



**Universidade Federal de Santa Catarina**  
**Curso de Tecnologias da Informação e Comunicação**  
**Desenvolvimento de Sistemas Web**  
**Professor: Adriano de Oliveira**

---

# **Desenvolvimento de Sistemas Web**

Prof.: Adriano de Oliveira  
email: [adriano.inovar@gmail.com](mailto:adriano.inovar@gmail.com)  
Turma: 05652 - 3-2020-2 e 5-1830-2



**Universidade Federal de Santa Catarina**  
**Curso de Tecnologias da Informação e Comunicação**  
**Desenvolvimento de Sistemas Web**  
**Professor: Adriano de Oliveira**

---

# **ARQUITETURA DE SISTEMAS WEB**

# ARQUITETURA DE SISTEMAS WEB

- ✓ Qual a melhor forma de desenvolver para a Web?



# ARQUITETURA DE SISTEMAS WEB

- ✓ Importante esclarecer que se a abordagem escolhida para a arquitetura for precária, a escolha da tecnologia influencia pouco.





# ARQUITETURA DE SISTEMAS WEB

- ✓ Existem “N” opções no mercado para desenvolvimento de software.



python

Microsoft  
.net



# ARQUITETURA DE SISTEMAS WEB

- ✓ O que faz diferença e agrega valor é uma arquitetura bem definida.



# ARQUITETURA DE SISTEMAS WEB

- ✓ A partir da arquitetura, seleciona-se as tecnologias adequadas à natureza dos projetos.
- ✓ Exemplo: casa
  - Para que público é a casa?
  - Qual o clima do local?
  - Quais as características do terreno?
  - Qual o custo/benefício das madeiras na região?





# ARQUITETURA DE SISTEMAS WEB

## ✓ Exemplo: software

- Para que público é o software? (b2b, b2c, ...)
- Qual o tipo do software (negócio, ferramenta, embarcado)
- Quais os requisitos funcionais?
- Quais os requisitos não-funcionais?
- Vai rodar em que ambiente?
- É cliente-servidor ou cliente?





## ARQUITETURA DE SISTEMAS WEB

- ✓ Ou seja, uma arquitetura e conjunto de tecnologias montados para um projeto “A” pode não ser a mais adequada para um projeto “B”.
- ✓ Cada caso é um caso.



# ARQUITETURA DE SISTEMAS WEB

- ✓ Uma analogia, usa-se o mesmo tipo de material para estas duas construções?



# **ARQUITETURA DE SISTEMAS WEB**

- ✓ **Modelo 2 Camadas**
- ✓ **Modelo 3 Camadas**
- ✓ **Modelo 4 Camadas**
- ✓ **Padrão MVC**

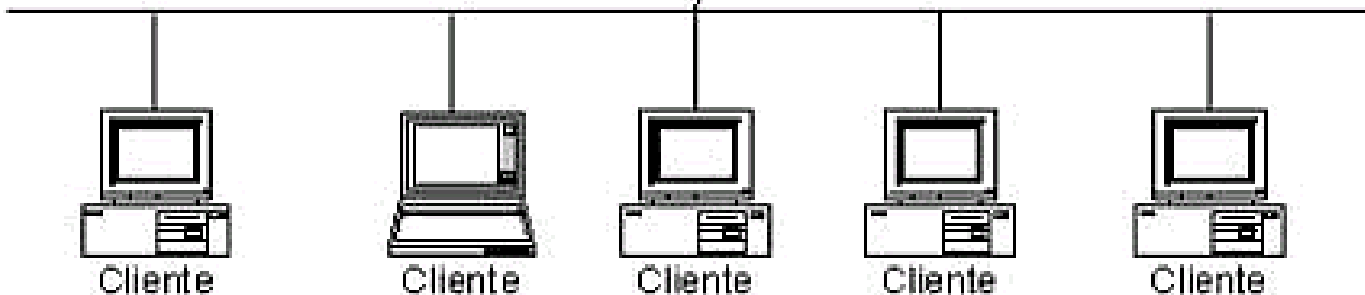
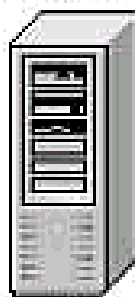


# ARQUITETURA DE SISTEMAS WEB

## ✓ Modelo 2 Camadas

No modelo de 2 camadas, toda a "Lógica do negócio" fica no Cliente. Quando o programa Cliente é instalado, são instaladas todas as regras de acesso ao Banco de dados.

