



# Desenvolvimento de Software para WEB

## Aula 2 - Fundamentos

Professor: Anderson Almada

---

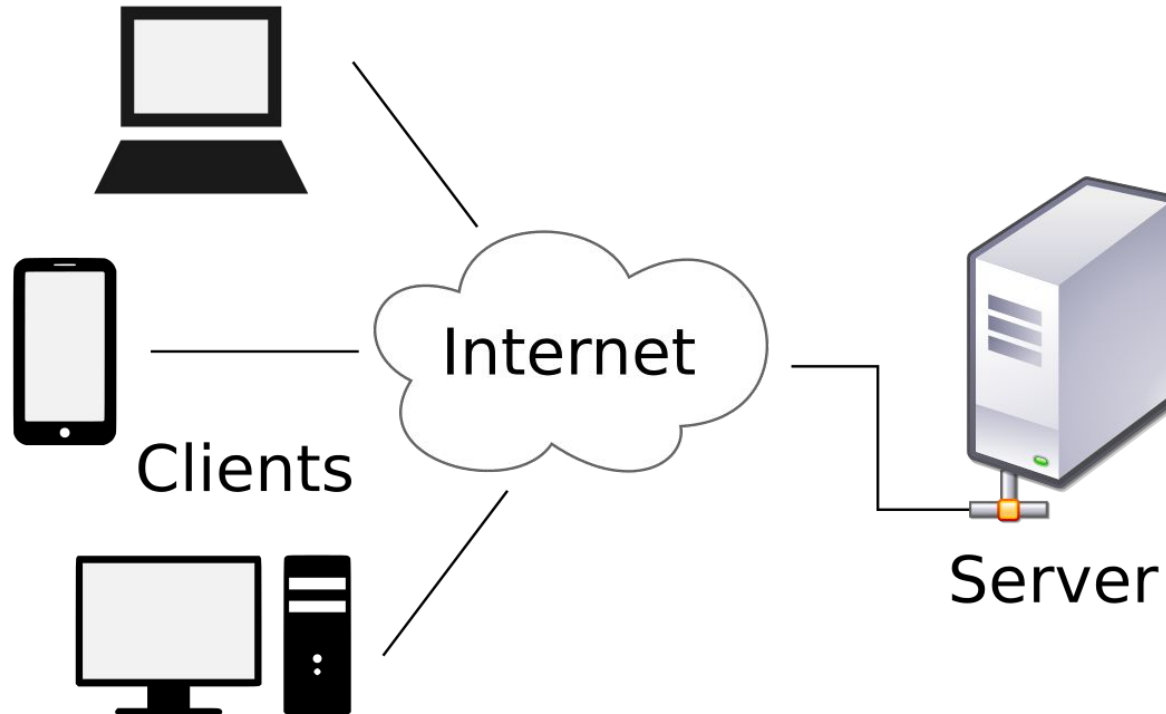
# Arquitetura??

# Arquitetura

- Arte de projetar e construir edifícios.
- Modo como estão interligadas as partes de um todo
- Forma, estrutura.



# Arquiteturas em Aplicações Cliente/Servidor



# Arquiteturas em Aplicações Cliente/Servidor

---

- **Servidor**
  - São equipamentos com maior poder de **processamento** e **armazenamento**.
  - **Recebe e responde** a solicitações dos Clientes;
  - Presta serviços **distribuídos**;
  - Atende a diversos Clientes **simultaneamente**;

