

INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
MARANHÃO

Prof. MSc. Flávio Barros

*flavioifma@gmail.com*

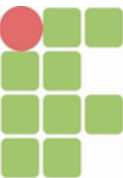
**[www.flaviobarros.com.br](http://www.flaviobarros.com.br)**

# Programação para Dispositivos Móveis

## Aula - Arquiteturas de Aplicação Móvel

## Roteiro

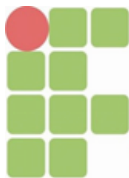
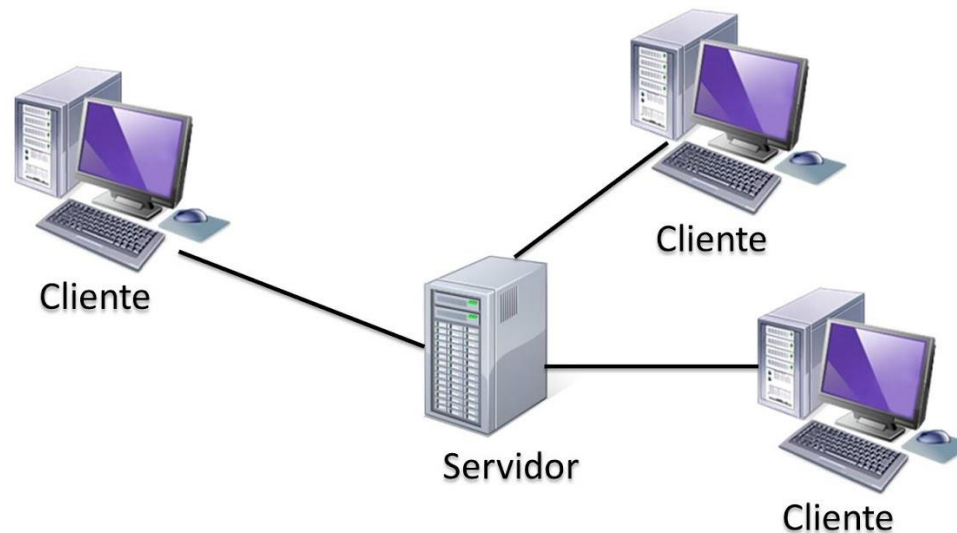
# Arquiteturas de Aplicação Móvel



# Arquiteturas de aplicação

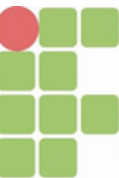
## Definição

- São comumente modeladas em termos de uma arquitetura Cliente/ Servidor, onde um ou mais dispositivos Clientes solicitam informações a um Servidor.



# Roteiro

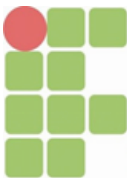
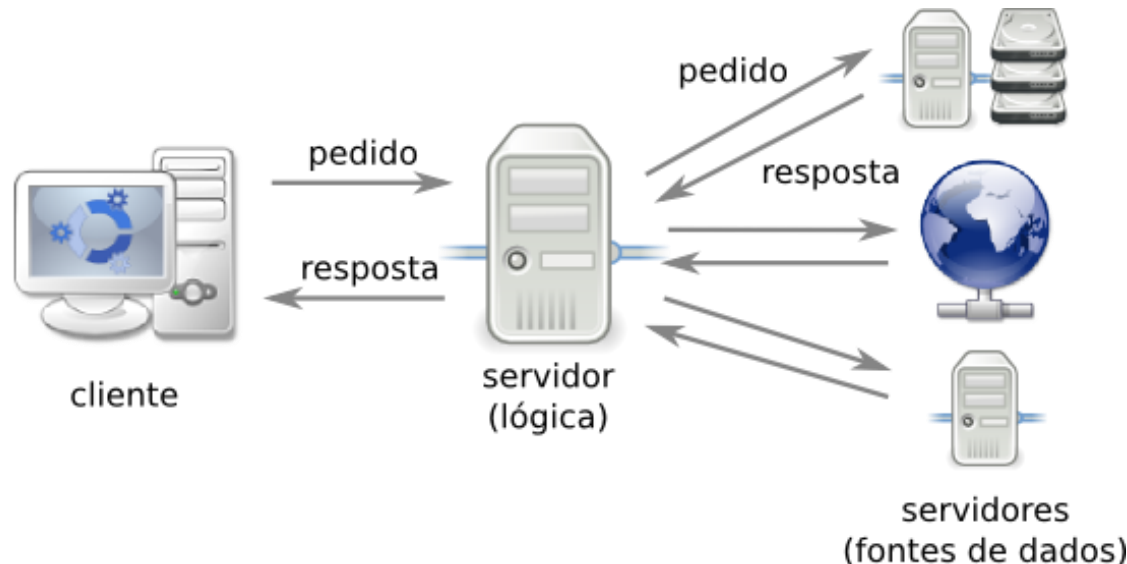
## Arquitetura Cliente/ Servidor



# Arquitetura : Cliente/ Servidor

## Características

- Refere-se ao método de distribuição de aplicações computacionais através de muitas plataformas;
- Tipicamente essas aplicações estão divididas entre um provedor de acesso e uma central de dados e numerosos clientes contendo uma interface gráfica para usuários para acessar e manipular dados.

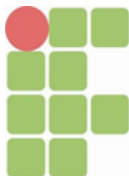
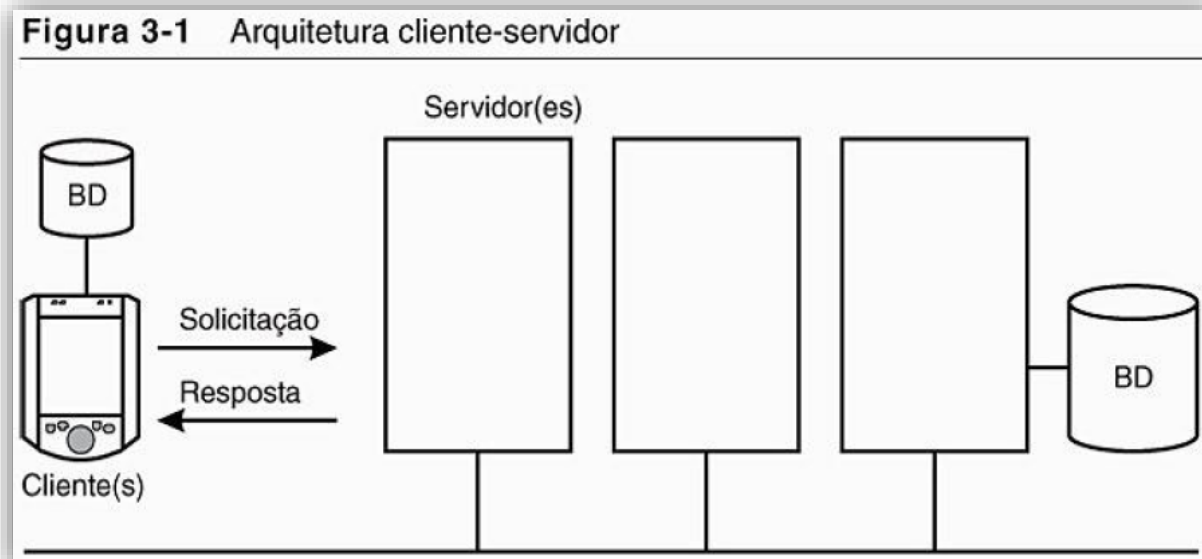


