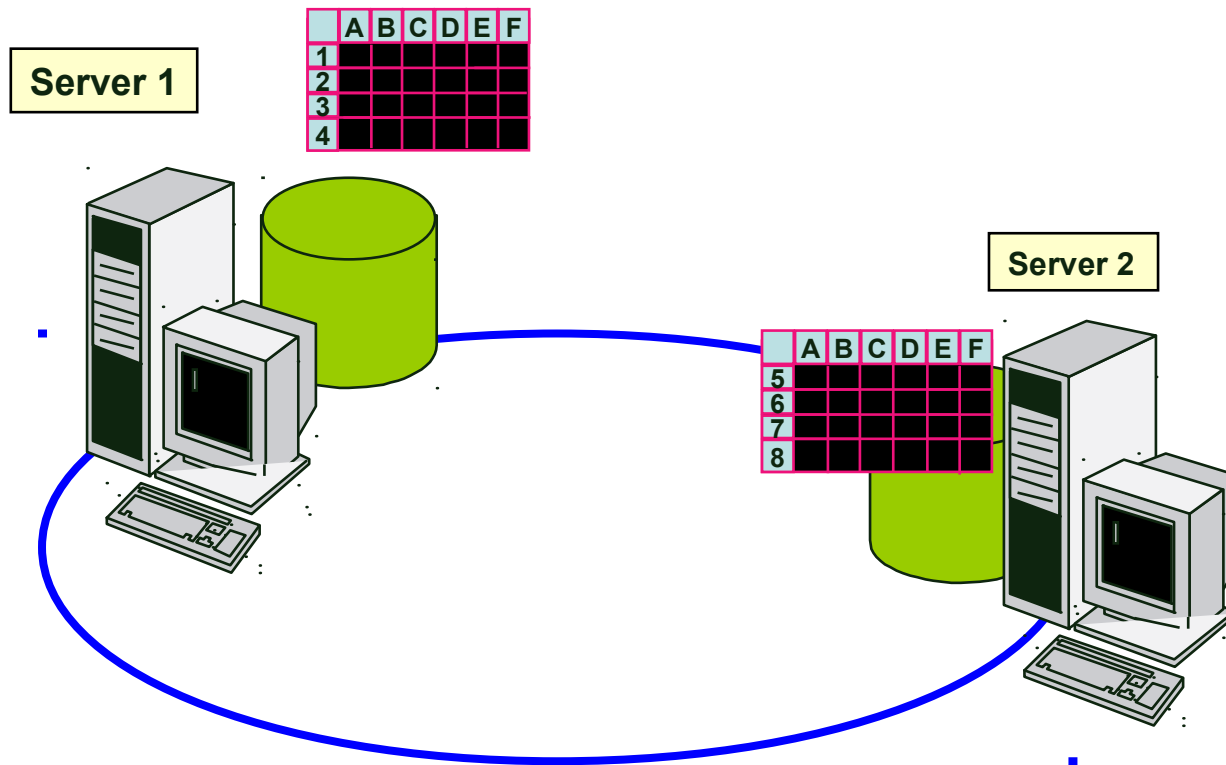
# Fragmentação

- Dados distribuídos em vários “sites” geograficamente dispersos;
- Dados distribuídos são parte de uma unidade lógica;
- Dados podem ser fragmentados em nível horizontal e/ou vertical;
- A localização dos dados fragmentados deve ser transparente para aplicações;
- O SGBD é responsável pelo ponteiamento e endereçamento dos fragmentos;
- O SGBD deve manter o controle transacional de alterações independentemente da localização dos fragmentos de dados envolvidos.