# Arquiteturas em Aplicações Cliente/Servidor

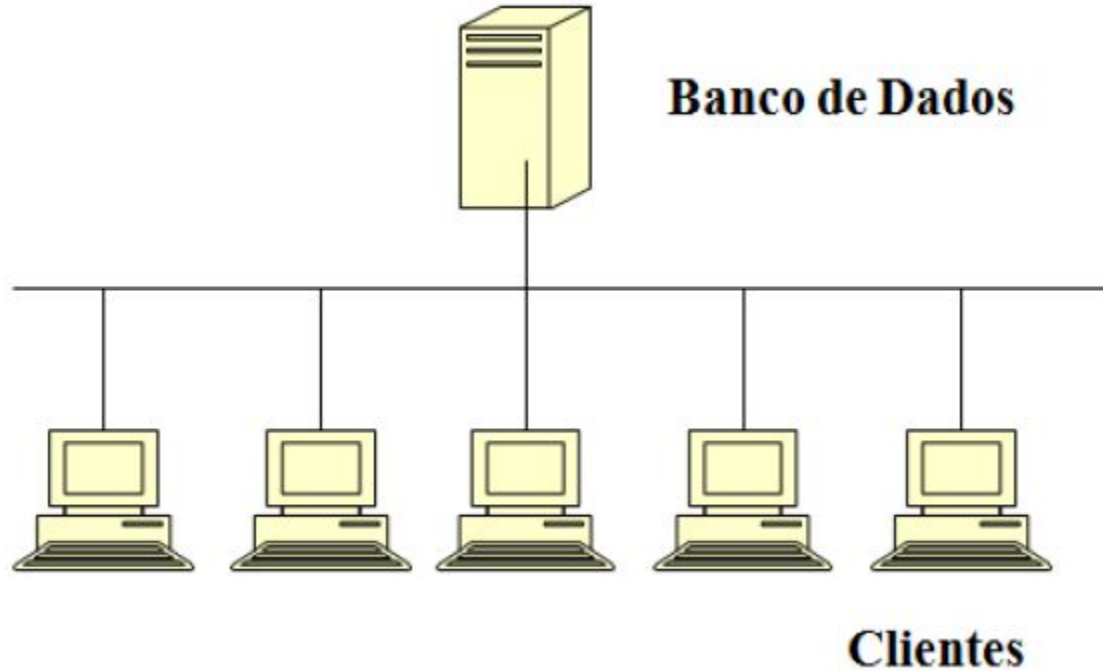
---

- **Cliente**

- Microcomputadores, smartphones, ... **ligados em rede**.
- **Inicia e termina** as conversações com os Servidores, solicitando **serviços**;
- **Não se comunica com outros Clientes**;
- Normalmente responsável por receber e enviar dados em uma comunicação com o usuário;
- Torna a rede **“transparente”** ao usuário;

# Arquitetura em duas camadas

---



# Arquitetura em duas camadas

---

- Foram utilizadas no início da implantação do modelo Cliente/Servidor.
- Aqui, a aplicação Cliente responde pelas seguintes funções:
  - Apresentação
  - Lógica do Negócio