# Arquitetura : Cliente/ Servidor

## Características

- Um ou mais dispositivos clientes solicitam informações a um servidor, onde este em geral responde com as informações solicitadas;
- As arquitetura Cliente Servidor e a Comunicação utilizam: **Camadas** e **Filas**.

[LEE; SCHNEIDER; & SCHELL, 2005]

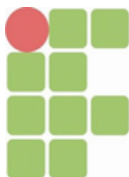
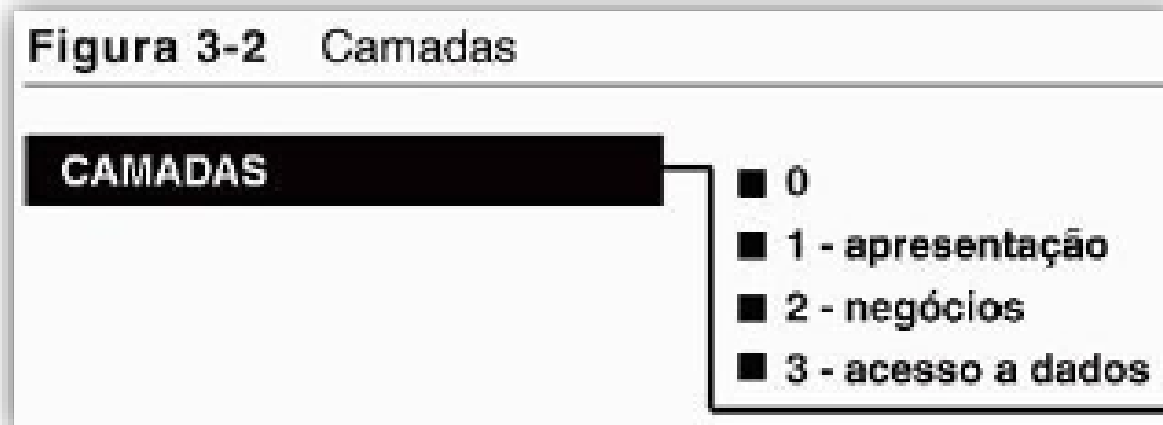


# Arquitetura : Cliente/ Servidor

- **Em Camadas**

- Divide o trabalho dentro do código, separando as tarefas pertinentes;
- Separam a lógica da aplicação, comunicação com banco de dados e a interface com o usuário;
- São elas:

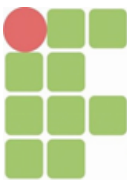
[LEE; SCHNEIDER; & SCHELL, 2005]



# Arquitetura : Cliente/ Servidor

- **Em Camadas**

- **Apresentação:** está mais próxima do usuário, é utilizada para exibir a interface com o usuário.
- **Negócios:** contém a lógica comercial do software.
- **Acesso a dados:** trata a comunicação com o banco de dados.





# Arquitetura : Cliente/ Servidor

- **Em Camadas**

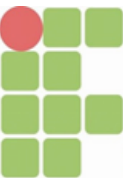
- Os clientes também podem ser **magros** ou **gordos**:

- **Cliente Magro**

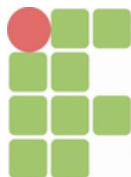
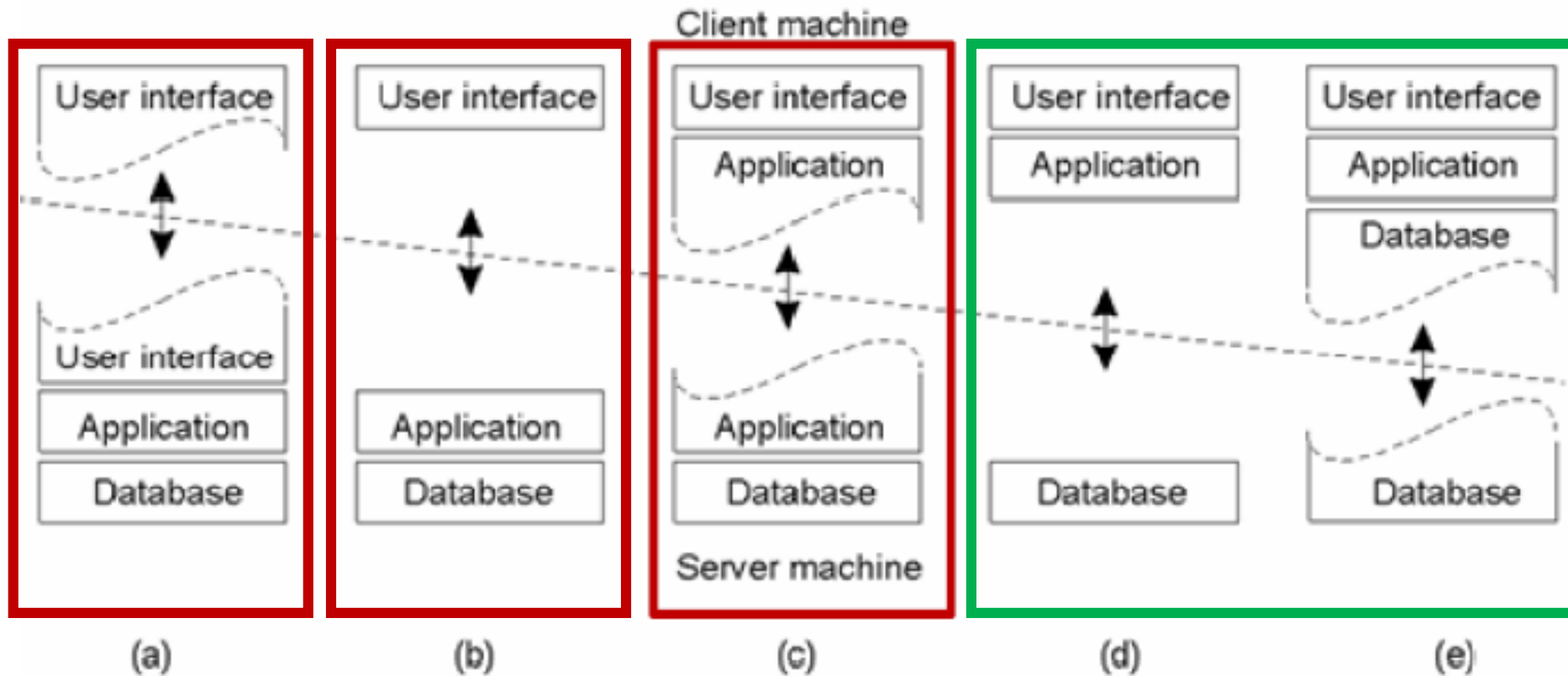
- Um cliente sem camada de código, ou seja, não tem código da aplicação personalizado;
- Abordagens “a”, “b” e “c”: **mais fáceis de gerenciar**.