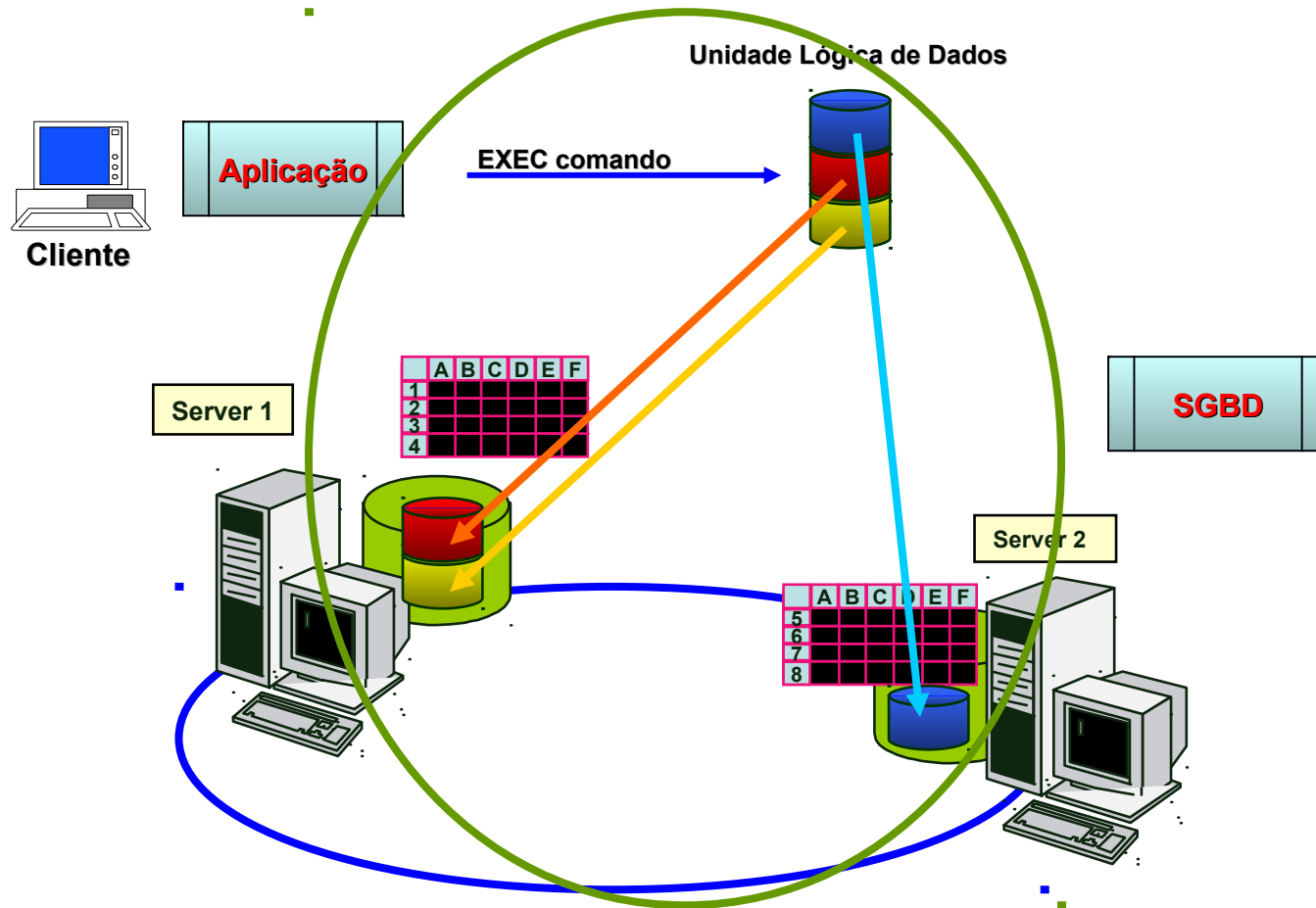
# Fragmentação Vertical



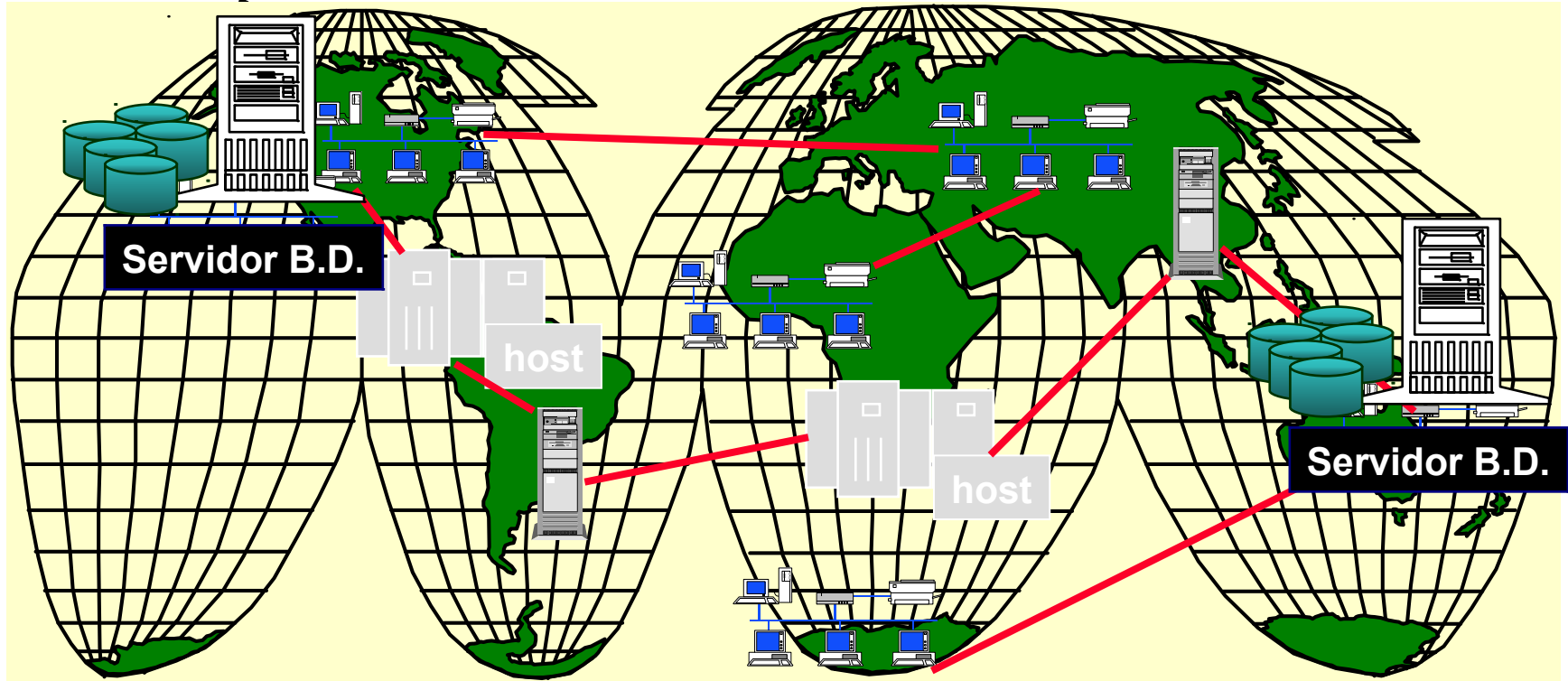
# Fragmentação Horizontal



# Fragmentação

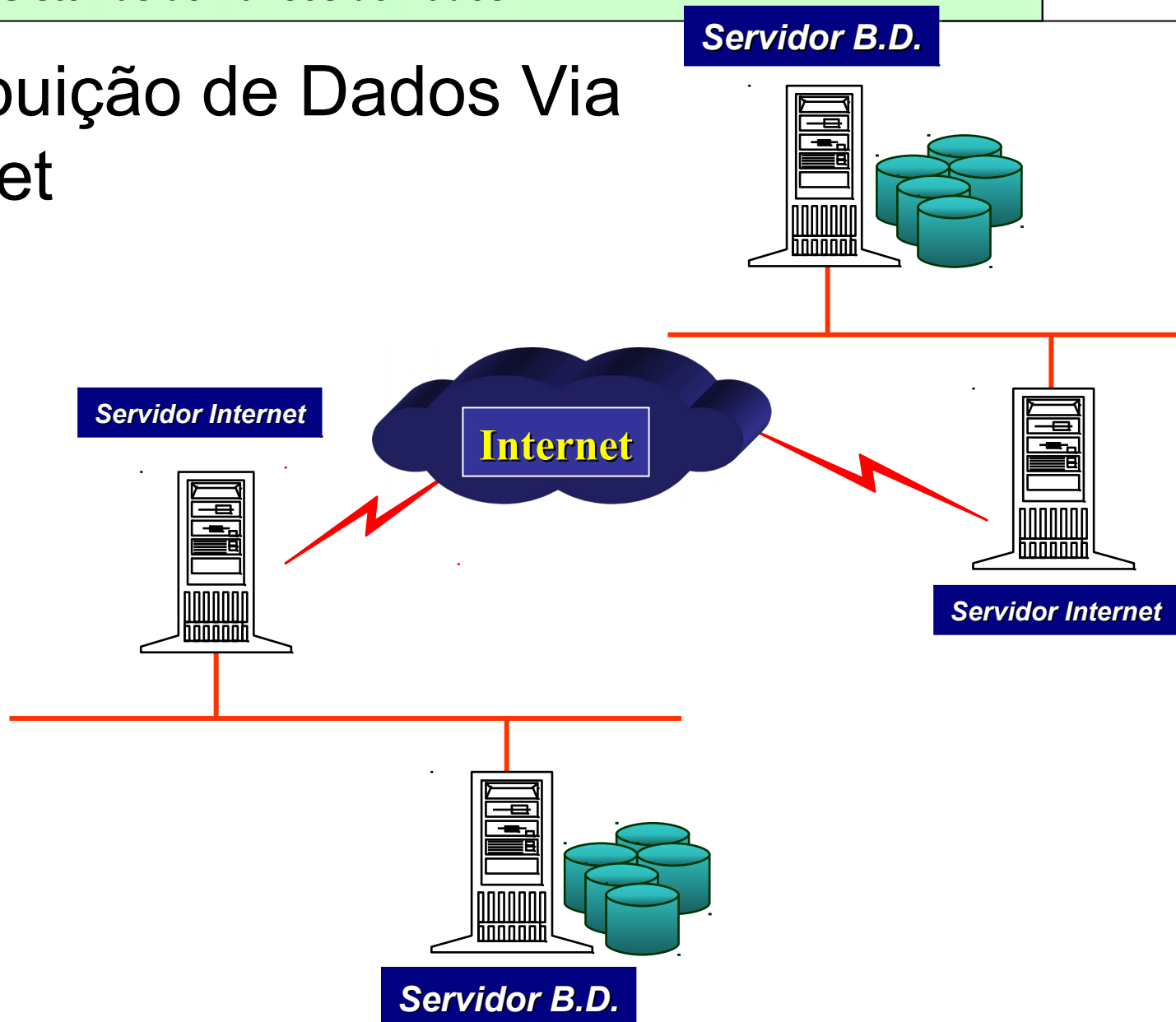


# Distribuição de Dados Via Internet



- Meio não confiável (segurança e conectividade);
- Deve ser usada preferencialmente para métodos assíncronos;
- Banda de comunicação: de média a estreita.

# Distribuição de Dados Via Internet



# Metodologia de Projeto

- Avaliar a necessidade de dados distribuídos
  - Distribuição x Centralização
- Avaliar as formas de conexão entre os vários servidores
- Avaliar o Custo de implantação
  - SGBD
  - Infra-Estrutura
  - Projeto e Desenvolvimento

## E hoje, vai ter saída após a aula?



## Tarefas para a próxima aula:

- Consultar site do Moodle