Banco de dados



# ARQUITETURA DE SISTEMAS WEB

## ✓ Modelo 2 Camadas

- Camada de Apresentação e Regras de Negócio
  - ***Apresentação:*** O Código que gera a Interface visível do programa, que é utilizada pelo usuário para acessar a aplicação, faz parte da aplicação Cliente.
  - ***Regra de Negócio:*** Aqui estão as regras que definem a maneira como os dados serão acessados e processados, as quais são conhecidas como "Lógica do Negócio".
- Camada de Persistência
  - O Banco de dados, fica armazenado em Servidor da rede.

# ARQUITETURA DE SISTEMAS WEB

## ✓ Modelo 3 Camadas

No modelo de 3 camadas, toda a "Lógica do negócio" fica no Servidor de Aplicações. Com isso, a atualização das regras do negócio, fica mais fácil.

Servidor de aplicações

Servidor de Banco de dados





# ARQUITETURA DE SISTEMAS WEB

## ✓ Modelo 3 Camadas

- Camada de Apresentação: O Código que gera a Interface visível do programa, que é utilizada pelo usuário para acessar a aplicação, faz parte da aplicação Cliente.
- Camada Lógica ou Regra do Negócio: Aqui estão as regras que definem a maneira como os dados serão acessados e processados, as quais são conhecidas como "Lógica do Negócio".
- Camada de Persistência: O Banco de dados, fica armazenado em Servidor da rede.

# ARQUITETURA DE SISTEMAS WEB

## ✓ Modelo 4 Camadas



## ARQUITETURA DE SISTEMAS WEB

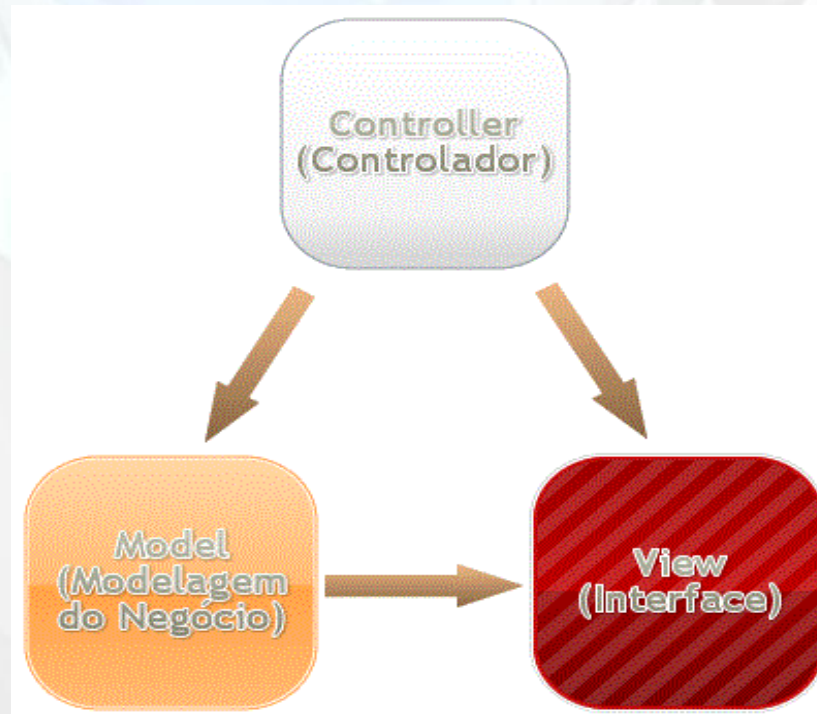
### ✓ Modelo 4 Camadas

- **Camada Cliente:** Neste caso o Cliente é o Navegador utilizado pelo usuário.
- **Camada de Apresentação:** Passa para o Servidor Web. A interface pode ser composta de páginas HTML, PHP, ou qualquer outra tecnologia capaz de gerar conteúdo para o Navegador.
- **Camada Lógica ou Regra de Negócios:** São as regras do negócio, as quais determinam de que maneira os dados serão utilizados. Esta camada está no Servidor de aplicações.
- **Camada de Persistência:** Nesta camada temos o servidor de Banco de dados, no qual reside toda a informação necessária para o funcionamento da aplicação.