- **Cliente Gordo**

- Servidor armazena todo o código personalizado da aplicação, um cliente com uma a três camadas de código da aplicação;
- Abordagens “d” e “e”: **evitar**.



# Arquitetura : Cliente/ Servidor



# Exemplo de Arquitetura Cliente/ Servidor em Camadas

## Cliente com 1 camada

**Aplicativo**  
Interface com o usuário

**Website**  
Interface com o usuário

**API**  
Regra de Negócios

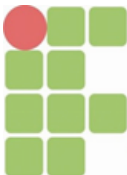
**Acesso a Dados**  
Manipulação da base de dados

Coleta o nome do usuário a ser buscado e detecta o clique no botão de busca.

Verifica o formato correto da entrada do usuário e aciona a camada de acesso a dados.

Realiza a query SQL que corresponde à recuperação de registros da busca.

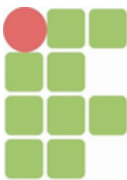
Servidor com 2 camadas



# Arquitetura : Cliente Servidor

- **Em Fila**

- Embora a divisão do código da aplicação em camadas favoreça a reutilização do código, **ela não torna a arquitetura escalável**;
- Deve-se **distribuir o código entre diversas máquinas**;
- As filas descrevem a divisão de trabalho de código da aplicação em **diversas máquinas**:
  - 1ª fila: **Apresentação**;
  - 2ª fila: **Aplicação**;
  - 3ª fila: **Base de dados**.



# Arquitetura : Cliente Servidor

- **Em Fila**

- 1ª fila: **Apresentação**

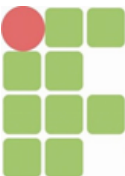
- O código que interage mais próximo do usuário;

- 2ª fila: **Aplicação**

- Armazena a lógica de negócios da aplicação e a lógica de acesso a dados;

- 3ª fila: **Base de dados**

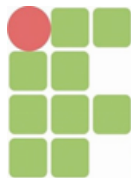
- Abriga o próprio banco de dados ou origem de dados.



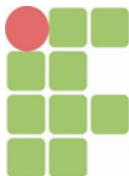
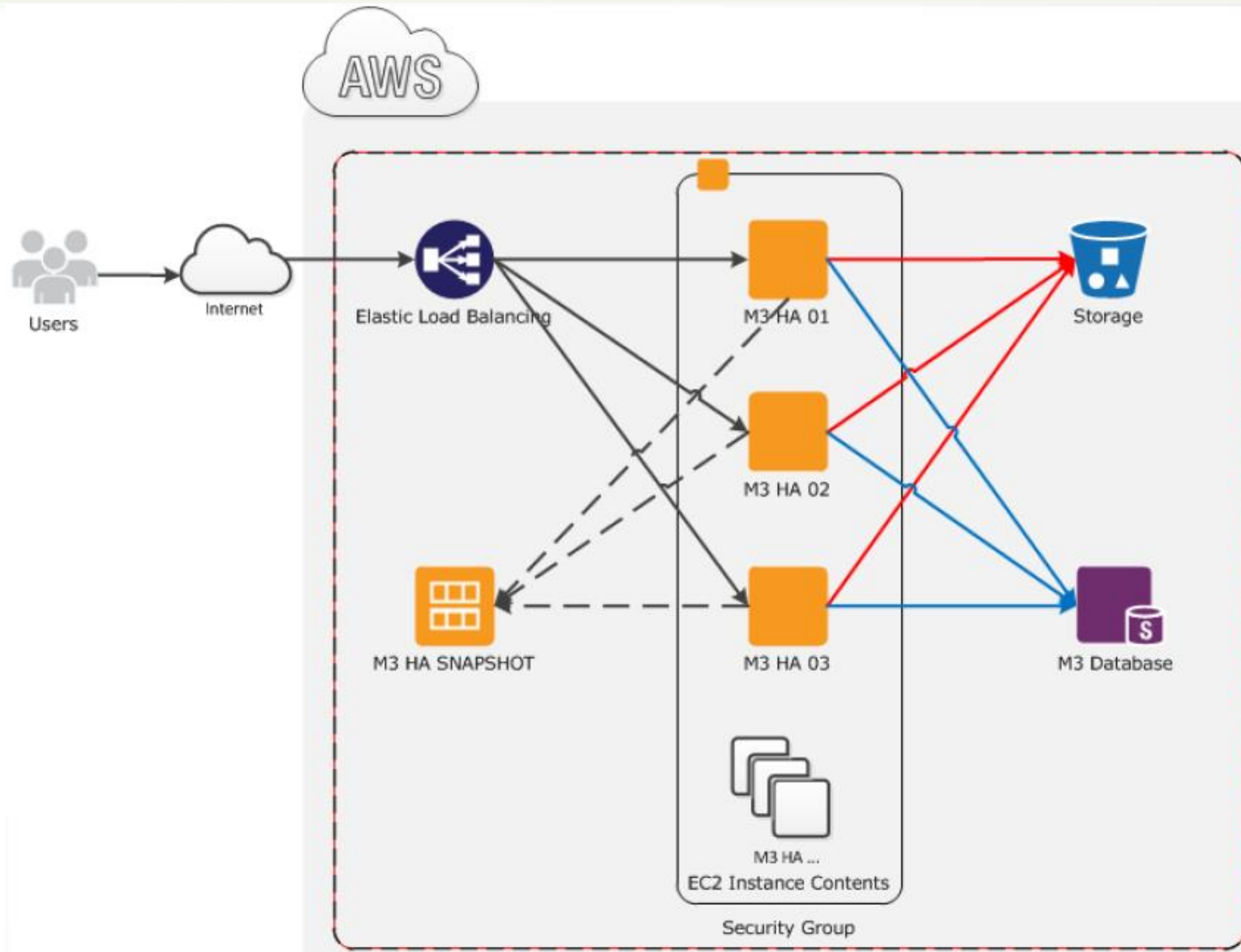
# Exemplo de Arquitetura Cliente/ Servidor

Empresa na Web faz queima de estoque e está recebendo muitos acessos simultâneos, tem muitos clientes querendo comprar na promoção.

