

Comunicação Cliente e Servidor com Persistência em Banco de Dados Relacional

Aula experimental

Aramis H.
PUCPR - 26/06/2024

Apresentação

Contato:

aramis.moraes@pucpr.edu.br



Agenda

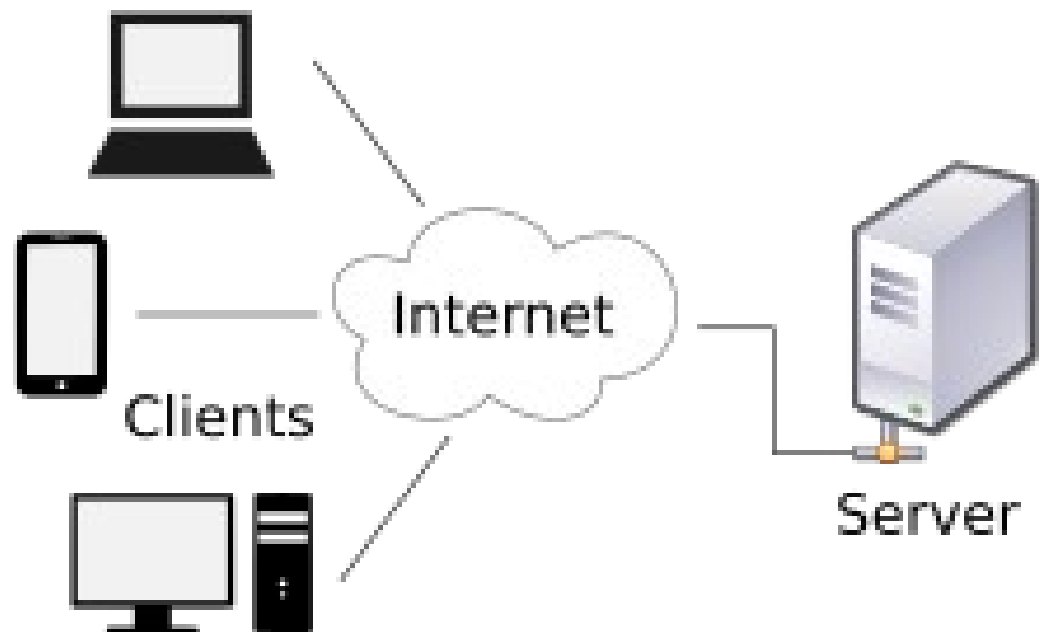
- Conceitos
- Arquiteturas
- Protocolo
- HTML/CSS/Javascript
- Banco de dados relacional

Conceitos - Cliente-Servidor

- Onde existe esse modelo?
- Qual foi seu contato com ele hoje?
- Quais exemplos desse sistema?

Conceitos - Cliente-Servidor

- Cliente
 - Navegadores Web, Aplicativos Móveis
- Servidor
 - *Web Servers*



Tipos de Arquitetura – Cliente-Servidor

- Arquitetura **Monolítica**
 - Estrutura mais simples de desenvolver e implantar, principalmente para projetos pequenos