# ARQUITETURA DE SISTEMAS WEB

## ✓ Padrão MVC



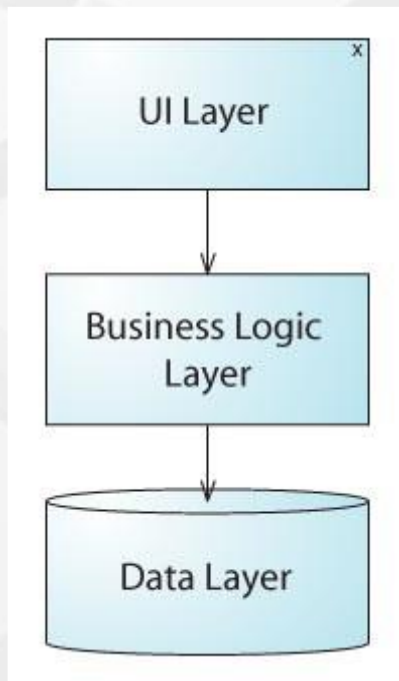
# ARQUITETURA DE SISTEMAS WEB

## ✓ Padrão MVC

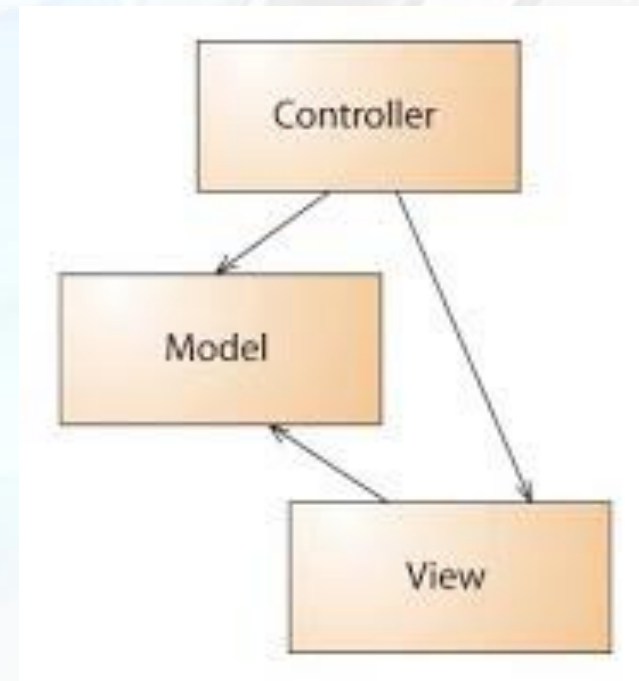
- O *MVC (model-view-controller)* é um padrão de arquitetura que tem por objetivo isolar a lógica do negócio da lógica de apresentação de uma aplicação.
- O padrão MVC divide a aplicação em três tipos de componentes:
  - **Modelo:** encapsula os dados e as funcionalidades da aplicação.
  - **Visão:** é a responsável pela exibição das informações, cujos dados são obtidos no modelo.
  - **Controlador:** recebe as requisições do usuário e aciona o modelo e/ou a visão.

# ARQUITETURA DE SISTEMAS WEB

## ✓ Padrão MVC vs Modelo 3 Camadas



Modelo 3 camadas



Padrão MVC



# ARQUITETURA DE SISTEMAS WEB

- ✓ Padrão MVC vs Modelo 3 Camadas
  - Modelo 3 Camadas
    - Um sistema de três camadas realmente é constituído por camadas (como um bolo).
  - MVC (Model View Controller)
    - O padrão MVC não mantém um sistema de camadas. O Controller acessa o Model e o View. O View, em seguida, acessa o Model.



**Universidade Federal de Santa Catarina**  
**Curso de Tecnologias da Informação e Comunicação**  
**Desenvolvimento de Sistemas Web**  
**Professor: Adriano de Oliveira**

---

# **APLICAÇÕES WEB**

# APLICAÇÕES WEB

## ✓ Sites e Aplicações Web

- Normalmente, as pessoas utilizam o termo site quando se referem a blogs, sites de notícias, sites institucionais, portais, lojas virtuais, entre outros.
- Já a denominação aplicação web é muito utilizada para sistemas de gestão empresarial que são acessados através de navegadores (browsers).

# APLICAÇÕES WEB

## ✓ Sites e Aplicações Web

- Não há uma definição exata que diferencie claramente sites e aplicações web. Alguns defendem que os sites são read-only (somente leitura) enquanto as aplicações web são read-write (leitura e escrita).
- Nessa definição, os sites apenas fornecem conteúdo enquanto as aplicações web podem fornecer e/ou receber conteúdo.



# APLICAÇÕES WEB

## ✓ Navegadores e Dispositivos

- As pessoas acessam sites e aplicações web através de navegadores (browsers) como Chrome, Firefox, Edge e Safari.
- Normalmente, esses navegadores possuem algumas diferenças na forma de exibir as páginas web aos usuários.
- Antigamente, essas diferenças eram maiores, mas com o passar do tempo, os navegadores ficaram cada vez mais parecidos nesse aspecto, contudo, os desenvolvedores web ainda devem tomar cuidado com essas diferenças.