- O servidor de aplicação não aguenta responder e processar todas as solicitações.
- Utilizando uma arquitetura escalável, basta subir mais servidores de aplicação para responder as demais solicitações.



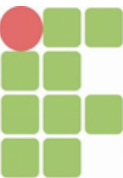
# Exemplo de Arquitetura Cliente Servidor em Filas



# Resumo

A capacidade de adicionar mais servidores é denominada escalonamento horizontal (*ou scaling out*).

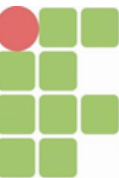
Já a capacidade de adicionar servidores mais poderosos é denominada escalonamento vertical (*scaling up*).





# Roteiro

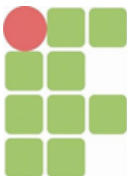
## Arquitetura Cliente



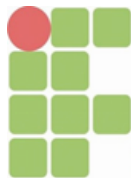
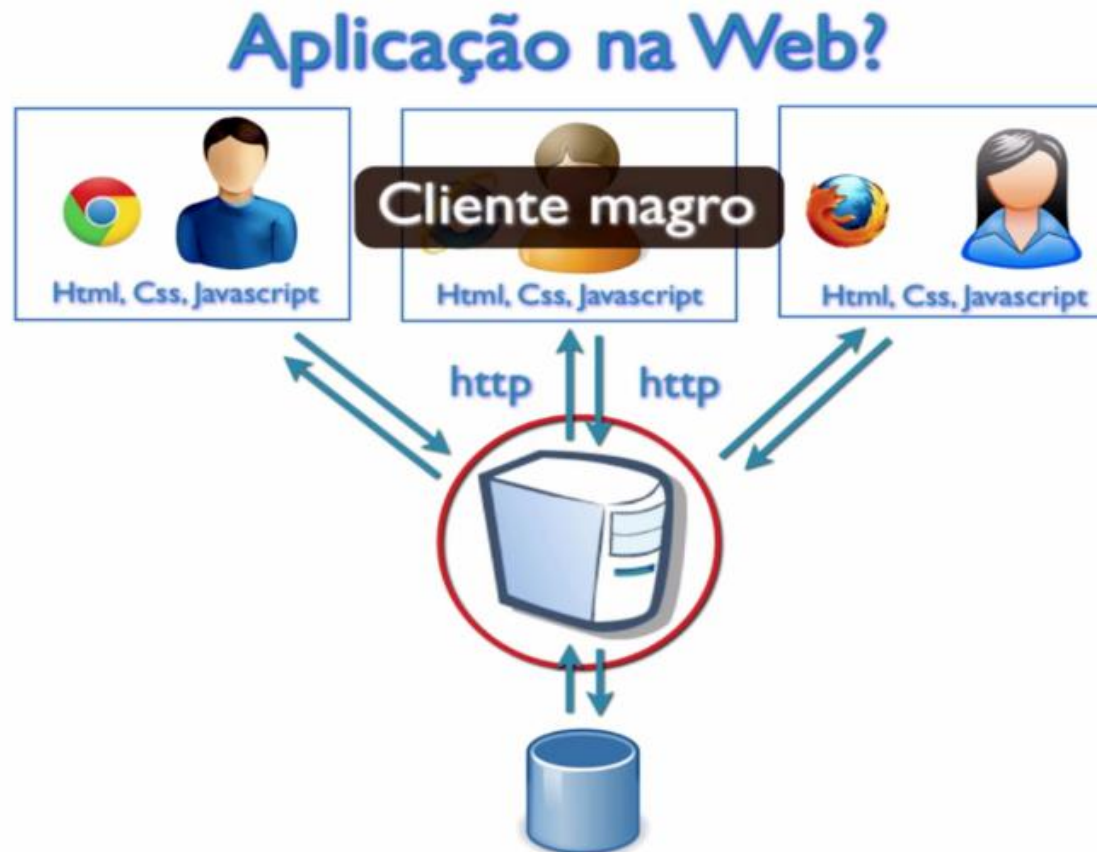
# Clientes Magros

- **Características**

- Não possuem código personalizado na aplicação, **dependem 100% do servidor**;
- Funcionam em **qualquer sistema operacional**;
- **Independente de dispositivo**;
- Geralmente **acessíveis através de navegadores web**;
- Conhecidos como **aplicativos Web Apps**.



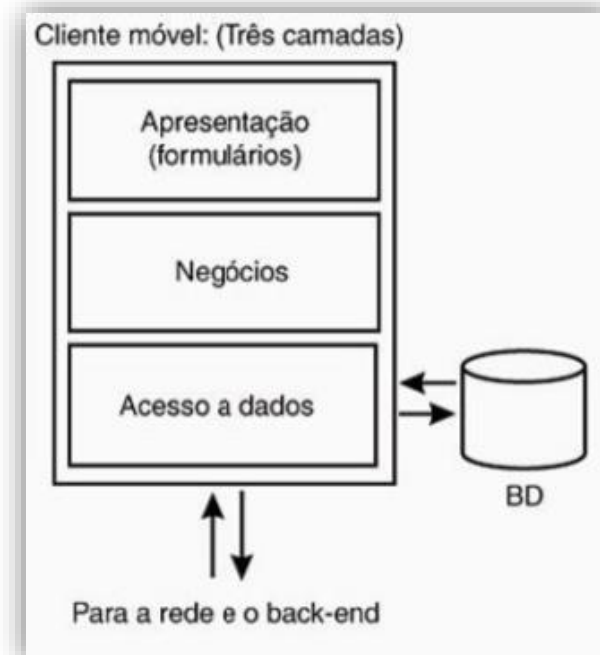
# Clientes Magros



# Arquitetura Cliente

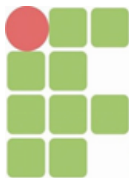
## • Clientes Gordos

- Dependem diretamente do sistema operacional e do dispositivo;
- Possuem de **uma a três camadas** de código;
- **Úteis** quando **não há garantias de comunicação permanente com o servidor**;
- Armazenam dados localmente até que possam ser sincronizados com um servidor;
- Conhecidos como **Aplicativos Nativos**;
- Alguns podem ser considerados **Aplicativos híbridos**.



