

---

# RELATÓRIO DA APRESENTAÇÃO JQUERY

---

SISTEMAS DISTRIBUÍDOS E TOLERÂNCIA A FALHAS

**Ângelo Santos M 4189**

**André Elias M4272**

## ÍNDICE

---

---

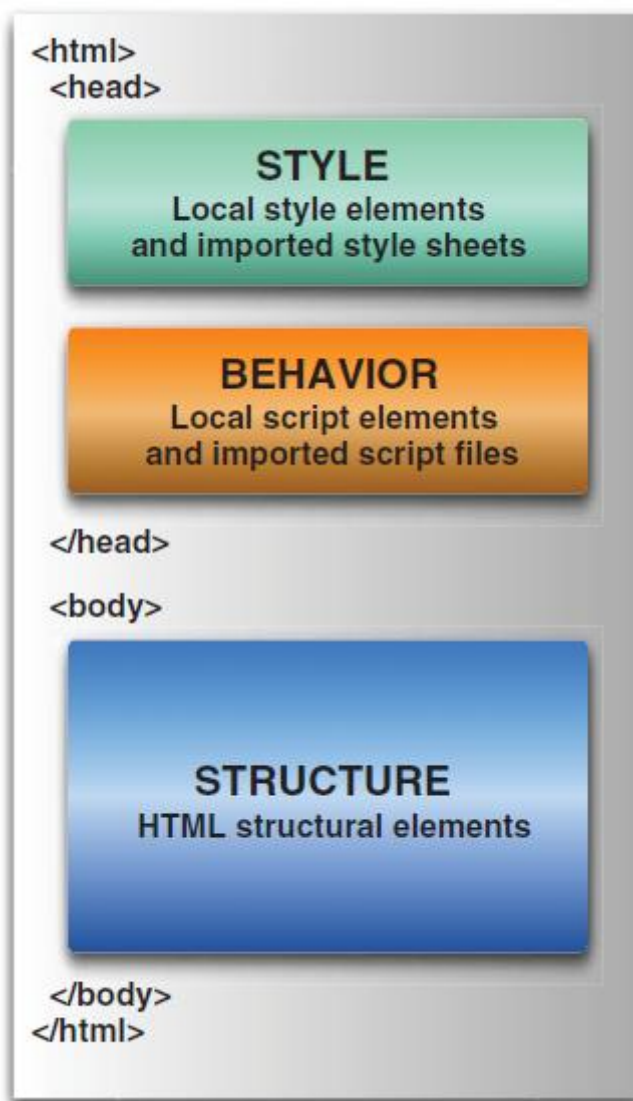
Arquitectura e funcionamento .....	3
Ferramentas.....	5
Exemplo 1 de JQUERY (eventos e EFEITOS).....	6
Exemplo 2 de JQUERY (accordion widget).....	11

---

# ARQUITECTURA E FUNCIONAMENTO

---

Aqui podemos ver a estrutura base de qualquer página HTML:



Todo o código *jQuery* deve ser inserido na secção aqui assinalada como “*BEHAVIOR*”, que se refere a toda a parte responsável pelo comportamento, interacção e animação da página.

Isto é feito através do uso das etiquetas `<script>` da seguinte forma:

```
<script type="text/javascript">
```

```
</script>
```

Entre as chavetas é feita a inclusão das bibliotecas bem Como introduzido qualquer código *jQuery*.

Sendo esta uma biblioteca baseada numa linguagem de *scripting* (*JavaScript*), o código é executado directamente pelos navegadores não sendo necessária a compilação ou qualquer tratamento prévio, apenas é necessário que a linguagem seja suportada pelo navegador.

**jQuery API**

**Core**

- `$(...)` The jQuery Function:
- `jQuery(expression, [context])`
- `jQuery(html)`
- `jQuery(elements)`
- `jQuery(callback)`
- jQuery Object Accessors:**
- `each(callback)`
- `size()`
- `length`
- `eq(position)`
- `get()`
- `index(subject)`
- Plugins:**
- `jQuery.fn.extend(object)`
- `jQuery.extend(object)`
- Interoperability:**
- `jQuery.noConflict()`
- `jQuery.noConflict(extreme)`

**Attributes**

- Attr:**
- `attr(name) / attr(properties)`
- `attr(key, value) / attr(key, fn)`
- `removeAttr(name)`
- Class:**
- `addClass(class)`
- `removeClass(class)`
- `toggleClass(class)`
- HTML:**
- `html() / html(val)`
- Text:**
- `text() / text(val)`
- Value:**
- `val() / val(val)`