# APLICAÇÕES WEB

## ✓ Web Servers e HTTP

- Os sites e as aplicações web são implantados em computadores conectados à Internet ou a uma rede privada qualquer (Intranet).
- Normalmente, os sites são implantados em computadores conectados à Internet pois assim poderão ser acessados praticamente de qualquer lugar do mundo.
- Por outro lado, as aplicações web, muitas vezes, são implantadas em computadores conectados a uma Intranet pois é comum ser necessário restringir o acesso externo à elas.

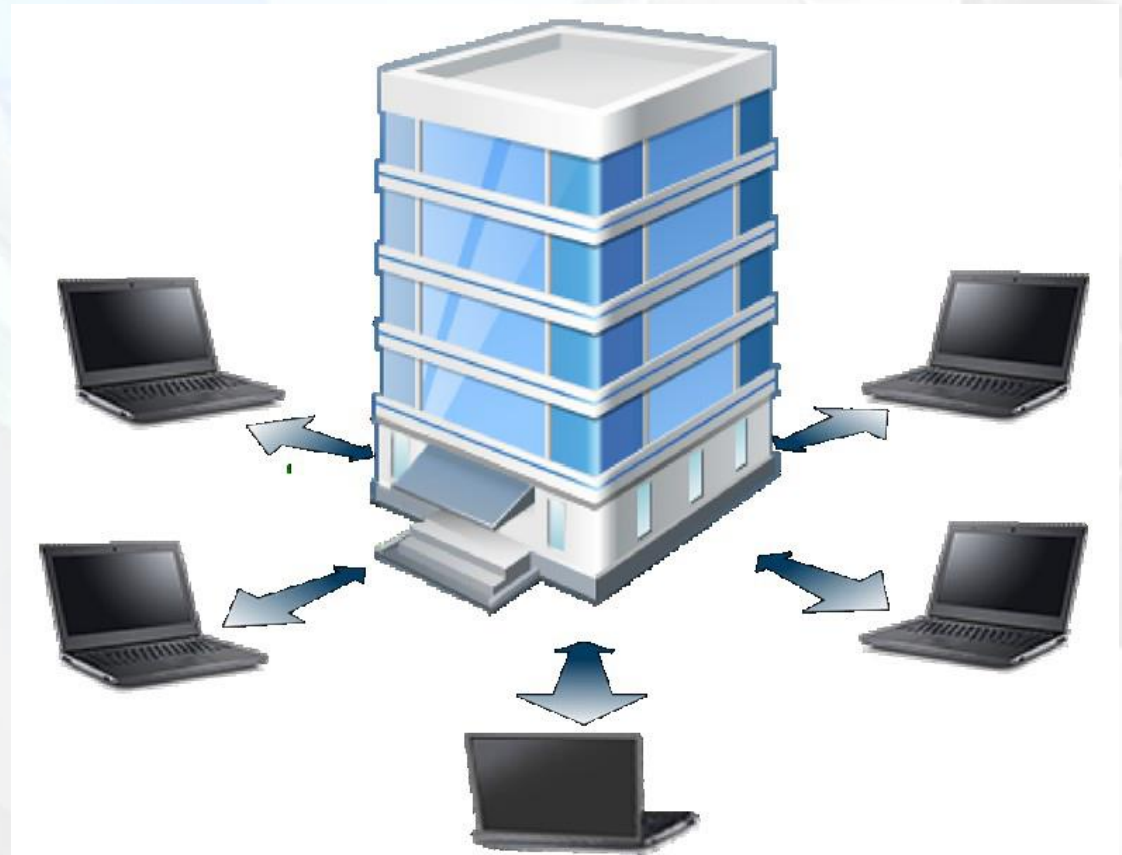
# APLICAÇÕES WEB

## ✓ Internet



# APLICAÇÕES WEB

## ✓ Intranet





# APLICAÇÕES WEB

## ✓ Requisição HTTP

- A comunicação entre navegadores e uma aplicação web é realizada por meio de requisições e respostas definidas pelo protocolo HTTP.
- HTTP é um protocolo que permite a obtenção de recursos na Web.
- As requisições são iniciadas pelo destinatário, geralmente um navegador da Web.

## APLICAÇÕES WEB

### ✓ URI's

- O protocolo HTTP permite a clientes e servidores interagir e trocar informações de uma maneira simples e confiável.
- O HTTP utiliza o URI (Uniform Resource Identifier) para identificar dados na Internet.
- Os URI's que especificam as localizações de documentos são chamados URL's (Uniform Resource Locators).
- URL's comuns fazem referência a arquivos, diretórios ou objetos e podem realizar tarefas complexas, como pesquisa em um banco de dados e pesquisas na Internet.