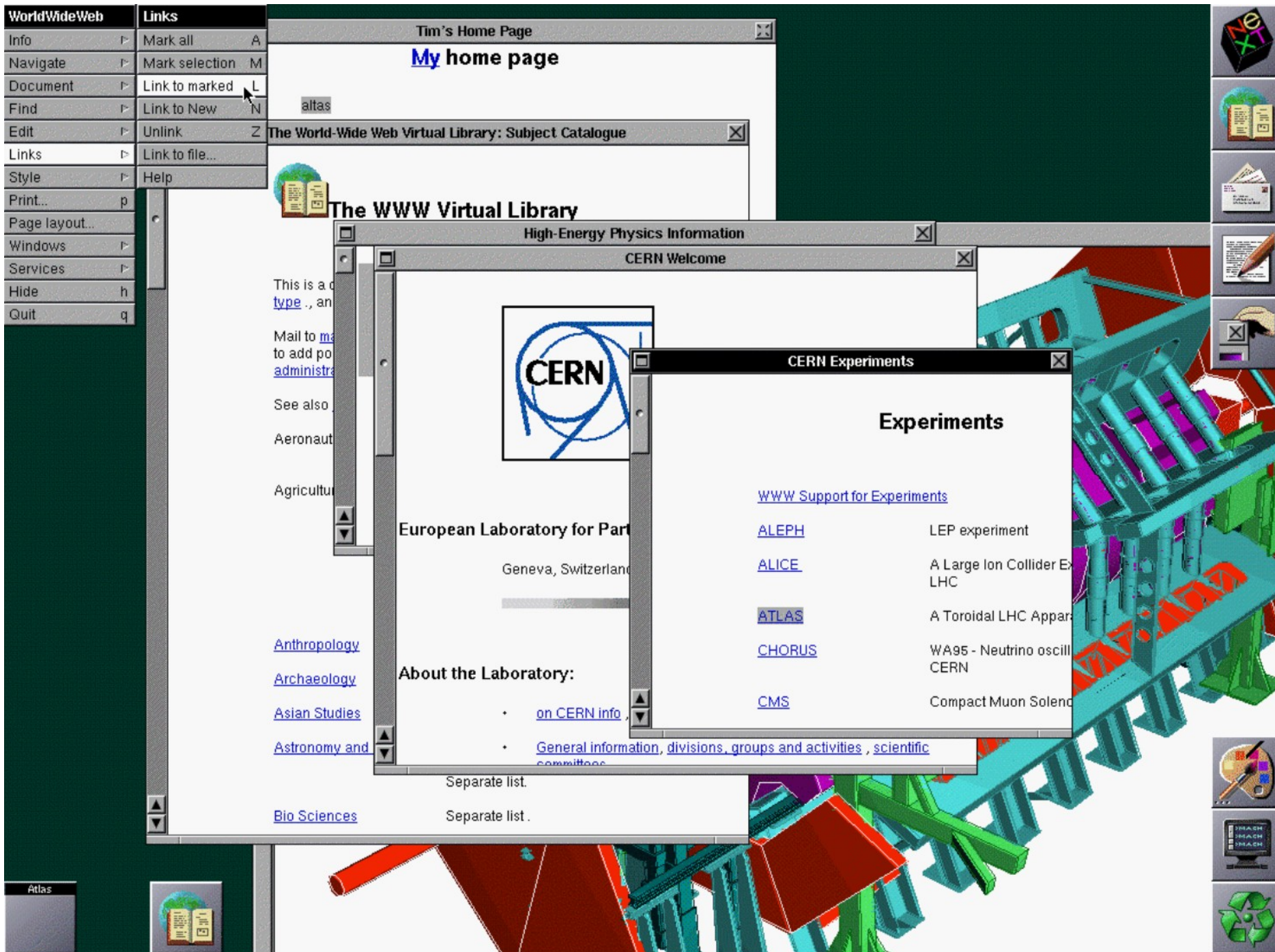
Tipos de Arquitetura - Cliente-Servidor

- Arquitetura **Microserviços**
 - Serviços individuais podem ser escalados independentemente.
 - Referência ao **Serverless** (containers)

Comunicação - Cliente-Servidor

- HTTP/HTTPS
 - 89 bis 91 spec
 - 90 browser
 - 1.0 – 96
 - 1.1 – 97
- **websockets**