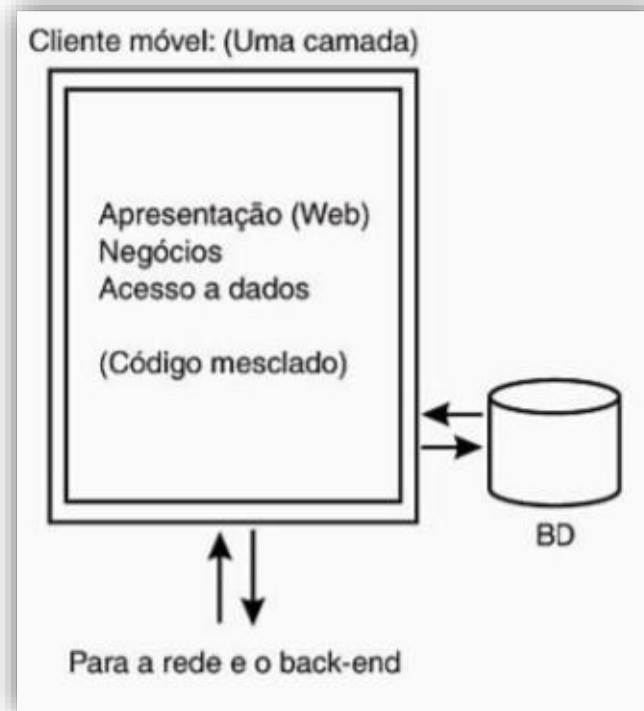
[LEE; SCHNEIDER; & SCHELL, 2005]

# Arquitetura Cliente

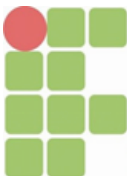


# Arquitetura Cliente

- **Hospedagem de página web**
  - Disponibilizam um **servidor web no dispositivo**;
  - Dependem parcialmente do sistema operacional e do dispositivo; Possuem as **3 camadas de código**.

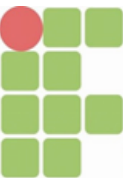


[LEE; SCHNEIDER; & SCHELL, 2005]



# Roteiro

## Arquitetura Servidor



# Arquitetura Servidor

- **One-tier (1 fila)**

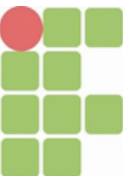
- As 3 camadas de código ficam num mesmo servidor.

- **Two-tier (2 filas)**

- O servidor do banco de dados fica separado do servidor da aplicação.

- **Three-tier ( 3 filas)**

- Os servidores de aplicação, negócios e banco de dados ficam em locais separados.





# Arquitetura Servidor

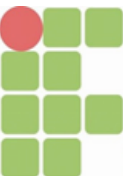
- **One-tier (1 fila)**

- **Vantagem**

- Muito conveniente;
    - Rápido de desenvolver e implantar.

- **Desvantagem**

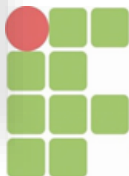
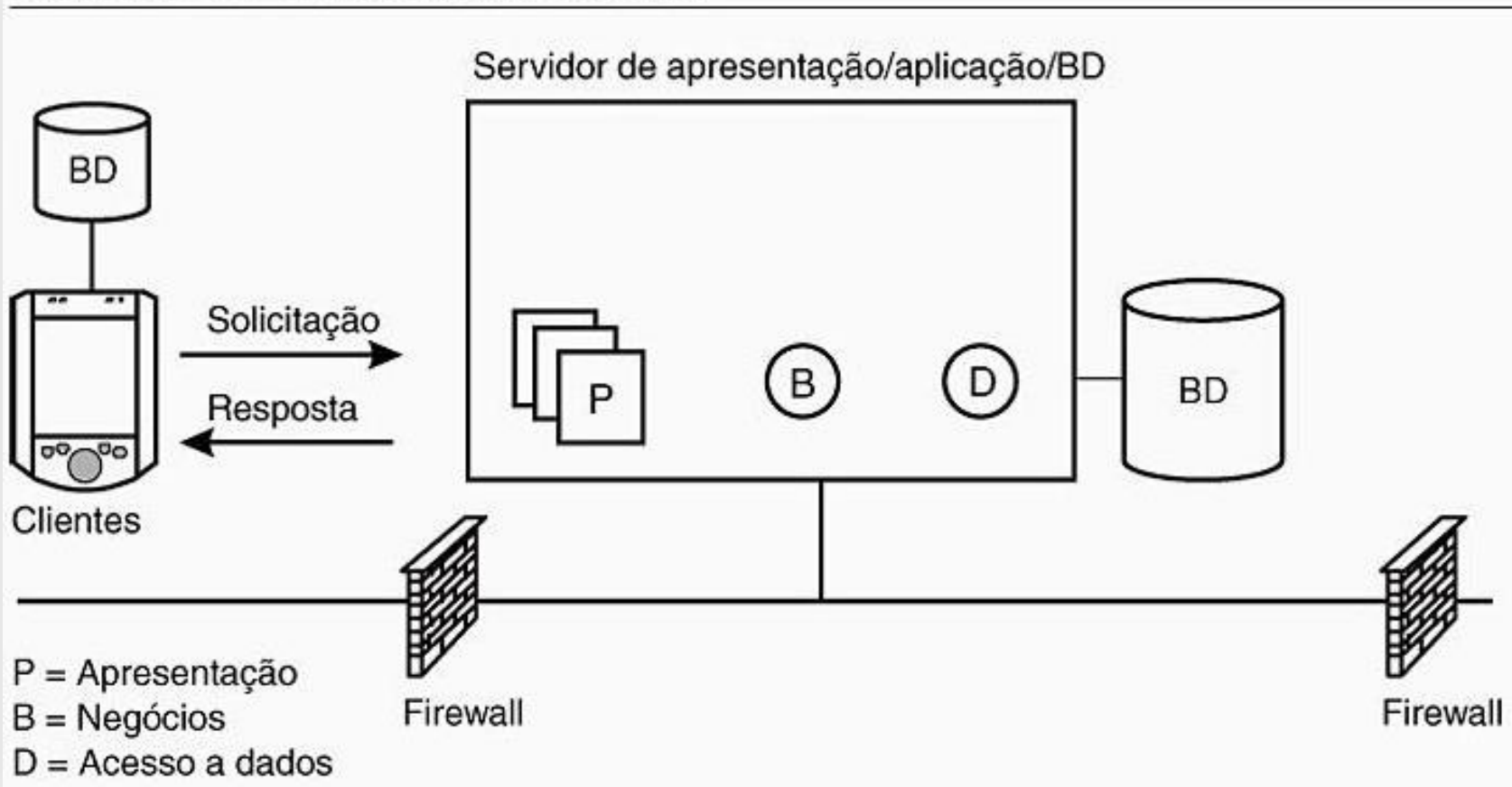
- Menos escalável;
    - Difícil lidar com a segurança da aplicação.



