



Prof. Taciano Balardin
tacianobalardin@gmail.com



E-MAIL:

taciano@ulbra.edu.br



SITE DA DISCIPLINA:

<http://www.taciano.pro.br/>

SENHA:

[@quarta](#)

JavaScript

FUNDAMENTOS PROFISSIONAIS

AULA 14

Trabalho de G2:

- Elaborar uma página pessoal que contenha:
 - Página Inicial: apresentação inicial e outras informações como foto pessoal, histórico escolar, informações profissionais, cursos realizados, etc (mini-currículo);
 - Interesse: página com textos e imagens sobre pelo menos 2 temas de interesse vistos até agora no curso;
 - Hobby / Passatempo: página com texto e imagens sobre coisas que você gosta de fazer nos horários livres;
 - Contato: um formulário de contato com pelo menos 5 campos (input's e pelo menos um select).

Trabalho de G2:

- Estrutura da página:
 - Topo;
 - Menu;
 - Conteúdo;
 - Rodapé.
- Estas páginas devem ser formatadas com arquivo CSS externo e único;
- Deve conter pelo menos dois códigos JavaScript que realizarão algum tipo de modificação na página (veremos na aula de hoje);
- A página deverá ser apresentada para a turma no dia 04/06;
- Tem peso 3 no G2:
 - 2 pontos para as páginas + 1 ponto de apresentação.

Nota de G2

- ✓ 10% - Correção da Prova de G1;
- ✓ 20% - Exercícios de hoje entregar dia 14/05 impresso ou a mão;
- 🕒 30% - Página pessoal;
- 🕒 40% - Prova.

