

Programação Web II

Internet, Sistemas Web e Arquitetura Cliente x Servidor

Prof. Ricardo Ramos de Oliveira

`ricardo.ramos@ifsuldeminas.edu.br`



Instituto Federal do Sul de Minas Gerais

Campus Poços de Caldas

Sul de Minas Gerais

Campus
Poços de Caldas

Curso Desenvolvimento de Sistemas

3º Módulo

10 de Fevereiro de 2022

Sumário

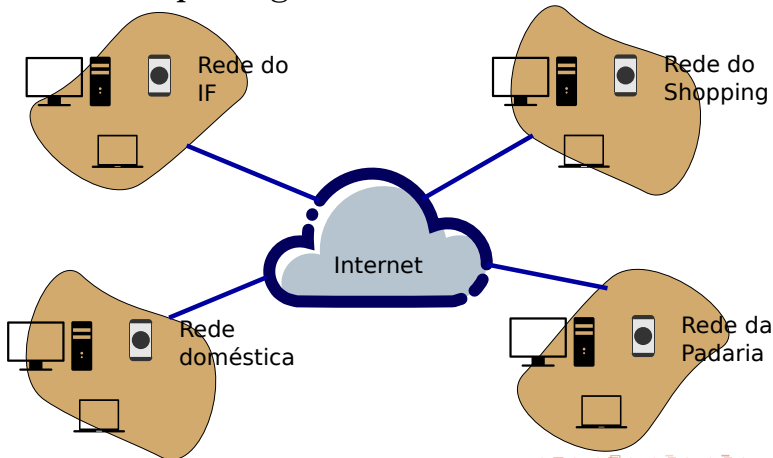
- 1 A Internet
- 2 Web
- 3 Arquitetura cliente-servidor
- 4 SOA e Web Services

A Internet

A Internet é a rede mundial de computadores. Uma rede que engloba muitas outras redes.

A Internet

A Internet é a rede mundial de computadores. Uma rede que engloba muitas outras redes.

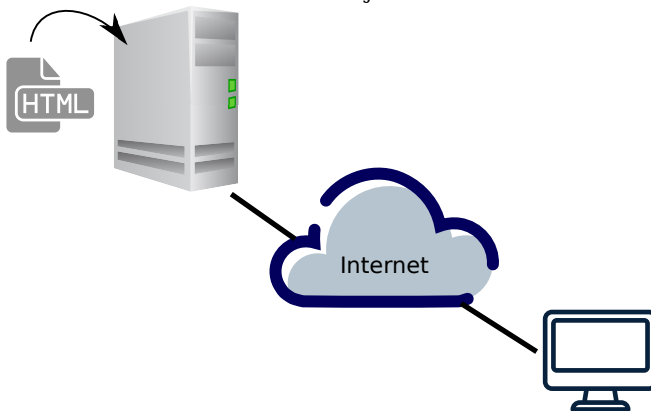


A Internet e a Web

A Internet é a infraestrutura para a *World Wide Web*.

Web: Sistema em Evolução

A Web é um sistema em evolução para a publicação e acesso a recursos e serviços através de Internet.



O acesso à Web

A maneira mais comum de acessar a Web é por meio dos navegadores Web (*browsers*).



Onde, quando e porquê?

- A Web nasceu no CERN (Centro Europeu de Pesquisa Nuclear) em 1989.
- Era usada para a troca de documentos entre a comunidade de físicos conectados pela Internet.

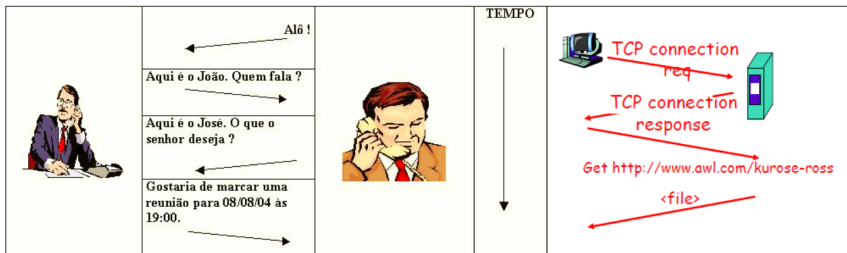
Componentes

- HTML, CSS e JavaScript;
- Protocolo HTTP;
- URLs;
- Arquitetura de sistema cliente-servidor;
- Web Services.