Prática - Webserver

- Webserver simples em python

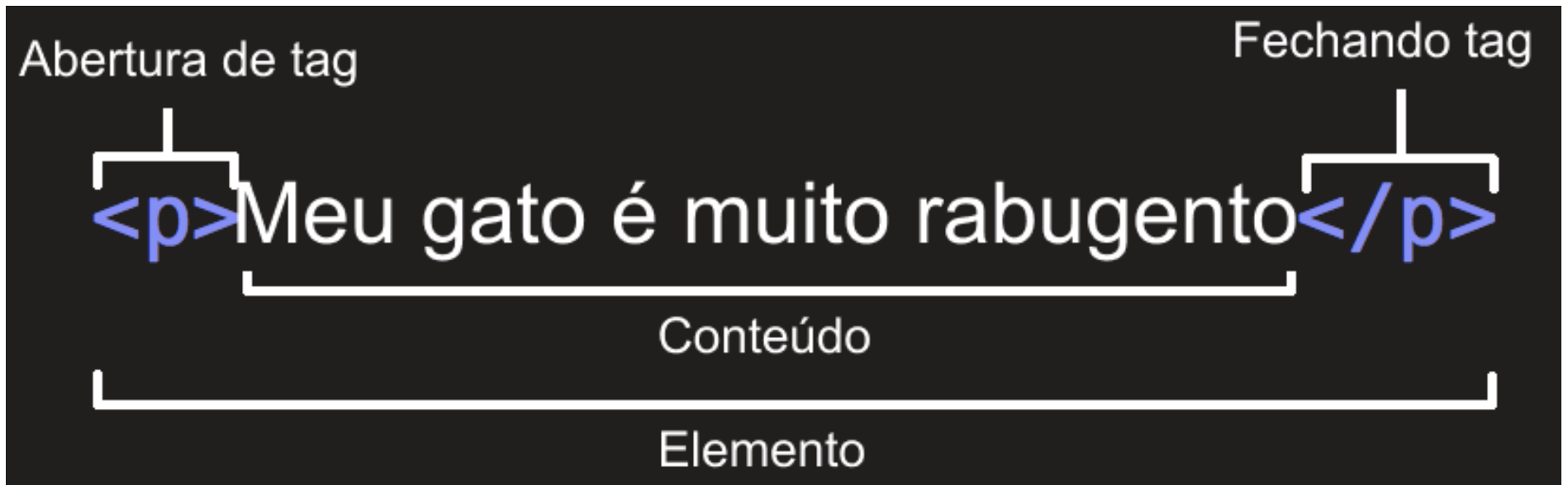
HTML

- HTML (HyperText Markup Language) é a **linguagem de marcação** para a criação e estruturação de páginas web na internet.
- Bloco básico

HTML


- Hypertext/Links
- Elementos/Tags
 - Tabelas
 - Formulários

HTML



HTML - Elemento

Atributo



```
<p class="editor-note">Meu gato é muito rabugento</p>
```

HTML – Comportamento dos elementos

- Aninhamento
- Inline/Bloco
- Elementos vazios

HTML - Aninhamento

- `<p>Meu gato é
muito mal-
humorado.</p>`

HTML - Inline/Bloco

```
<em>primeiro</em>
```

```
<em>segundo</em>
```

```
<em>terceiro</em>
```

primeiro segundo terceiro

```
<p>quarto</p>
```

```
<p>quinto</p>
```

```
<p>sexto</p>
```

quarto

quinto

sexto

HTML – Anatomia de um documento

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title>Minha página de teste</title>
  </head>
  <body>
    
  </body>
</html>
```

HTML - Imagens

- ``

HTML - Marcando o texto - Cabeçalhos

<!-- 4 níveis de título -->

<h1>Meu título principal</h1>

<h2>Meu título de alto nível</h2>

<h3>Meu subtítulo</h3>

<h4>Meu segundo subtítulo</h4>

HTML - Marcando o texto - Parágrafo

```
<p>Este é um parágrafo  
simples</p>
```

HTML - Listas

```
<ul>
```

```
<li>tecnólogos</li>
```

```
<li>pensadores</li>
```

```
<li>construtores</li>
```

```
</ul>
```

HTML - Links

```
<a href="https://www.google.com">Google</a>
```

Prática - HTML

Faça uma pagina

- Cabeçalho
- Imagem
- Paragrafo curto
- Lista
- Paragrao longo
- Paragrao longo + link

CSS

(Aparência/Decoração)

- CSS (Cascading Style Sheets) é uma linguagem usada para **estilizar** a apresentação de documentos HTML, definindo o layout, cores, fontes e outros aspectos visuais de uma página web.

CSS

```
p {  
  color: red;  
}
```

CSS

```
<link href="estilos/estilo.css"  
rel="stylesheet" />
```

```
p {  
    color: red;  
}
```

CSS

Selector

p

{

color: red;

}

Property

Property value

Declaration

CSS

```
p {  
  color: red;  
  width: 500px;  
  border: 1px solid black;  
}
```

CSS