## APLICAÇÕES WEB

### ✓ URL's

- Se você conhecer o URL de um recurso ou arquivo público disponível em qualquer lugar da Web, você poderá acessá-lo por meio do HTTP. Detalhamento da url:

<http://www.sistema.br/pesquisa/agenda.html>

# APLICAÇÕES WEB

## ✓ URL's:

Componentes do HTTP: <http://www.sistema.br/pesquisa/agenda.html>

<a href="http://">http://</a>	indica que o recurso deve ser obtido utilizando o protocolo http.
<a href="http://www.sistema.br">www.sistema.br</a>	é o <i>hostname</i> totalmente qualificado do servidor ( <i>host</i> ) em que o recurso reside. O <i>hostname</i> é convertido em um endereço IP por servidor (DNS) que mantém um banco de dados dos nomes de <i>host</i> e seus endereços IP correspondentes.
<a href="http://www.sistema.br/pesquisa/agenda.html">/pesquisa/agenda.html</a>	o restante do URL especifica o nome do recurso solicitado e seu caminho ou localização no servidor Web.



## APLICAÇÕES WEB

### ✓ Fazendo uma solicitação e recebendo uma resposta

- Quando dado uma URL, um navegador Web realiza uma transação HTTP para recuperar e exibir a página Web nesse endereço.
- O navegador Web envia uma solicitação HTTP ao servidor.

# APLICAÇÕES WEB

## HTTP/1.1 Headers vs HTTP/2 Headers

```
GET / HTTP/1.1  
Host: www.devxperience.com.br  
HTTP/1.1 200 OK  
Content-Type: text/html; charset=utf-8  
Content-Language: pt-br Date: Thu, 01 Set 2017 00:00:00  
Cache-Control: private
```

189 bytes

```
?c?8)E8??%:?$
```

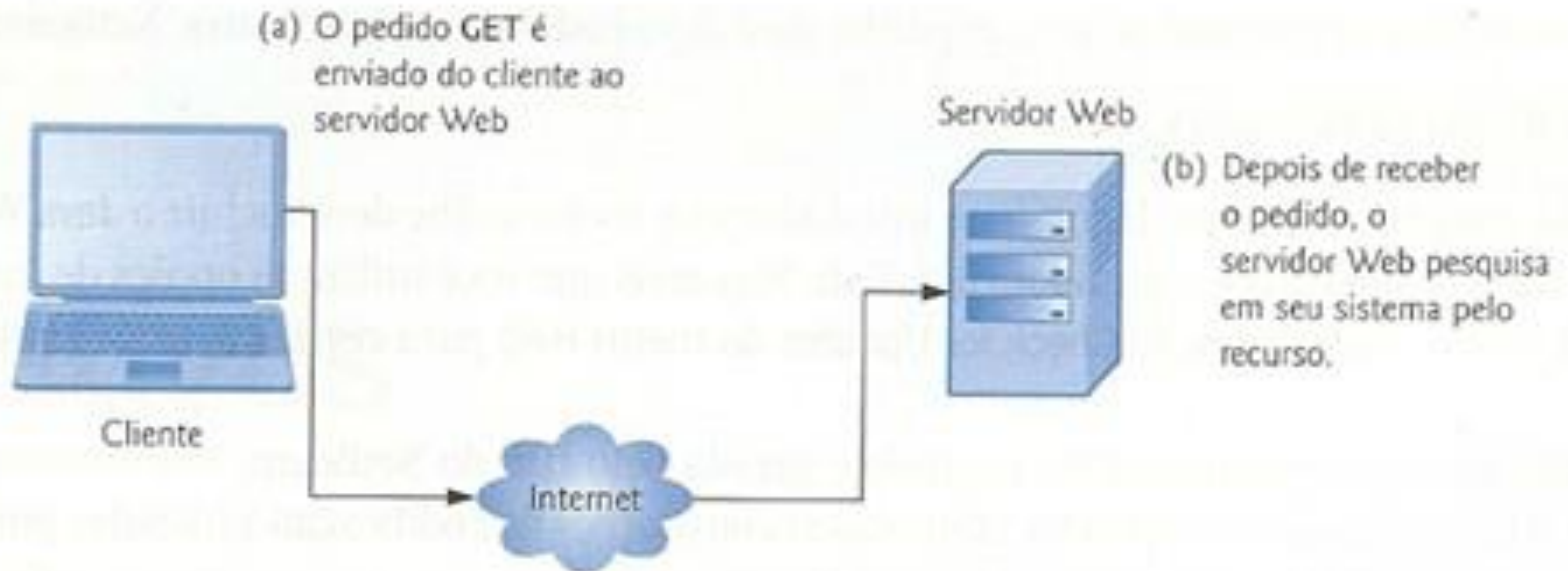
```
T?%.?JI?;1?R  
,1?T?%.?JI?;1?R  
?c?8)E8??%:?$
```