Protocolo

Protocolo

Conjunto de regras sobre o modo de como se dará a comunicação entre as partes envolvidas.



Protocolo HTTP

Protocolo HTTP

- O HTTP *Hypertext Transfer Protocol*, significa Protocolo de Transferência de Hipertexto;
- É um protocolo de comunicação utilizado nas aplicações Web;
- O HTTP funciona como um protocolo de requisição-resposta na arquitetura cliente-servidor.

URLs

Um URL (*Uniform Resource Locator*) tem como objetivo identificar um recurso e seguem a seguinte forma:
esquema:identificador-específico-do-esquema

URLs HTTP

O esquema amplamente usado para acessar recursos na Web é o HTTP. Os URLs HTTP são da seguinte forma:

```
http://nome-do-servidor[:porta][/nome-de-caminho][?consulta][#fragmento]
```

Os itens em colchetes são opcionais.

Cliente-Servidor

- Faça uma página HTML;
- Imagine que sua página seja um recurso que você deseja compartilhar com outras pessoas;
- Abra o terminal no linux;
- Navegue pelo terminal até o diretório em que está o seu arquivo `.html`;
- Digite o comando: `php -S 0.0.0.0:8000`
- Verifique qual é o seu ip pelo comando: `ifconfig`;
- Peça para o seu colega visitar a sua página digitando corretamente o URL HTTP a seguir em qualquer navegador Web:

`http://ip:8000/arquivo.html`

Cliente-Servidor

- Ao disponibilizar sua página na rede você estará compartilhando um recurso ou provendo serviços com várias outras pessoas;
- Todos que desejam acessar seu recurso/serviços fazem o papel de **clientes**;
- Você que disponibiliza um recurso/serviço faz o papel de um **servidor**.

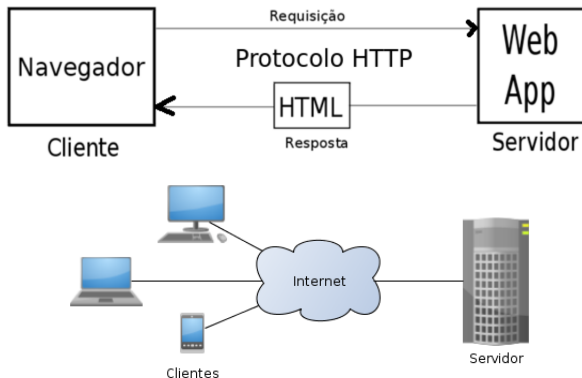
Cliente-Servidor

Em uma arquitetura cliente-servidor:

- Servidor é aquele que fornece recursos ou serviços;
- Cliente é aquele que requer os recursos ou serviços.

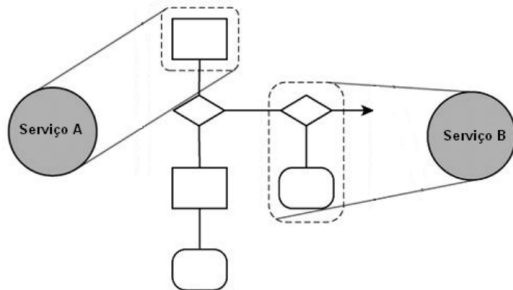
Cliente-Servidor

Arquitetura Cliente Servidor



SOA

- A Arquitetura Orientada a Serviços ou *Service-Oriented Architecture* (SOA) é um conjunto de funcionalidades bem definidas em forma de serviços disponibilizados através da rede;



SOA

- A Arquitetura SOA baseia-se em duas entidades: **consumidores** e **provedores de serviços**;
- Tais entidades estão relacionadas com conceitos fundamentais de troca de mensagens entre servidores e clientes em um sistema web distribuído;

SOA

- Um provedor possui características semelhantes a um servidor que disponibiliza serviços na rede, enquanto consumidores são considerados clientes que utilizam os serviços disponíveis por provedores de serviços;
- Uma entidade também pode assumir ambos os papéis ao mesmo tempo, caracterizando a composição de serviços.

- A Figura a seguir ilustra um programa do consumidor de serviço enviando, pela rede, uma requisição ao provedor de serviço, o qual retorna, pela rede, a resposta àquela requisição para o devido consumidor de serviço.