# Arquitetura Servidor

[LEE; SCHNEIDER; & SCHELL, 2005]

**Figura 3-12** Arquitetura de uma fila



# Arquitetura Servidor

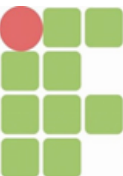
- **Two-tier (2 filas)**

- **Vantagem**

- Permite escolher outro local para armazenamento de dados.

- **Desvantagem**

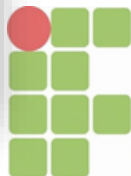
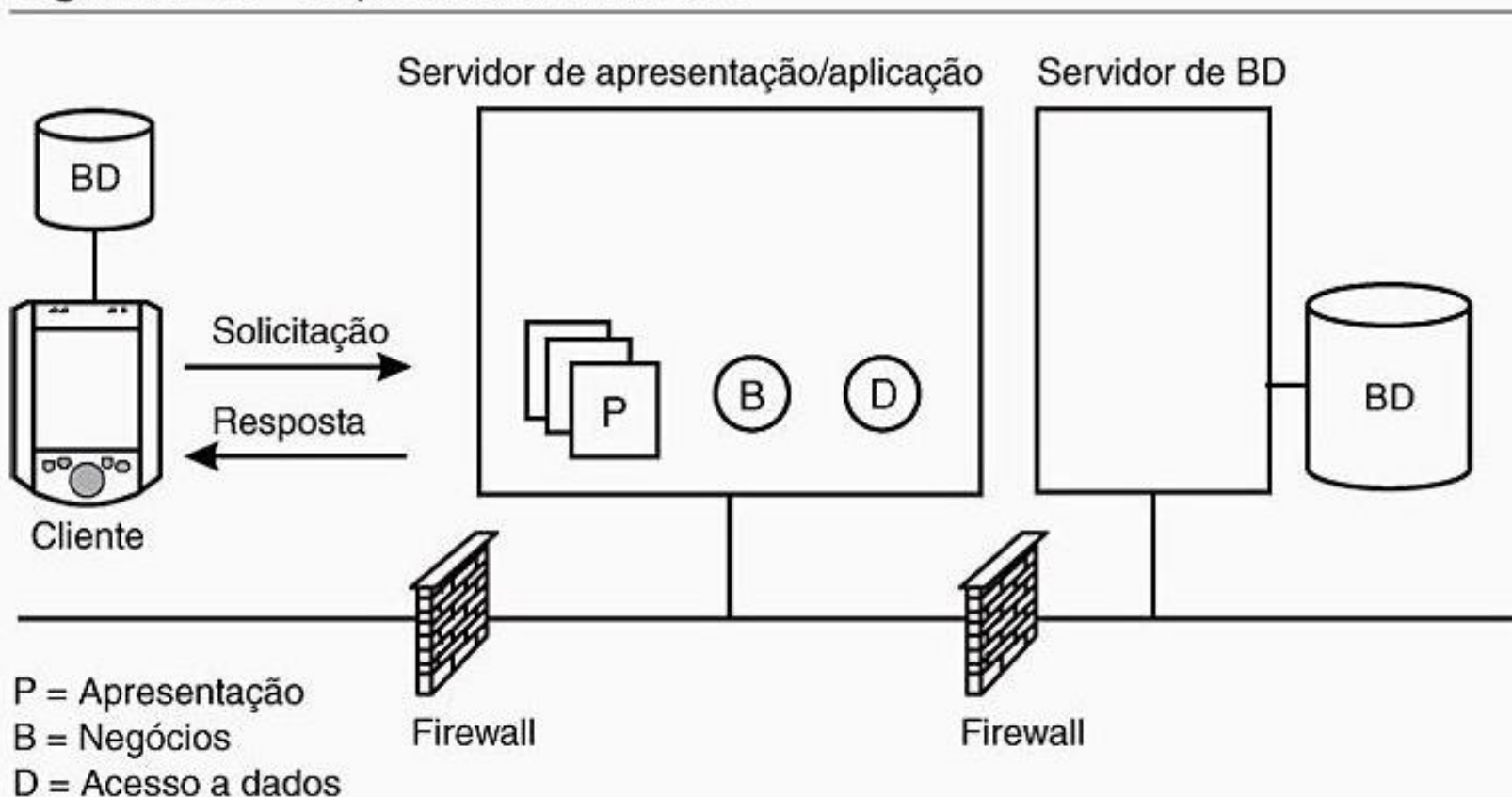
- Mais caro;
- Menos escalável;
- Difícil lidar com a segurança da aplicação.



# Arquitetura Servidor

[LEE; SCHNEIDER; & SCHELL, 2005]

**Figura 3-13** Arquitetura de duas filas



# Arquitetura Servidor

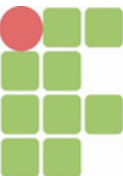
- **Three-tier ( 3 filas)**

- **Vantagem**

- Escalável;
- Protegida por Firewalls;
- Permite a especialização de Servidor do Banco de Dados.