Java **não é**

JAVASCRIPT

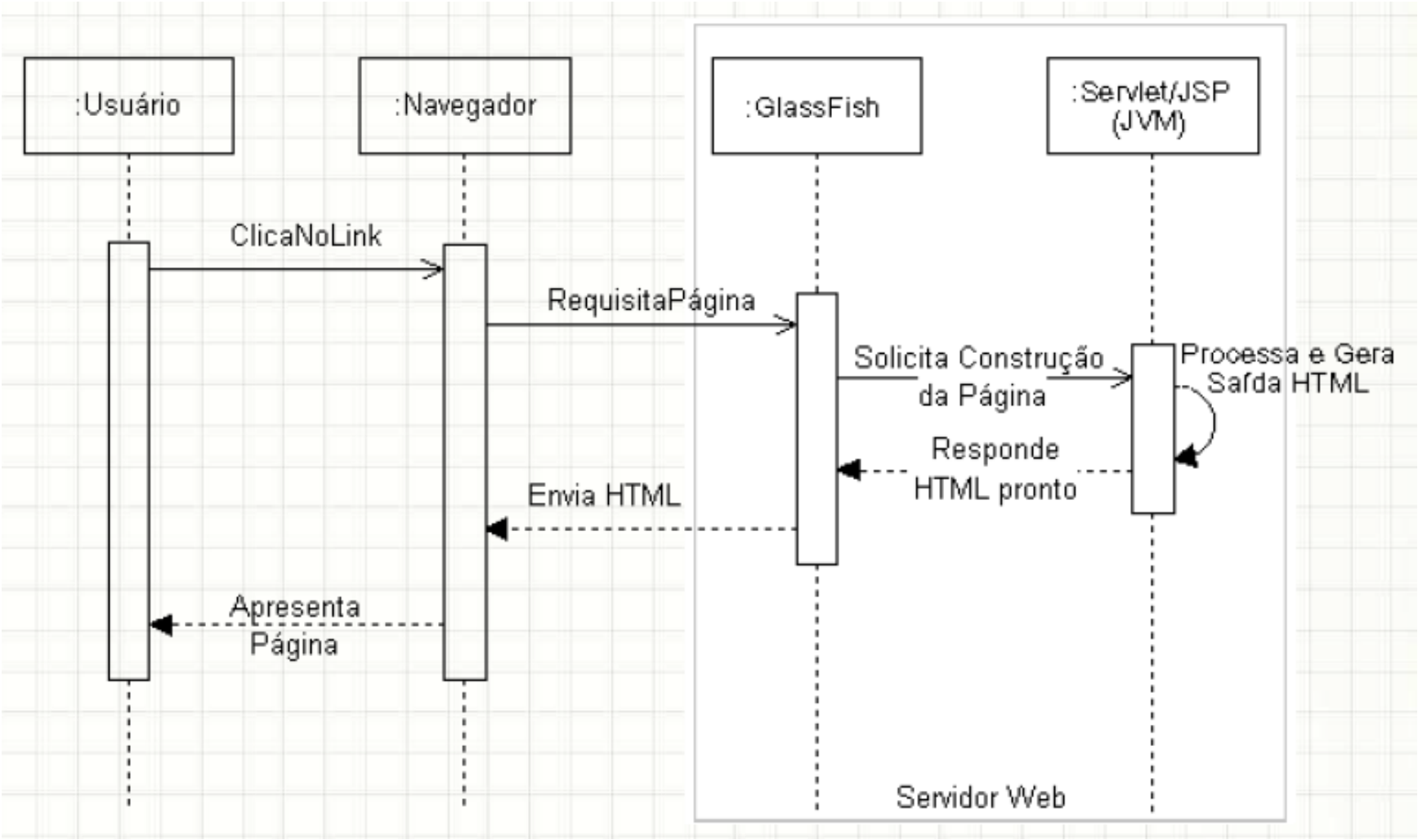
Introdução

- Páginas Web Dinâmicas?
 - Capazes de Mudar!
- Como implementar?
 - Através de programação:
 - Associada ao código da página;
 - Associada à ações do usuário.
- Quais linguagens permitem fazer isso?

Linguagens mais usadas

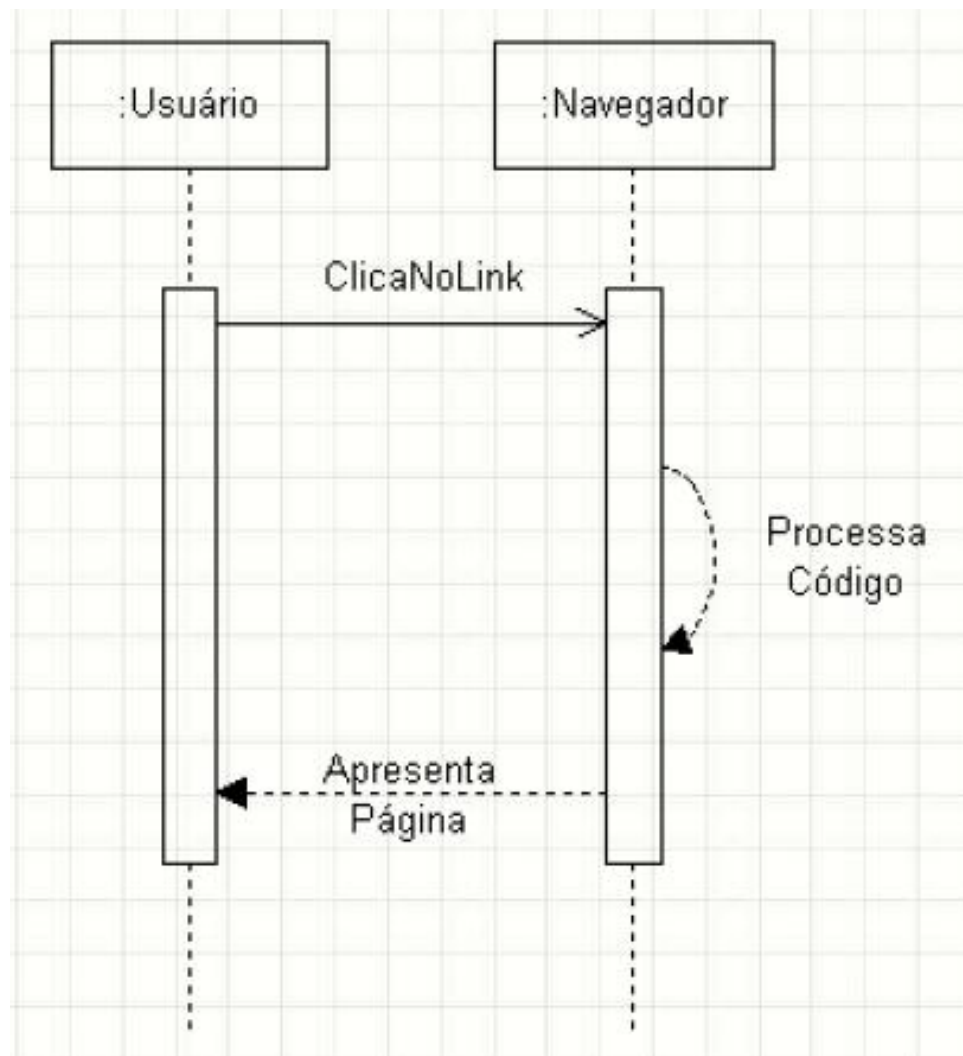
Nome	Responsável	Tipo	Execução
JavaScript	Mozilla	Interpretada	Cliente
PHP	PHP	Interpretada	Servidor
ASP	Microsoft	Interpretada	Servidor
ASP.NET	Microsoft	Compilada	Servidor
JSP	Oracle	Compilada	Servidor