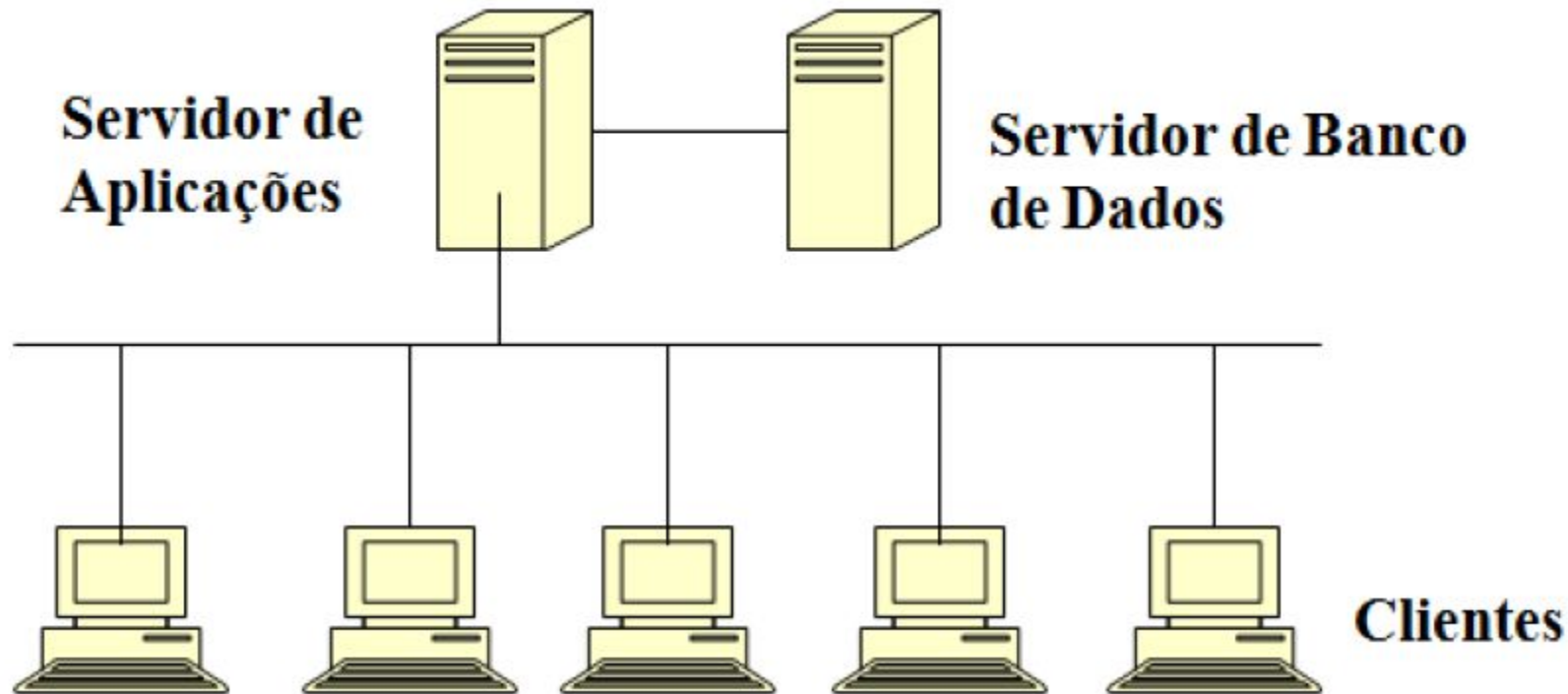
# Arquitetura em duas camadas

---

- **Características**

- É necessária uma nova versão do programa cada vez que alterações ocorreram e as regras mudarem.
- Também é necessário que todos os computadores que estejam com a versão anterior recebam a nova versão.

# Arquitetura em três camadas



# Arquitetura em três camadas

---

- **Características**

- Retira as Regras do Negócio do Cliente e as centralizam no Servidor de Aplicações:
  - O acesso ao Banco de Dados é feito através das regras contidas no Servidor de Aplicações.
  - Facilidade de atualização das regras de negócio, pois estão centralizadas em um único ponto.