61 bytes

32% do  
conteúdo original

## APLICAÇÕES WEB

- ✓ Fazendo uma solicitação e recebendo uma resposta



# APLICAÇÕES WEB

## ✓ Resposta HTTP

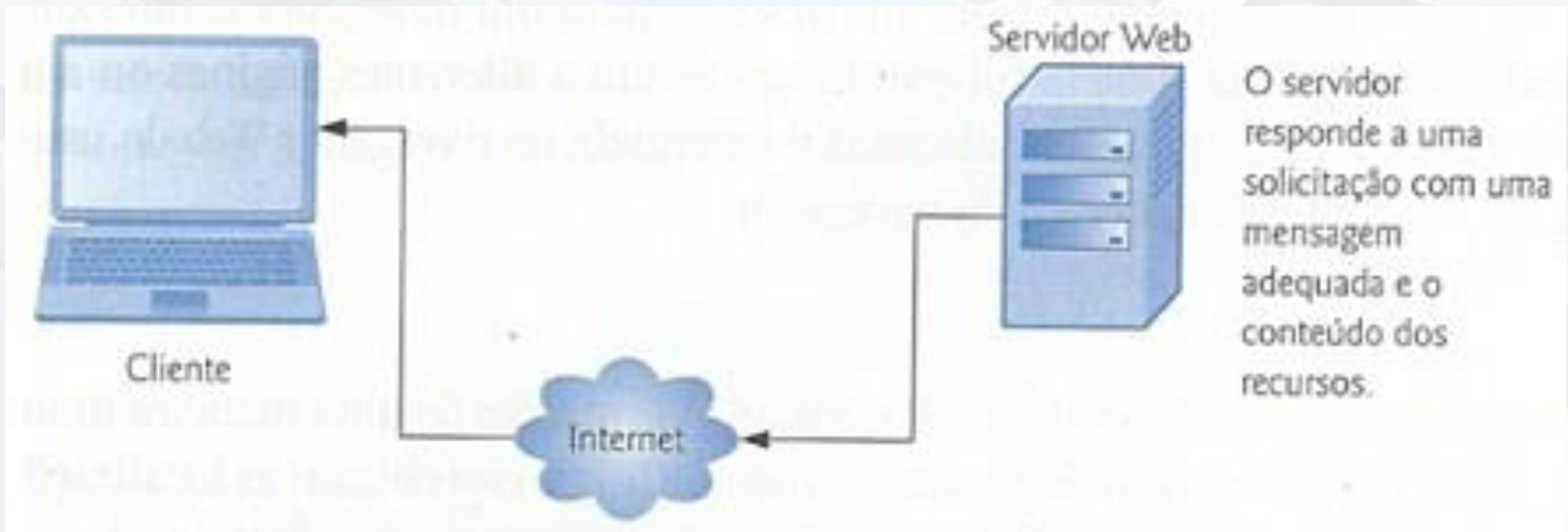
- Qualquer servidor que entende o protocolo HTTP pode converter essa solicitação e responder apropriadamente.
- O servidor primeiro responde enviando uma linha do texto indicando a versão HTTP, seguida por um código numérico e uma frase descrevendo o status da transação. Por exemplo:

***HTTP/2 200 OK indica sucesso***



# APLICAÇÕES WEB

## ✓ Resposta HTTP



## APLICAÇÕES WEB

### ✓ Cabeçalho HTTP

- O servidor envia um ou mais cabeçalhos HTTP que fornecem informações adicionais sobre os dados que serão enviados. No caso abaixo, o servidor está enviando um documento de texto XHTML, portanto, um cabeçalho HTTP seria lido:

*Content-type: text/html*

# APLICAÇÕES WEB

## ✓ Cabeçalho HTTP

- As informações deste cabeçalho especificam o tipo de Extensões Multi-função para Mensagens de Internet (sigla MIME do inglês Multipurpose Internet Mail Extensions) do conteúdo que o servidor está transmitindo ao navegador.
- O MIME é um padrão Internet que especifica formatos de dados para que os programas possam interpretar os dados corretamente.
- Por exemplo, o tipo MIME image/jpeg indica que o conteúdo é uma imagem JPEG. Quando o navegador recebe esse tipo de MIME ele tenta exibir a imagem.