Server-side



Client-side

**MUITO MAIS
RÁPIDO!**



Server-side x Client-side

- E a segurança?
 - Client-side é mais fraco.
 - Usuário tem acesso ao código;
 - Usuário pode modificar o código.
 - Transferir para o lado cliente é bom;
 - Mas requer cuidado!
 - Apenas se não comprometer a segurança!



JavaScript

- Usaremos para:
 - Mudar cores;
 - Mudar textos;
 - Manipular formulário.
- Além disso, é possível:
 - Implementar menus;
 - Criar pop-up, alertas;
 - Validar formulários;
 - Gerar solicitações AJAX, etc.



Como integrar JS e HTML?

- Similar ao CSS!
 - Não vamos inserir no HTML;
 - Entretanto, assim como no CSS, é possível.
- Processo de Criação:
 1. Criar uma página com alguns elementos;
 2. Criar a função a associar a algum evento;
 3. Associar a função ao evento de um elemento.

Exemplo 1

1. Crie uma página **teste.html** com o corpo abaixo:

```
<body>
```

```
  <p>Teste </p>
```

```
  <p>
```

```
    <input type="button" value="Cor" />
```

```
  </p>
```

```
</body>
```

Exemplo 1

2. Crie um arquivo **efeitos.js**:

2.1 Neste arquivo, vamos criar uma **função** chamada **mudaCorDeFundo**.

```
function mudaCorDeFundo() {  
  
}
```

- Para modificar o fundo, precisamos acessar o elemento **document.body**

Exemplo 1

3. Vamos mudar a propriedade
style.backgroundColor do corpo (body)

```
function mudaCorDeFundo() {  
    document.body.style.backgroundColor = "blue";  
}
```

- Observe que o que é **background-color** no CSS, no JavaScript virou **backgroundColor**;
- Agora precisamos associar o **JS ao HTML**.