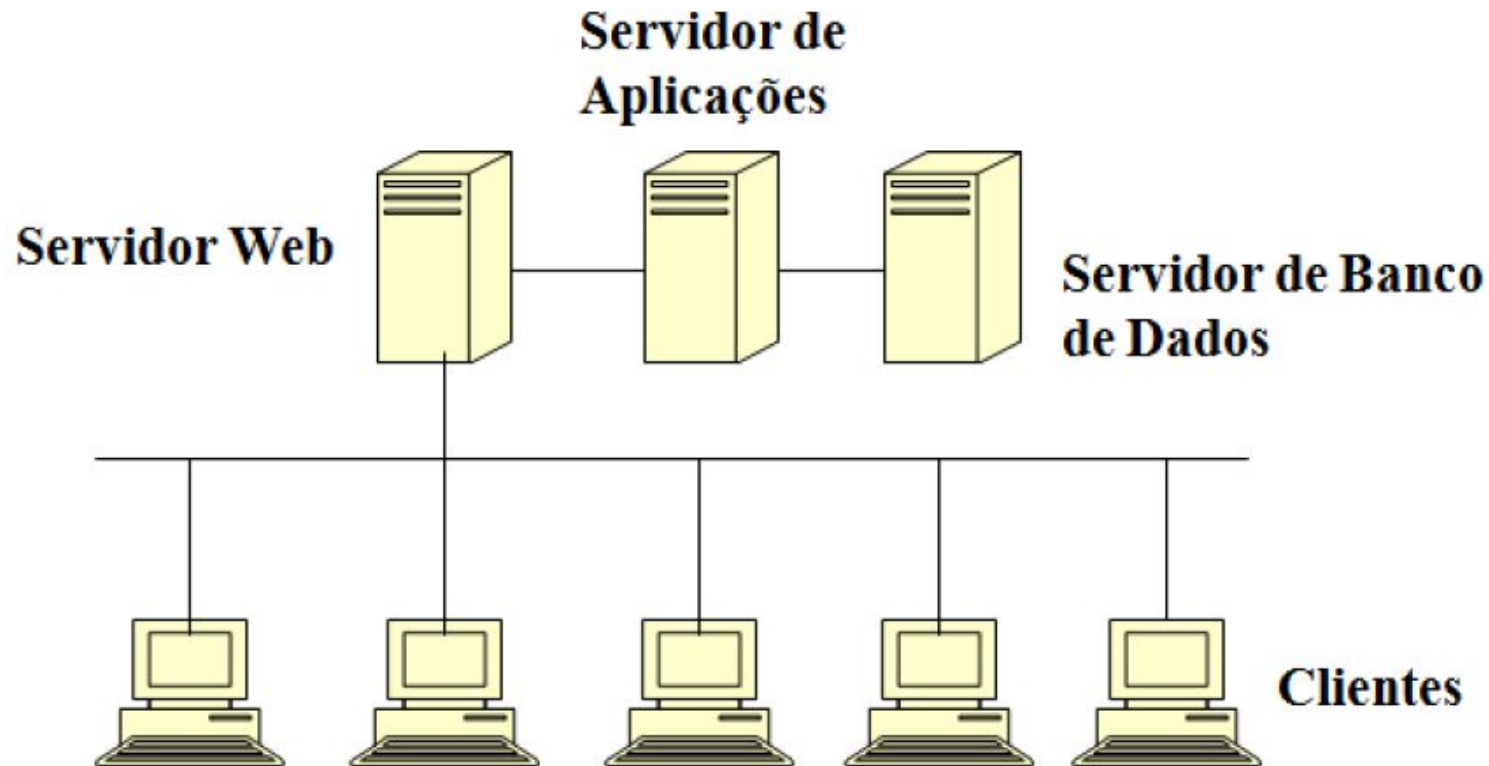
# Arquitetura em três camadas

---

- **Características**

- Todo o acesso do cliente ao Banco de Dados é feito de acordo com as regras contidas no Servidor de Aplicações.
  - Não é possível o acesso ao Banco de Dados diretamente, somente através dos recursos do Servidor de Aplicações;
- No Servidor de Banco de Dados reside toda informação necessária para o funcionamento da aplicação.

# Arquitetura em quatro camadas



# Arquitetura em quatro camadas

---

- **Baseado na WEB**

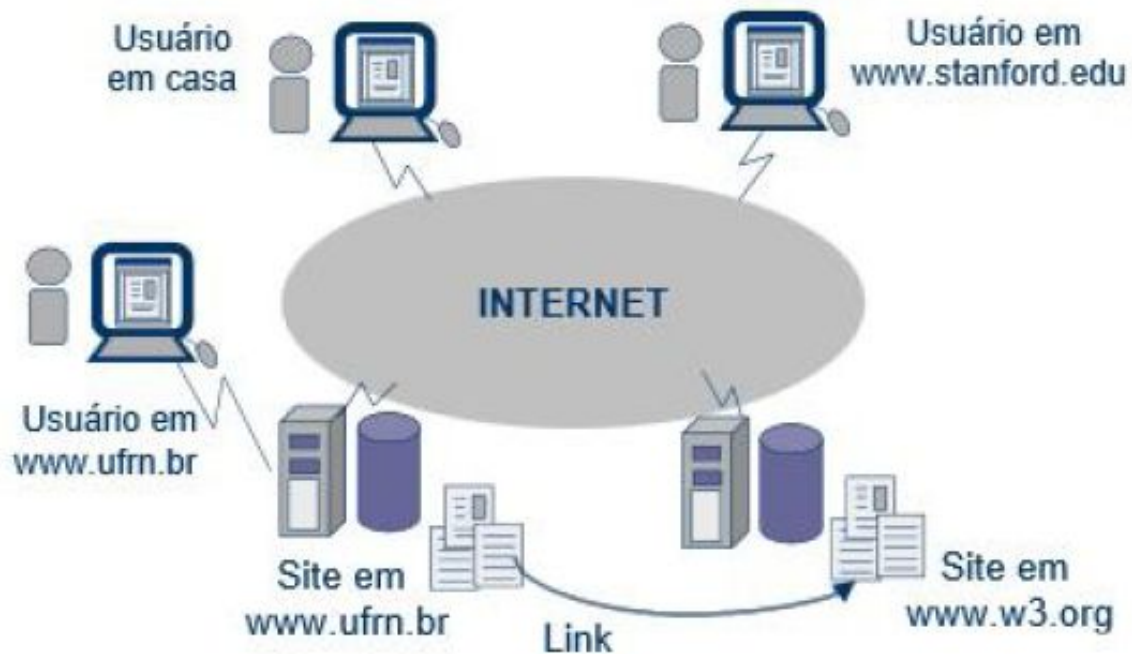
- Retirar a apresentação do cliente e centralizá-la em um determinado ponto, o qual na maioria dos casos é um Servidor WEB.
- O cliente deixa de existir como um programa que precisa ser instalado em cada computador.
- O acesso à aplicação é feito através de aplicativos “genéricos” chamados de **navegador**.