## APLICAÇÕES WEB

### ✓ Solicitações HTTP Get e Post

- Os dois tipos de solicitação HTTP mais comuns são Get e Post (também conhecidos como métodos de solicitação). Uma solicitação Get, em geral, pede um recurso específico em um servidor.
- Os usos comuns das solicitações Get são recuperar uma imagem ou um documento HTML ou, ainda, buscar resultados de pesquisa com base de um termo de pesquisa submetido pelo usuário.
- Uma solicitação Get envia informações ao servidor no URL, por exemplo:  
*[www.google.com.br/search?q=ufsc](http://www.google.com.br/search?q=ufsc)*



## APLICAÇÕES WEB

### ✓ Solicitações HTTP Get e Post

*[www.google.com.br/search?q=ufsc](http://www.google.com.br/search?q=ufsc)*

- Nesse caso, search é o nome do handler de formulário no servidor Google, 'q' é o nome da variável no formulário de pesquisa do Google e 'ufsc' é o termo de pesquisa.
- O '?' separa a string de consulta do restante do URL em uma solicitação.
- Um par de nome/valor é passado para o servidor com o nome e o valor separados por um sinal de igual (=).

## APLICAÇÕES WEB

### ✓ Solicitações HTTP Get e Post

- Se mais de um par de nome/valor é for submetido, cada um é separado do próximo por caractere & (“e” comercial).
- O servidor utiliza os dados passados em uma string de consulta para recuperar um recurso HTML cujo atributo method é configurado como “get”, digitando o URL (possivelmente contendo uma string de consulta) diretamente na barra de endereços do navegador ou por meio de um hiperlink.

# APLICAÇÕES WEB

## ✓ Solicitações HTTP Get e Post

- Uma solicitação POST geralmente posta (envia) dados para o servidor, como dados de formulário ou documentos.
- A solicitação POST envia dados como parte de mensagem HTTP, não como parte do URL.
- Uma solicitação GET, em geral, limita a string de consulta (isto é, tudo à direita do '?') a um número específico de caracteres, assim, frequentemente é necessário enviar grandes volumes de informação utilizando o método POST.
- Este último também é às vezes preferido porque ele oculta os dados submetidos do usuário, incorporando-os a uma mensagem HTTP.

# APLICAÇÕES WEB

## ✓ Acesso simultâneo

- Além de todas as particularidades observadas no protocolo HTTP, na grande maioria dos casos, as aplicações Web devem ser acessadas por diversos usuários ao mesmo tempo.
- Consequentemente, os desenvolvedores Web devem criar ou utilizar algum mecanismo eficiente que permita esse tipo de acesso.



# APLICAÇÕES WEB

## ✓ Acesso simultâneo

- As páginas de uma aplicação Web devem ser geradas dinamicamente, por exemplo, quando um usuário de uma aplicação de e-mail acessa sua caixa de entrada, ele deseja ver todos os e-mails enviados até aquele momento.
- A página contendo a lista de e-mails deve ser gerada novamente todas as vezes que essa página for requisitada.

