SERVLETS E JSP (JEE) - JSP - Ajax

Diogo Cezar Teixeira Batista

diogo@diogocezar.com.br

http://www.diogocezar.com

Universidade Tecnológica Federal do Paraná - UTFPR

Cornélio Procópio - 2012

Ajax I

- AJAX é uma solução lado-cliente baseada em HTML, JavaScript e DOM que permite que a comunicação entre o *browser* e o servidor *web* ocorra de forma assíncrona;
- AJAX não é uma linguagem nova, nem mesmo uma tecnologia nova;
- AJAX não é uma solução lado-servidor;

Ajax II

- Por que usar?
  - a comunicação HTTP é ineficiente:
    - para cada requisição há uma resposta;
    - cada resposta devolve uma página inteira;
    - é preciso esperar toda a página carregar antes de usar uma aplicação *web*;

Ajax III

- AJAX permite comunicação assíncrona:
  - pequenos trechos de dados podem ser transferidos assincronamente;
  - permite que aplicação funcione enquanto dados são transferidos;

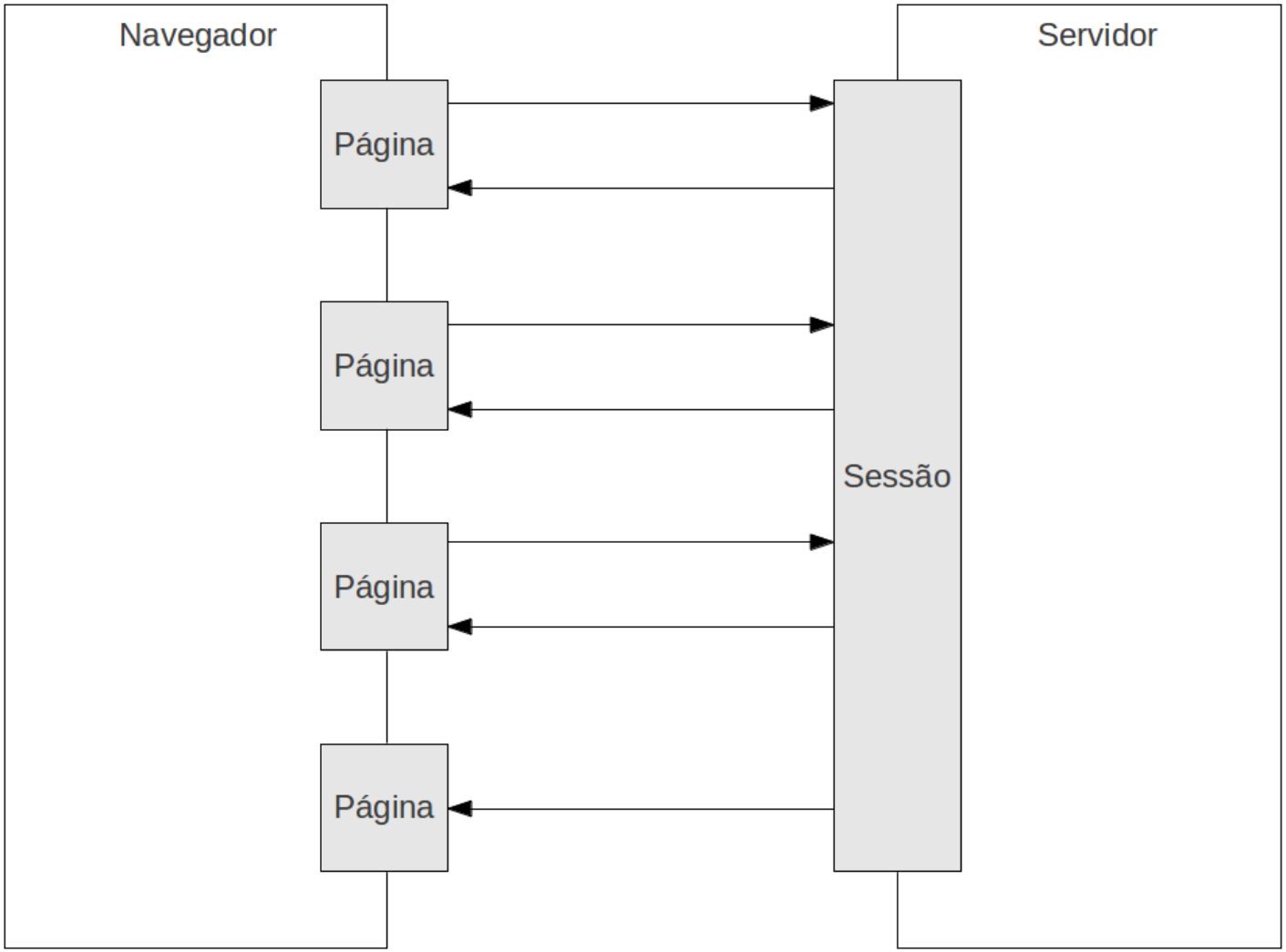
Ajax IV

- Quando usar?
  - use em aplicações *web* interativas que sofrem com o modelo requisição-resposta:
    - aplicações com menus, muitas opções, que requerem interatividade em tempo real;
    - aplicações que modelam sistemas desktop;
  - não use em aplicações que realmente precisam carregar uma página inteira:
    - Ex: tabelas, listas, etc...

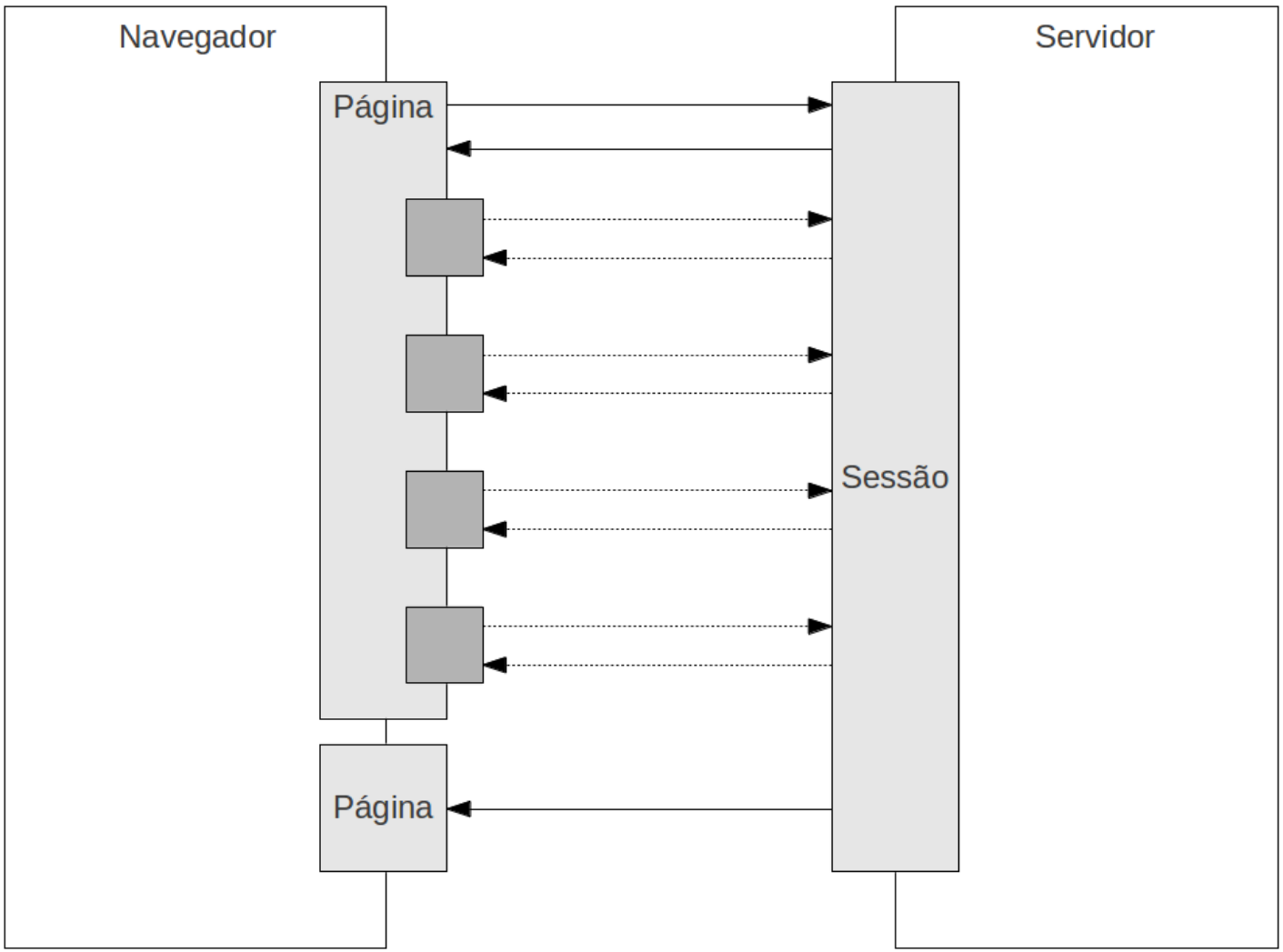
Ajax V

- Quem usa?
  - Google;
  - Yahoo;
  - Microsoft;