# Arquitetura em quatro camadas

---

- **Cliente:** Navegador
- **Apresentação:** Servidor Web, onde serão feitas as alterações de interface
- **Lógica (Regras do Negócio):** Servidor de Aplicações, onde serão feitas as alterações nas regras do negócio, quando necessárias.
- **Dados:** Servidor de Banco de Dados, com todas as informações

# Arquitetura básica da WEB





# Arquitetura básica da WEB

---

- A Web baseia-se em:
  - **Apresentação:** Softwares que interpretam conteúdos vindos da Internet, exibindo-os para o usuário.



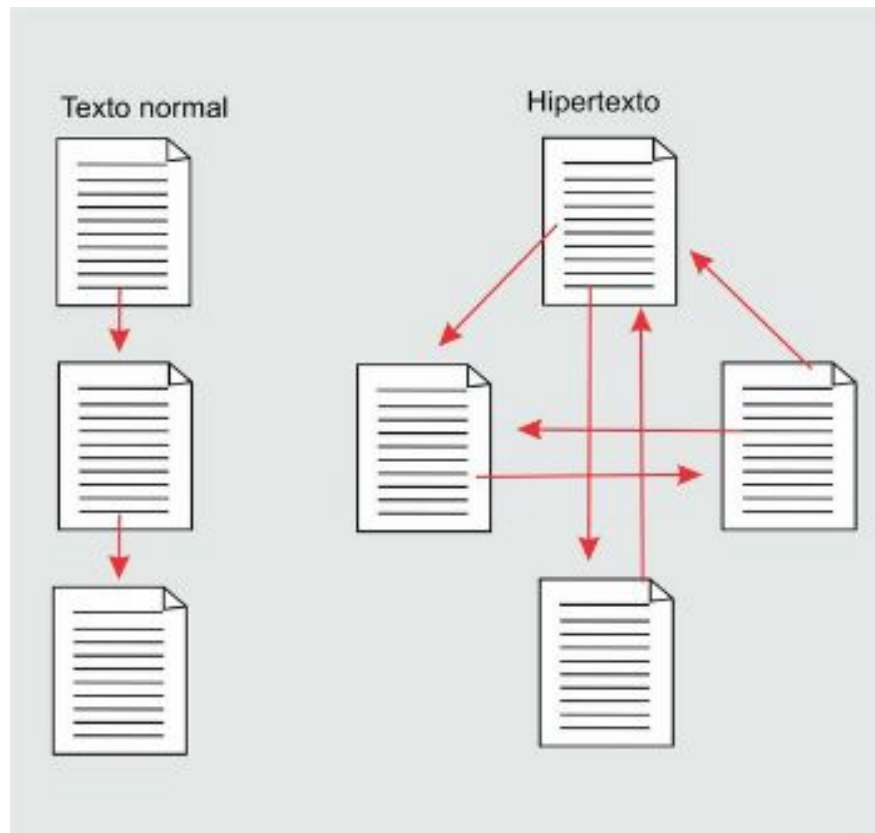
*JavaScript*

# Arquitetura básica da WEB

---

- A Web baseia-se em:
  - **Protocolo:** Convenção que controla e possibilita uma conexão, comunicação, transferência de dados entre dois sistemas computacionais.
    - **HTTP - Hypertext Transfer Protocol**
      - Um dos principais meios de comunicação entre um *browser* e um servidor Web.