- **Desvantagem**

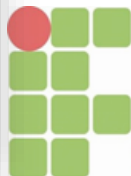
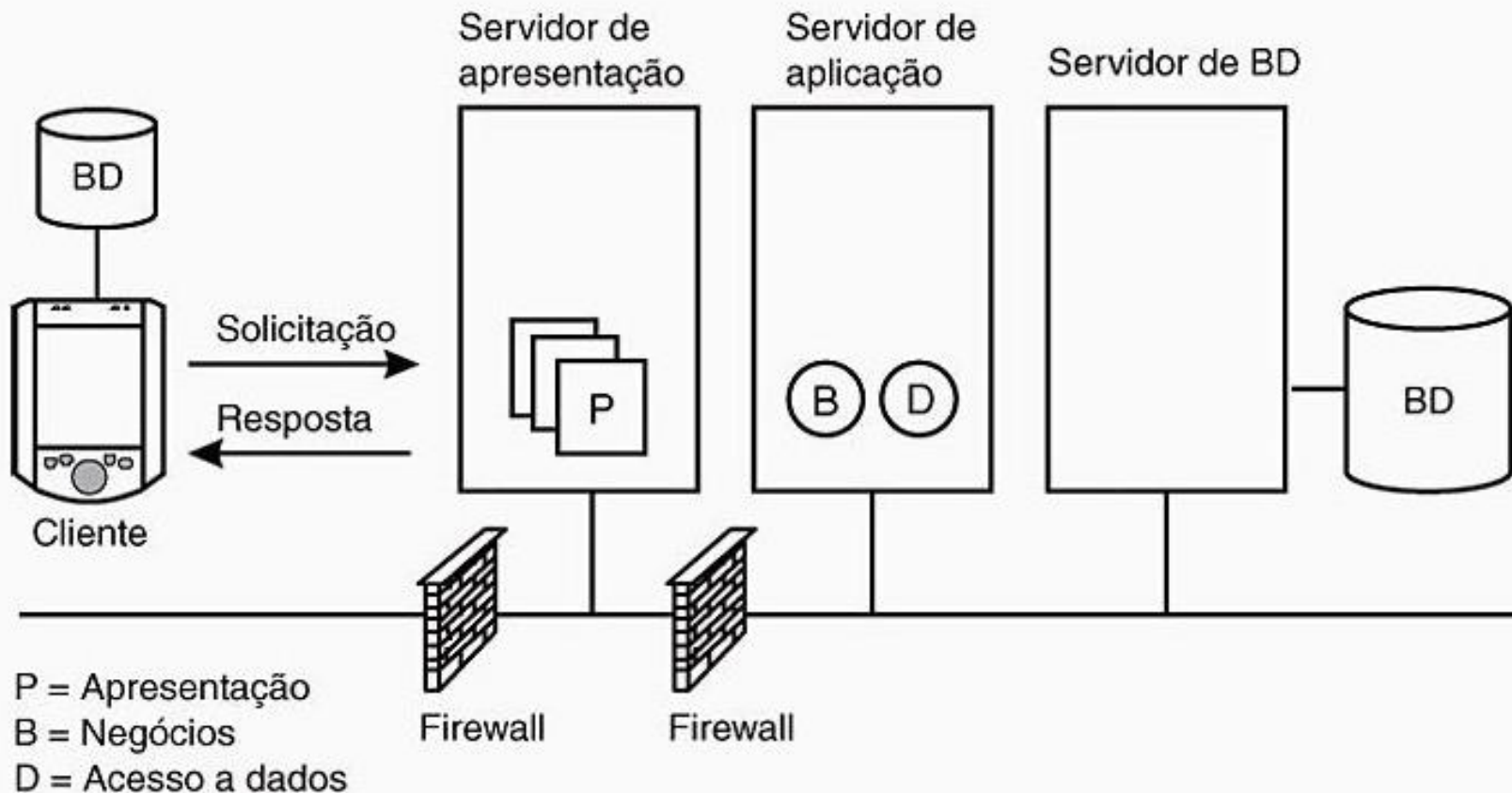
- Mais difícil de ser desenvolvida e gerenciada;
- Mais cara.



# Arquitetura Servidor

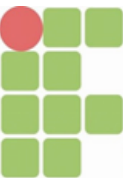
[LEE; SCHNEIDER; & SCHELL, 2005]

**Figura 3-14** Arquitetura de três filas



## Roteiro

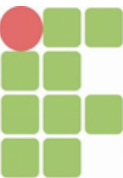
# Sincronização dos Dispositivos Móveis



# Sincronização dos Dispositivos Móveis

## Fundamentos

- O tipo de conexão afeta a maneira como se pode sincronizar dados entre o dispositivo móvel e sistemas back-end;
- A sincronização pode ser efetuada de duas maneiras:
  - Comunicação contínua;
  - Método de Armazenamento e Encaminhamento.

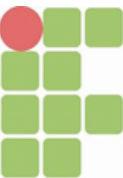




# Sincronização dos Dispositivos Móveis

## Comunicação contínua

- As sincronizações de dados entre cliente e servidores são contínuas e podem ser **Síncrona** ou **Assíncrona**.
- **Síncrona**
  - Uma solicitação para armazenar dados é enviada para o servidor e seguida pelos dados a serem armazenados;
  - Os dados são completamente armazenados antes que o servidor confirme o recebimento deles e libere a interface com o cliente.

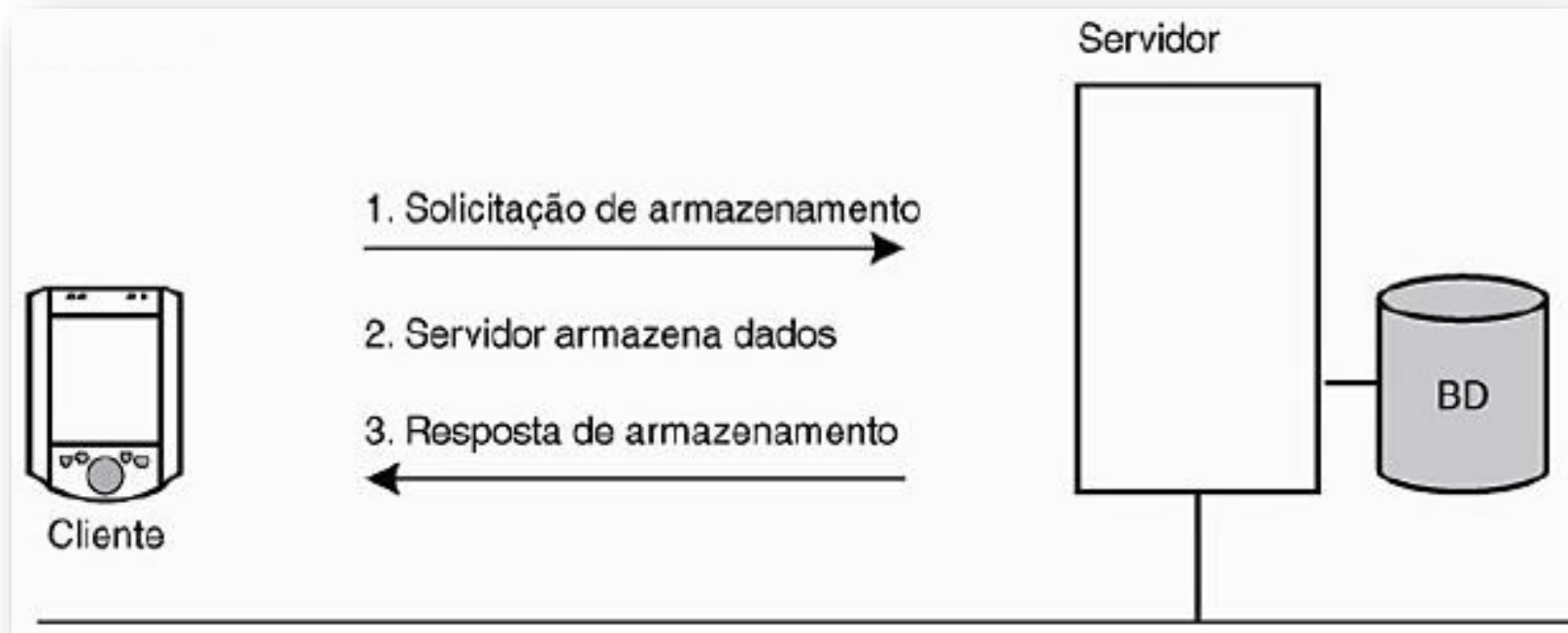


# Sincronização dos Dispositivos Móveis

## Comunicação contínua

- Síncrona

[LEE; SCHNEIDER; & SCHELL, 2005]