Exemplo Comunicação Request-Response



Exemplo Comunicação Ajax



Recursos do JQuery I

- uma das principais funcionalidades do JQuery é o gerenciamento de AJAX;
- funções próprias para controle e gerenciamento da transmissão de dados;
- em geral: consegue-se acessar o conteúdo de uma página jsp dentro do javascript;
- existem diferentes formas para se trabalhar com AJAX em JQUERY, basicamente todas seguem a mesma lógica, mas a sintaxe pode ser diferenciada;

Recursos do JQuery II

- Sintaxe: `load(url,parametros,callback)`
  - *url*: A URL que é solicitada a requisição.
  - *parametros*: Um objeto cuja propriedades são serializadas em uma série de parâmetros codificados corretamente e que se passam a requisição. Se utilizado, deve-se especificar se a requisição utiliza o método POST, caso contrário, se omitido, a requisição utiliza o método GET.
  - *callback*: Uma função chamada após a solicitação já ter sido processada.
- Esse comando serve apenas para ler HTML's;

```
$("#conteudo").load("arquivo.html");
```

Recursos do JQuery III

```
<script>
  $(document).ready(function() {
    $("input[type=button]").click(function(event) {
      $("#conteudo").load("arquivo.html",aviso());
    });
  });
  function aviso(){
    alert("O conteudo sera carregado agora!");
  }
</script>
```

- carrega o conteúdo de arquivo.html na div com id conteúdo e dispara a função aviso() quando o conteúdo for lido;

Recursos do JQuery IV

- pode-se filtrar o resultado obtido, com os seletores disponíveis pelo próprio JQuery;

```
$("#box").load("meu_arquivo.html #conteudo")
```

- o exemplo acima só irá retornar o div com id conteudo do arquivo `meu_arquivo.html`
- esse recurso é exclusivo do método load();

Recursos do JQuery V

- Sintaxe: `$.post(url, parametros, funcao_retorno(data));`
  - *url*: A URL que é solicitada a requisição.
  - *parametros*: Um objeto cuja propriedades são serializadas em uma série de parâmetros codificados corretamente e que se passam a requisição.
  - *funcao_retorno(data)* : Uma função chamada após a solicitação já ter sido processada.
  - *data* é o resultado retornado
- utilizado para resgatar recursos das linguagens de programação, em nosso caso, páginas jsp;

Recursos do JQuery VI

- ao utilizar $.post os atributos serão enviados por post;
- ao utilizar $.get os atributos serão enviados por get;
- para passar parâmetros deve-se utilizar a seguinte estrutura:

```
{variavel1: valor1,
variaval2: valor2,
...
variaveln: valorn}
```

Recursos do JQuery VII

```
<script type="text/javascript">
  $(document).ready(function() {
    $("input[type=button]").click(function(event) {
      var acao = $(this).attr("value");
      $.post("pagina.jsp",{acao:acao},
function(data){
  $("#resultado").empty().html(data);
}
      );
    });
  });
</script>
```

Recursos do JQuery VIII

```
<%
   String acao = request.getParameter("acao");
   if(acao.isEmpty){
%>
   <p>Ação está vazia!</p>
<%
   }
   else{
<%
   <p>Conteúdo a ser lido</p>
%>
   }
%>
```

Recursos do JQuery IX

- é possível estabelecer ações para outras divs enquanto o AJAX é executado, ideal para criar uma div “Loading”;
- para isso, utiliza-se os comandos:

```
$("#loading").ajaxStart("faça alguma coisa");
$("#loading").ajaxStop("faça alguma coisa");
```

- existem ainda outros comandos de controle para o Ajax:
  - `ajaxComplete(callback)`
  - `ajaxError(callback)`
  - `ajaxSend(callback)`
- documentação completa: `http://docs.jquery.com/Ajax`