```
p,  
li,  
h1 {  
    color: red;  
}
```

CSS - Tipos de seletores

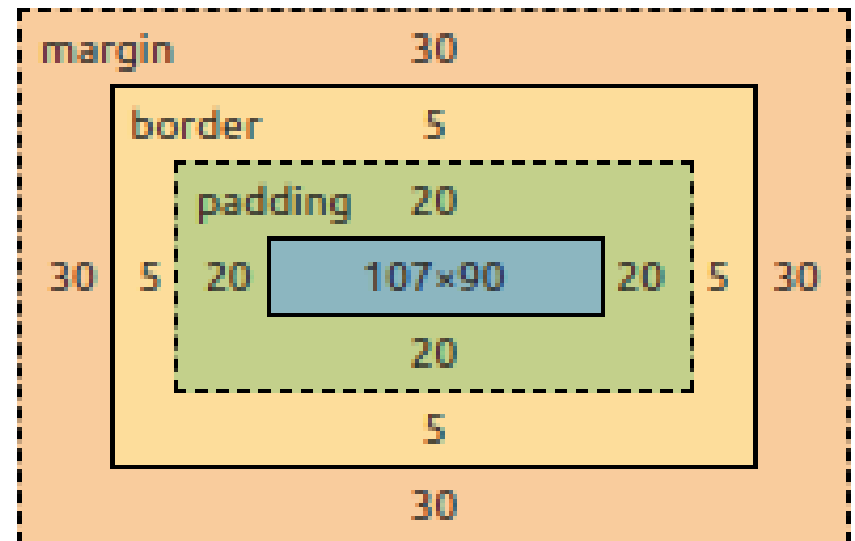
- p
- #my-id
- .my-class
- a: hover
- ...outros

CSS – „Caixas“ dos elementos

- padding, o espaço ao redor do conteúdo (ex.: ao redor do texto de um parágrafo).
- border, a linha sólida do lado de fora do padding.
- margin, o espaço externo a um elemento.

CSS – „Caixas“ dos elementos

Este é um exemplo de um elemento com padding, border e margin.



Prática - CSS

- Aplique o CSS no documento HTML criado anteriormente

Javascript (Comportamento)

- JavaScript é uma **linguagem de programação** utilizada principalmente para criar interatividade em páginas web.

Javascript

```
<script src="scripts/main.js"></script>
```

Javascript

```
<script src="scripts/main.js"></script>
```

```
const myHeading =  
document.querySelector("h1");  
myHeading.textContent = "Olá mundo!";
```

Javascript

- Declare uma variável
 - Regras: nomes, ;
- Tipos: String, Number, Boolean, Array, Object

Javascript

- Operadores:
 - + (numeros e **strings**)
 - -, *, / (puramente contas)
 - atribuição
 - Igualdade e Igualdade estrita (tipo e valor iguais; **obs:memória***) (==; ===)
 - Negação, não igual (!, !=, !==)

Javascript

- Condicionais

```
let iceCream = "chocolate";  
if (iceCream === "chocolate") {  
    alert("Sim, eu amo sorvete de chocolate!");  
} else {  
    alert("Aaaah, mas chocolate é o meu favorito...");  
}
```


Javascript

- Funções:


-

```
function multiply(num1, num2) {  
    let result = num1 * num2;  
    return result;  
}
```

Javascript

- Eventos:

```
document.querySelector("html").addEventListener("click", function () {  
    alert("Ai! Pare de me cutucar!");  
});
```



- Função anônima
 - () => ou function ()

Prática - HTML+Javascript

- Demonstração prática HTML e Javascript

Princípio – Reatividade Framework

- Frameworks reativos facilitam o desenvolvimento de interfaces **dinâmicas** em aplicações web, exemplo: Vue3.

Banco de dados

- Um banco de dados é um sistema organizado para armazenar e gerenciar informações de forma estruturada.

Banco de dados

- As etapas típicas de modelagem de um banco de dados são:

Banco de dados

- Requisitos: Entender as necessidades do usuário e os requisitos funcionais e não funcionais do sistema.

Banco de dados

- Conceitual: Criar um modelo conceitual que represente as entidades principais, seus atributos e os relacionamentos entre elas.

Banco de dados

- Lógica: Transformar o modelo conceitual em um modelo lógico, especificando os detalhes de como as entidades e relacionamentos serão representados no banco de dados.

Banco de dados

- Física: Implementar o modelo lógico em um sistema de gerenciamento de banco de dados específico, considerando aspectos como tipos de dados, índices, chaves, etc.

Juntando conceitos – Exemplo completo

- Um aplicativo com ciclo completo.

Conclusão

- Conceitos
- Arquiteturas
- Protocolo
- HTML/CSS/Javascripts

Perguntas?