# Sincronização dos Dispositivos Móveis

## Comunicação contínua

- Assíncrona

- Uma solicitação para armazenar dados é enviada para o servidor seguida pelo armazenamento de dados;
- Os dados são colocados em uma área de armazenamento - BD, no servidor;
- Os dados não precisam ser armazenados completamente antes que o servidor realize a confirmação ao cliente;
- Quando a solicitação de armazenamento estiver completa, iniciará uma conversa para informar ao cliente que está pronto.

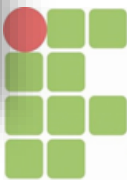


# Sincronização dos Dispositivos Móveis

## Comunicação contínua

- Assíncrona

[LEE; SCHNEIDER; & SCHELL, 2005]



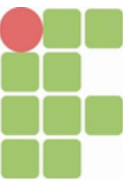
# Sincronização dos Dispositivos Móveis

## Método de Armazenamento e Encaminhamento

- Fundamentação

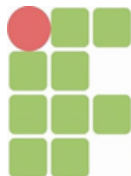
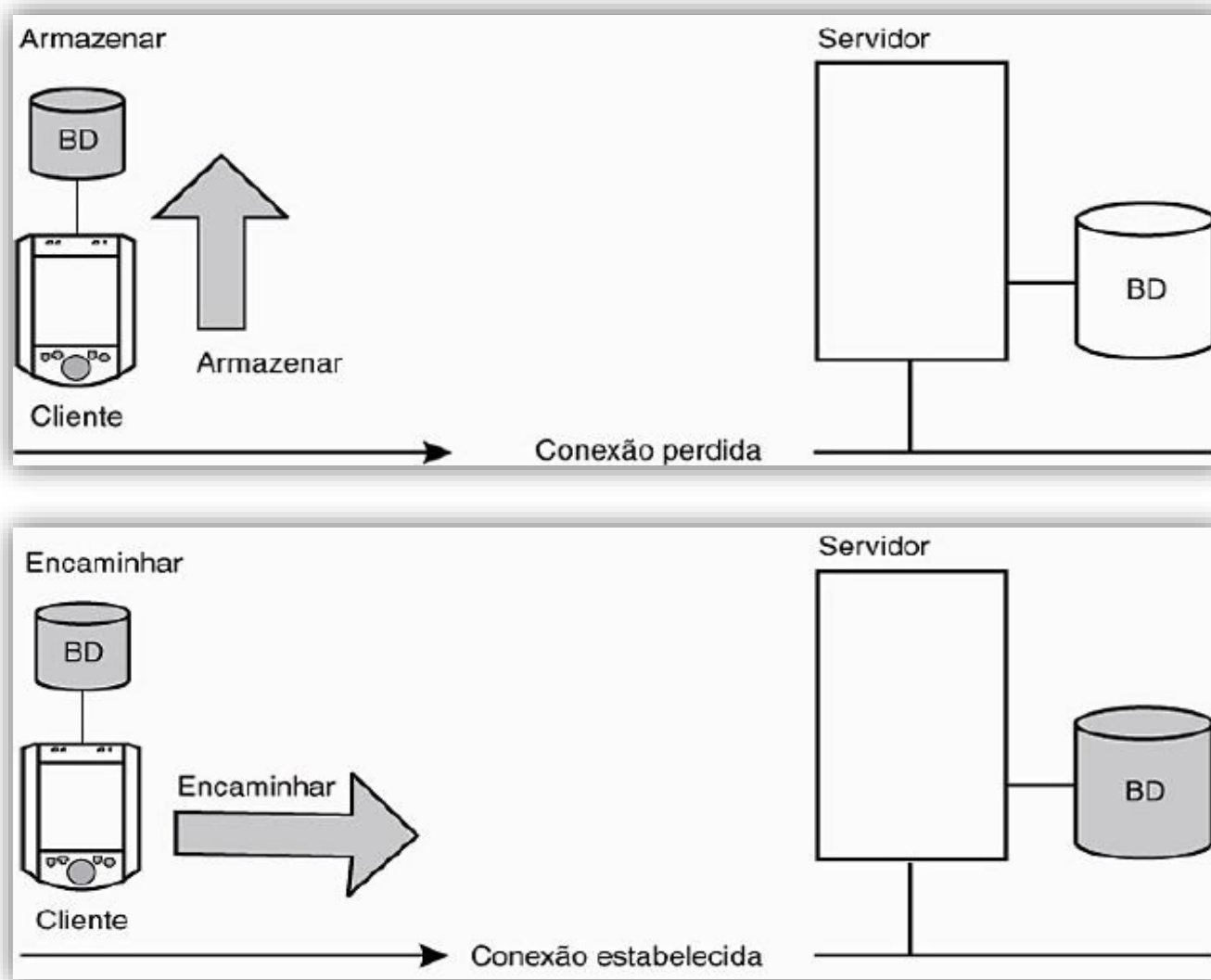
- Quando não se garanti a conectividade entre o cliente e servidor, é possível ainda armazenar e transmitir as informações usando método chamado "**Armazenar e Encaminhar**".

**Situação:** Usuário móvel queira inserir dados enquanto o seu dispositivo móvel não esteja conectado a um servidor, inicialmente esta aplicação armazena os dados em um BD local. Quando a conexão for estabelecida, a aplicação móvel encaminhará os dados do BD local para o BD no servidor.



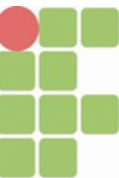
# Método de Armazenamento e Encaminhamento

[LEE; SCHNEIDER; & SCHELL, 2005]



# Roteiro

## Referências



# Referências

- LEE, Valentino; SCHNEIDER, Heather; SCHELL, Robbie. **Aplicações Móveis: arquitetura, projeto e desenvolvimento**. São Paulo: Pearson Education do Brasil, 2005.
- AQUINO, Fabíula de. **Arquitetura Cliente Servidor**.
- TEIXEIRA, Iury. **Tecnologias Móveis**.