# Arquitetura básica da WEB



# Arquitetura básica da WEB

---

- A Web baseia-se em:
  - A comunicação em HTTP é feita através de mensagens que podem ser requests/responses.
  - Request
    - Formadas e enviadas por um Cliente HTTP.
  - Response
    - Respostas são formadas e retornadas por um servidor após a requisição.

# Arquitetura básica da WEB

---

- A Web baseia-se em:
  - A comunicação em HTTP é feita através de mensagens que podem ser requests/responses.
  - Request (**navegador**)
    - Formadas e enviadas por um Cliente HTTP.
  - Response (**apache HTTP server**)
    - Respostas são formadas e retornadas por um servidor após a requisição.

# Arquitetura básica da WEB

---

- A Web baseia-se em:
  - **Serviços:**
    - **Protocolo IP:** Um dos mais importantes protocolos da arquitetura TCP/IP, ele permite a localização de um computador na internet, através de um conjunto de dígitos chamado de endereço IP.

# Arquitetura básica da WEB

---

- A Web baseia-se em:
  - **Serviços:**
    - **DNS:** Facilita o acesso a computadores da Internet e evita a necessidade de se decorar números IP;
      - Nesse serviço guarda-se tabelas que associam o nome de uma máquina à um endereço IP.
      - `www.cesar.org.br` > `200.255.126.219`

# Arquitetura básica da WEB

---

- A Web baseia-se em:
  - **Serviços:**
    - **Portas**
      - Maneira na qual é possível oferecer diversos serviços diferentes em uma mesma máquina.
      - As portas são identificadas por números
        - 80, 8080, 5433 e etc.

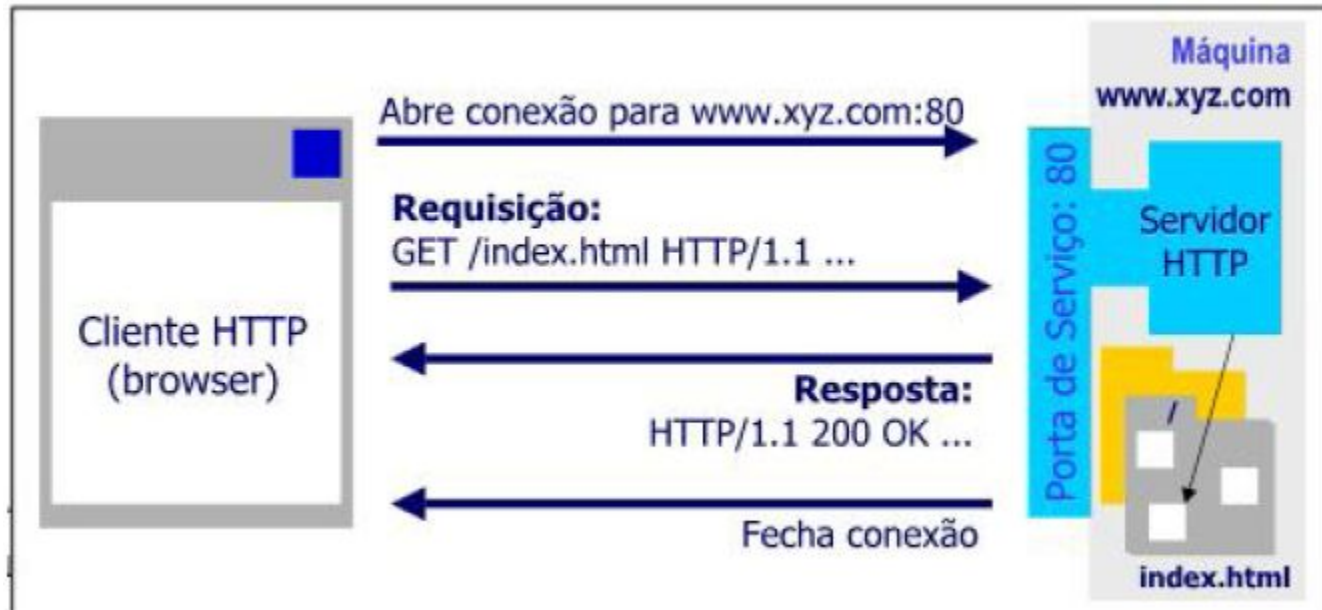


# Arquitetura básica da WEB

---

- A Web baseia-se em:
  - **Serviços:**
    - **Serviço Web**
      - É um serviço que provê acesso (HTTP) aos seus clientes para transferência de conteúdos.