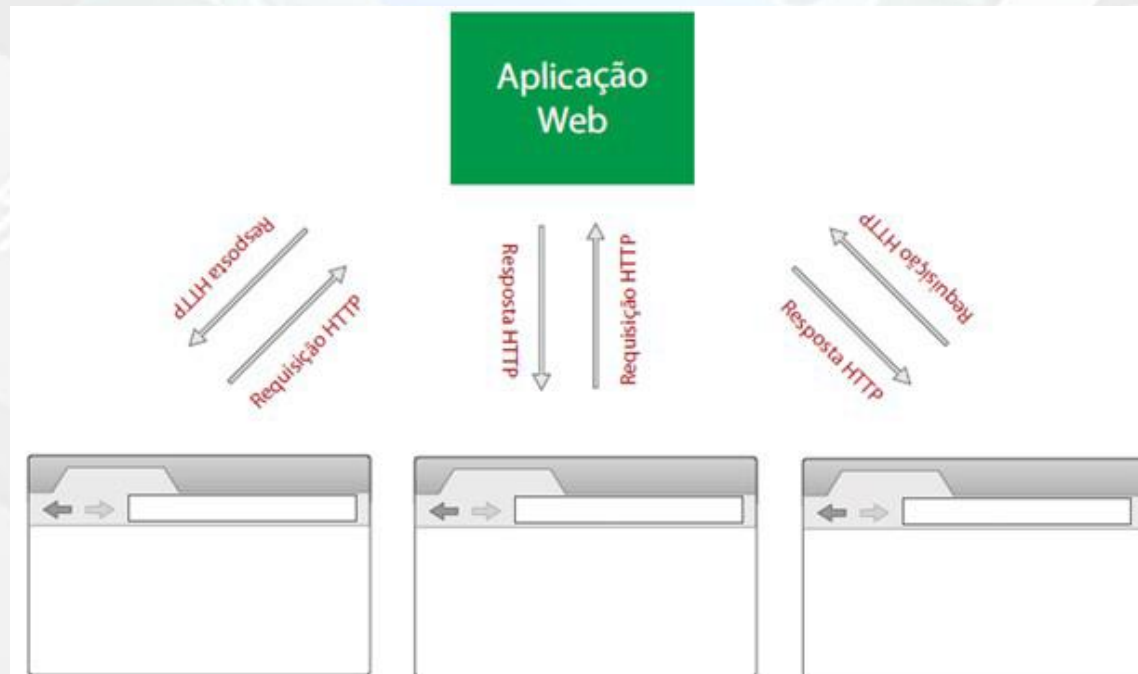
# APLICAÇÕES WEB

## ✓ Acesso simultâneo

- Logo, os desenvolvedores Web devem criar ou utilizar um mecanismo eficiente que permita que o conteúdo das páginas das aplicações Web seja gerado dinamicamente.

# APLICAÇÕES WEB

## ✓ Acesso simultâneo



# APLICAÇÕES WEB

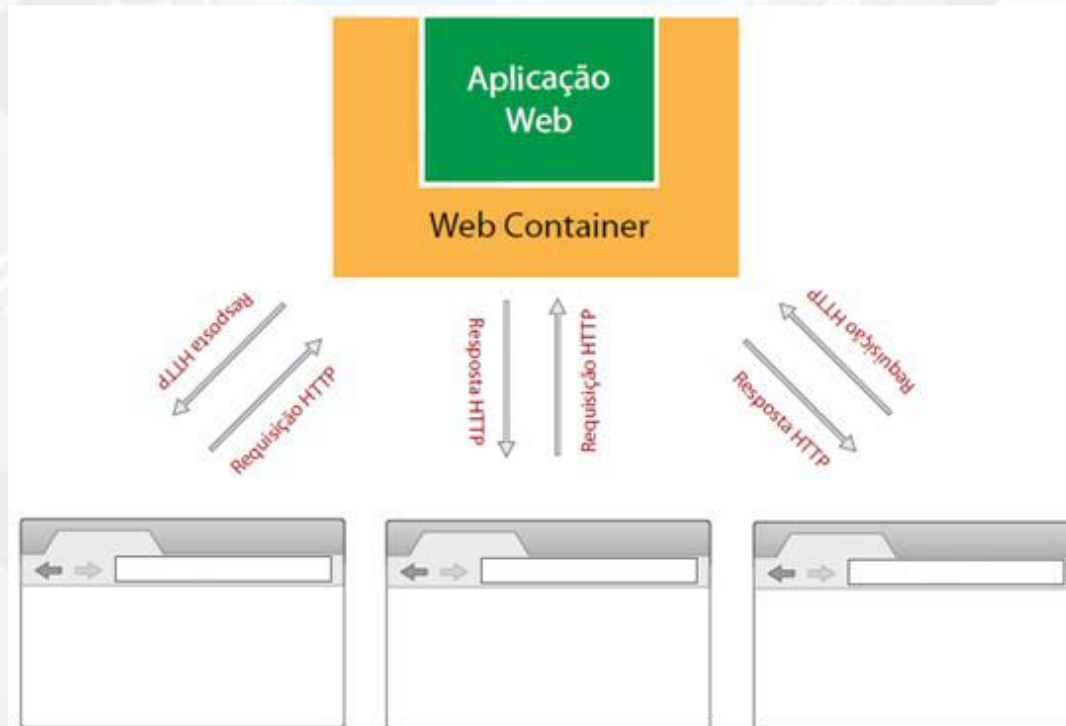
## ✓ Acesso simultâneo

- Solução: Resolver os três problemas apresentados (Manipulação de protocolo HTTP, Acesso simultâneo e conteúdo dinâmico) consumiria boa parte do tempo de desenvolvimento, além de exigir conhecimentos técnicos extremamente específicos por parte dos programadores.
- Para facilitar o desenvolvimento da aplicação Web, a plataforma Java oferece uma solução genérica que pode ser utilizada para desenvolver aplicações Web que você verá a seguir:



# APLICAÇÕES WEB

## ✓ Acesso simultâneo



# APLICAÇÕES WEB

## ✓ Acesso simultâneo

- Uma aplicação Web deve ser implantada em um Web Container para obter os recursos fundamentais que ela necessita. Um Web Container é responsável:
  - Pelo envio e recebimento de mensagens HTTP.
  - Por permitir que as aplicações sejam acessadas simultaneamente por vários usuários de uma maneira eficiente.
  - Por permitir que as páginas de uma aplicação Web sejam geradas dinamicamente.

# APLICAÇÕES WEB

## ✓ Servidores Web

- Apache HTTP Server
- Nginx
- IIS Express (Internet Information Services)
- Oracle HTTP Server