Exemplo 1

4. Voltamos ao arquivo **teste.html**

- No cabeçalho, dentro da seção <head>, inclua a seguinte linha:

```
<script type="text/javascript" src="efeitos.js"> </script>
```

- Agora carregue a página e veja o que acontece quando clicar no botão :-)

Exemplo 1

5. Precisamos dizer **QUANDO** aquela função será executada!

- Isso é feito associando-a ao evento **onclick** do botão que criamos no HTML!
- O jeito mais “simples” de fazer isso é:

```
<input type="button" value="Cor"  
      onclick="javascript:mudaCorDeFundo()" />
```

- Agora carregue a página e veja o que acontece quando clicar no botão :-)

Exemplo 1

- Mas isso mistura **código** com **HTML**!
- Tem como fazer diferente?
 - Claro!
 - Mas, para isso, precisamos dar um ID para o botão:

```
<input type="button" value="Cor" id="mudacor" />
```

Exemplo 1

- Voltamos agora ao **efeitos.js**;
- Precisamos dizer ao navegador que a função **mudaCorDeFundo** está associada ao evento **onclick** do elemento de ID **mudacor**.

Exemplo 1

- No **efeitos.js**:

```
window.onload = iniciar;
```

Evento acionado quando o navegador termina de processar **todo o HTML**.

```
function iniciar() {  
    var botao = document.getElementById("mudacor");  
    botao.onclick = mudaCorDeFundo;  
}
```

```
function mudaCorDeFundo() {  
    document.body.style.backgroundColor = "blue";  
}
```

Exemplo 2

- Mudando um texto: editar arquivo **teste.html**

```
<body>
```

```
<p id="ajuda"> <br/> </p>
```

```
<p>
```

```
<input type="button" value="Cor" id="mudacor" />
```

```
</p>
```

```
</body>
```

Exemplo 2

- Para mudar o texto, alteramos a propriedade **innerHTML** do elemento:

```
var texto = document.getElementById("ajuda");
```

```
texto.innerHTML = "Clique para trocar de cor!!";
```

- Usualmente isso é associado ao evento **onmouseover**, que é quando o mouse está sobre o elemento;
- **onmouseout** indica quando o mouse saiu do elemento

Eventos Comuns

- Document, Window e Body

Elementos	Evento	Quando Executa
Document e Body	onload	Quando o documento inicia o carregamento
Document e Body	onunload	Quando é iniciada a finalização do documento (o usuário foi para outra página)
Window	onload	Quando o documento foi totalmente carregado e processado

Eventos Comuns

■ Forms

Evento	Quando Executa
onchange	Quando o conteúdo do elemento foi alterado
onfocus	Quando o elemento ganhou foco
onselect	Quando um item foi selecionado (combobox)
onsubmit	Quando o formulário for enviado

Eventos Comuns

- Teclado (quase todos elementos)

Evento	Quando Executa
onkeydown	Quando uma tecla é apertada
onkeyup	Quando uma tecla é solta
onkeypress	Quando uma tecla for apertada e solta

Eventos Comuns

- Mouse (quase todos elementos)

Evento	Quando Executa
onclick	Quando o elemento for clicado
ondblclick	Quando o elemento for duplamente clicado
onmousemove	Quando o mouse se mover sobre o elemento
onmouseout	Quando o mouse sair de cima do elemento
onmouseover	Quando o mouse passar sobre o elemento
onmouseup	Quando o botão do mouse for solto sobre o elemento

Propriedades Comuns

- Plano de fundo

Propriedade	O que altera
backgroundColor	Cor de fundo do elemento
backgroundImage	Imagem de fundo do elemento

- Textos

Propriedade	O que altera
color	Muda cor do texto
fontSize	Muda tamanho da fonte
textAlign	Muda alinhamento do texto

Propriedades Comuns

■ Bordas e Margens

Propriedade	O que altera
borderColor	Muda cor da borda
borderStyle	Muda estilo da borda
borderWidth	Muda tamanho da borda
margin	Muda distância da margem
padding	Muda espaçamento interno do elemento

Propriedades Comuns

- Layout

Propriedade	O que altera
cursor	Muda o cursor apresentado
display	Muda a maneira de apresentação do elemento
visibility	Muda a visibilidade do elemento
width	Muda a largura do elemento