# Arquitetura básica da WEB



# Arquitetura básica da WEB

---

- Alguns métodos básicos de Requisição HTTP
  - **GET**
  - **POST**
  - **PUT**
  - **DELETE**
  - HEAD
  - OPTIONS
  - PATCH

# Arquitetura básica da WEB

---

- Códigos de Status no HTTP
  - Códigos que informam ao cliente HTTP o status do servidor após a requisição.

<b>Código</b>	<b>Categoria</b>	<b>Informação</b>
<b>1xx</b>	Informativo	Isso significa que a solicitação foi recebida e o processo continua.
<b>2xx</b>	Sucesso	Isso significa que a ação foi recebida, entendida e aceita com sucesso
<b>3xx</b>	Redirecionamento	Isso significa que outras ações devem ser tomadas para concluir a solicitação.
<b>4xx</b>	Erro do cliente	Significa que a solicitação contém sintaxe incorreta ou não pode ser preenchida
<b>5xx</b>	Erro do servidor	Isso significa que o servidor não conseguiu atender a uma solicitação aparentemente válida.

# Arquitetura básica da WEB

---

- Códigos de Status no HTTP
  - **200 - OK** - A requisição foi completada com sucesso
  - **400 - Bad Request** - O pedido não pôde ser entregue devido a sintaxe incorreta.
  - **401 - Unauthorized** - Quando a autenticação é possível, mas não conseguiu ou ainda não foram fornecidos.
  - **404 - Not Found** - O recurso requisitado não foi encontrado.
  - **500 - Internal Error** - O servidor encontrou uma condição inesperada.

# Arquitetura básica da WEB

---

- **Cookies**
  - Grupo de dados trocados entre o servidor de páginas e o navegador colocado em um ficheiro criado no computador do usuário.

# Arquitetura básica da WEB

- Cookies - Exemplos

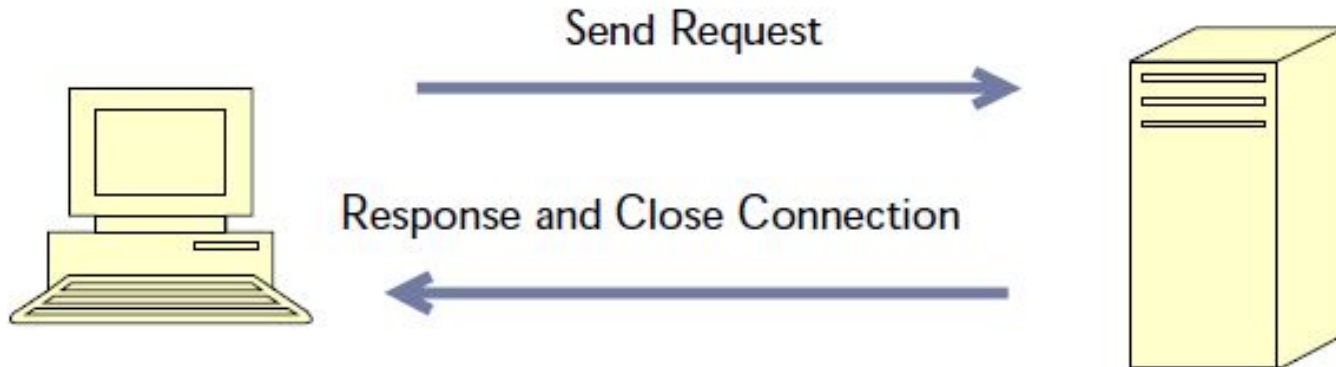
Domain	Path	Content	Expires	Secure
<u>toms-casino.com</u>	/	<u>CustomerID=497793521</u>	15-10-02 17:00	Yes
<u>joes-store.com</u>	/	Cart=1-00501;1-07031;2-13721	11-10-02 14:22	No
<u>aportal.com</u>	/	<u>Prefs=Stk:SUNW+ORCL;Spt:Jet</u> <u>s</u>	31-12-10 23:59	No
<u>sneaky.com</u>	/	<u>UserID=3627239101</u>	31-12-12 23:59	No