Figura: Interação Provedor-Consumidor

Web Services

- Dentre as diferentes explicações sobre Web services presentes na literatura, o World Wide Web Consortium (W3C) apresenta uma definição mais completa e amplamente aceita na área da computação:
- "Um Web service é um sistema de software projetado para suportar interações máquina-para-máquina interoperáveis pela rede. Tipicamente transportado utilizando o protocolo HTTP com uma serialização XML/J-SOON em conjunto com outros padrões relacionados com a Web"

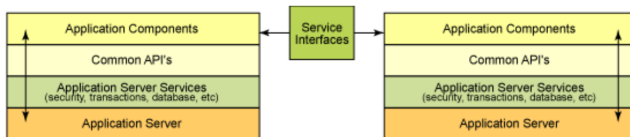
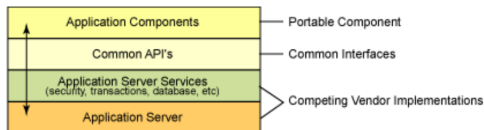
Web Services

- Ao contrário das aplicações Web convencionais no qual foram projetados para suportar interações aplicação-usuário (B2C), a tecnologia Web services foi desenvolvida para realizar interações aplicação-aplicação (B2B), embora também possa ser utilizada para interações com o usuário, pois operações com Web services podem ser realizadas através do navegador;
- Todavia, como está descrito na definição da W3C, seu foco é a interação aplicação-aplicação;
- O seu principal objetivo é solucionar o problema de integração de aplicativos heterogêneos (interoperabilidade).

Interoperabilidade

- Uma aplicação portátil não é necessariamente interoperável.
- Igualmente assim, uma aplicação interoperável não é necessariamente portátil.

Interoperabilidade



Web Services

Web Service



JSON

- O *JavaScript Object Notation* (JSON) é um formato de intercâmbio de dados, ou seja, basicamente um conjunto de chaves e valores, que podem ser interpretados por qualquer linguagem. (Independente de Linguagem de Programação);
- JSON é um formato de dados baseado em texto seguindo a sintaxe de objeto literais do JavaScript;
- É comumente usado para transmitir dados em aplicativos da Web (por exemplo, enviar alguns dados do servidor para o cliente, para que possam ser exibidos em uma página da Web ou vice-versa)

JSON

- A capacidade de armazenar dados e representar as estruturas de dados gerais como: registros, listas e árvores;
- A sintaxe JSON é significativamente mais simples, então o processamento é mais eficiente em comparação com o XML.

JSON

Exemplo de um Arquivo em JSON

```
1 {  
2   "autor": "George Orwell",  
3   "titulo": "1984",  
4   "id": 42  
5 }
```

XML

- O *eXtensible Markup Language* (XML) é uma linguagem genérica e padronizada de marcação, isto é, uma linguagem capaz de descrever uma organização lógica, estruturada dos dados através de tags definidas;
- O XML e o JSON são formatos populares e amplamente aceitos, eles são a base principal para o desenvolvimento dos web services;
- Portanto, esta é a diferença mais significativa entre os dois formatos no intercâmbio de dados.

XML

Exemplo de um Arquivo em XML

```
1 <livro>
2   <autor>Orwell</autor>
3   <titulo>1984</titulo>
4   <id>42</id>
5 </livro>
```

REST vs SOAP

- O *Representational State Transfer* ou REST é um estilo arquitetural proposto por Roy Thomas Fielding no ano de 2000;
- Surgiu como uma forma de simplificar o desenvolvimento de Serviços Web RESTful;
- O *Simple Object Access Protocol* (SOAP) ou WS-* possui uma complexidade, devido à grande quantidade de padronização (muitas camadas) e ao formato de mensagem SOAP e a linguagem de definição de interface (WSDL-XML).

REST vs SOAP

- *Web Services Description Language (WDSL)*;
- O SOAP possui o funcionamento muito diferente do REST;
- REST é muito mais simples de trabalhar do que o SOAP.

Referências

- 1 OLIVEIRA, Ricardo Ramos de. **Avaliação de manutenibilidade entre as abordagens de web services RESTful e SOAP-WSDL**. 2012. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2012. Acesso em: 2020-11-07.

Referências