# Arquitetura básica da WEB

- **Sessões**

- Armazena as informações do lado servidor
- O protocolo HTTP usa conexões stateless



# Arquitetura básica da WEB

---

- **Sessões**

- Sessões permitem usuários manterem estado entre várias conexões.
- O servidor gera um **session ID** que é enviado para o cliente na primeira resposta, assim o Cliente reenvia esse session ID sempre que for fazer uma nova requisição.

# Arquitetura básica da WEB

---

- **URL - Uniform Resource Locator**

- Todas as comunicações na plataforma Web utilizam a sintaxe chamada URL para localizar os recursos que são transferidos.
- Contém duas informações essenciais:
  - **COMO** transferir o objeto;
  - **ONDE** encontrá-lo.

# Arquitetura básica da WEB

---

- **URL - Uniform Resource Locator**

protocolo://maquina:porta/recurso

<http://www.dominio.com.br/texto.html>

# Arquitetura básica da WEB

---

- **URL - Uniform Resource Locator**

**protocolo**://maquina:porta/recurso

**http**://www.dominio.com.br/texto.html

# Arquitetura básica da WEB

---

- **URL - Uniform Resource Locator**

protocolo://**maquina**:porta/recurso

http://**www.dominio.com.br**/texto.html

# Arquitetura básica da WEB

---

- **URL - Uniform Resource Locator**

protocolo://maquina:**porta**/recurso

http://www.dominio.com.br/texto.html

# Arquitetura básica da WEB

---

- **URL - Uniform Resource Locator**

protocolo://maquina:porta/**recurso**

http://www.dominio.com.br/**texto.html**



# Arquitetura básica da WEB

---

- **MIME - Multipart Internet Mail Extensions**
  - Tipo de dado recebido pelo *browser* para representação da informação na tela.
    - Imagem;
    - Arquivo HTML;
    - Texto em formato JSON

# Arquitetura básica da WEB

---

- **MIME - Multipart Internet Mail Extensions**

- Formato: Tipo/Subtipo

- Para imagens

- **image/jpg**

- Para texto

- **text/plain**

- Para página html

- **text/html**

\* <https://www.freeformatter.com/mime-types-list.html>

\* [https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Basico\\_sobre\\_HTTP/MIME\\_types](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Basico_sobre_HTTP/MIME_types)



Struts<sup>2</sup>



JavaScript

ExtJS



# Tecnologias do Ambiente Web



APACHE  
HTTP SERVER



Microsoft  
.net



symfony



ASP.NET



# Conceitos básicos - Frond-End

---

- **Tecnologias de Apresentação**

- Se destinam à formatação e estruturação das páginas Web;
- Podem ser usadas também para construir a interface de aplicações Web no browser.



# Conceitos básicos - Front-End

---

- **Tecnologias Interativas**

- Permitem o desenvolvimento de aplicações e páginas com um alto nível de interatividade com o usuário.
- Podem ser executadas:
  - Lado do Cliente
    - Javascript (JS)
      - JQuery

# Conceitos básicos - Frond-End

---

- **Tecnologias Interativas**
  - Podem ser executadas:
    - Lado do Servidor
      - Java
      - PHP
      - Spring

# Conceitos básicos - Frond-End

---

- **Servidor Web**

- Modelo de Entrega de Páginas:

- **Estáticos**

- Conteúdo vem diretamente de um ficheiro no servidor;
      - Não utiliza scripts para geração de conteúdos.

- **Dinâmicos**

- Conteúdo é criado dinamicamente por scripts do lado do servidor para então ser retornado para o cliente.





# Dúvidas??

E-mail: [almada@crateus.ufc.br](mailto:almada@crateus.ufc.br)