**Events**

- Page Load:**
- `ready(fn)`
- Event Handling:**
- `bind(type, [data], fn)`
- `one(type, [data], fn)`
- `trigger(type, [data])`
- `triggerHandler(type, [data])`
- `unbind([type], [data])`
- Event Helper:**
- `blur() / blur(fn)`
- `change() / change(fn)`
- `click() / click(fn)`
- `dblclick() / dblclick(fn)`
- `error() / error(fn)`
- `focus() / focus(fn)`
- `keydown() / keydown(fn)`
- `keypress() / keypress(fn)`
- `keyup() / keyup(fn)`
- `load(fn)`
- `mousedown(fn)`
- `mousemove(fn)`
- `mouseout(fn)`
- `mouseover(fn)`
- `mouseup(fn)`
- `resize(fn)`
- `scroll(fn)`
- Interaction Helpers:**
- `select() / select(fn)`
- `submit() / submit(fn)`
- `unload(fn)`
- `hover(over, out)`
- `toggle(fn, fn)`

**Selectors**

- Basics:**
- `#id`
- `element`
- `.class`
- `*`
- `selector1.selector2.selectorN`
- Hierarchy:**
- `ancestor descendant`
- `parent > child`
- `prev + next`
- `prev ~ siblings`
- Attribute Filters:**
- `[attribute]`
- `[attribute=value]`
- `[attribute!=value]`
- `[attribute$=value]`
- `[attribute*=value]`
- `[selector1][selector2][selectorN]`
- Basic Filters:**
- `:first`
- `:last`
- `:not(selector)`
- `:even`
- `:odd`
- `:eq(index)`
- `:gt(index)`
- `:lt(index)`
- `:header`
- `:animated`
- Content Filters:**
- `:contains(text)`
- `:empty`
- `:has(selector)`
- `:parent`
- Visibility Filters:**
- `:hidden`
- `:visible`
- Child Filters:**
- `:nth-child(index/even/odd/equation)`
- `:first-child`
- `:last-child`
- `:only-child`
- Forms:**
- `:input`
- `:text`
- `:password`
- `:radio`
- `:checkbox`
- `:submit`
- `:image`
- `:reset`
- `:button`
- `:file`
- `:hidden`
- Form Filters:**
- `:enabled`
- `:disabled`
- `:checked`
- `:selected`

**Traversing**

- Filtering:**
- `eq(index)`
- `hasClass(class)`
- `filter(expr) / filter(fn)`
- `is(expr)`
- `map(callback)`
- `not(expr)`
- `slice(start, [end])`
- Finding:**
- `add(expr)`
- `children([expr])`
- `contents()`
- `find(expr)`
- `next([expr])`
- `nextAll([expr])`
- `parent([expr])`
- `parents([expr])`
- `prev([expr])`
- `prevAll([expr])`
- `siblings([expr])`
- Chaining:**
- `andSelf()`
- `end()`

**Manipulation**

- Changing Contents:**
- `html() / html(val)`
- `text() / text(val)`
- Inserting Inside:**
- `append(content)`
- `appendTo(content)`
- `prepend(content)`
- `prependTo(content)`
- Inserting Outside:**
- `after(content)`
- `before(content)`
- `insertAfter(content)`
- `insertBefore(content)`
- Inserting Around:**
- `wrap(html) / wrap(elem)`
- `wrapAll(html) / wrap(elem)`
- `wrapInner(html)`
- `wrapInner(elem)`
- Replacing:**
- `replaceWith(content)`
- `replaceAll(selector)`
- Removing:**
- `empty()`
- `remove([expr])`
- Copying:**
- `clone() / clone(true)`

**Utilities**

- User Agent:**
- `jQuery.browser`
- `jQuery.browser.version`
- `jQuery.boxModel`
- Array and Object operations:**
- `jQuery.each(object, callback)`
- `jQuery.extend(target, object1, [objectN])`
- `jQuery.grep(array, callback, [invert])`
- `jQuery.makeArray(obj)`
- `jQuery.map(array, callback)`
- `jQuery.inArray(value, array)`
- `jQuery.unique(array)`
- Test operations:**
- `jQuery.isFunction(obj)`
- String operations:**
- `jQuery.trim(str)`

**Ajax**

- Ajax Request:**
- `jQuery.ajax(options)`
- `load(url, [data], [callback])`
- `jQuery.get(url, [data], [callback])`
- `jQuerygetJSON(url, [data], [callback])`
- `jQuery.post(url, [data], [callback], [type])`
- Ajax Event:**
- `ajaxComplete(callback)`
- `ajaxError(callback)`
- `ajaxSend(callback)`
- `ajaxStart(callback)`
- `ajaxStop(callback)`
- `ajaxSuccess(callback)`
- Misc:**
- `jQuery.ajaxSetup(options)`
- `serialize()`
- `serializeArray()`

**Effects**

- Basics:**
- `hide() / hide(speed, [callback])`
- `show() / show(speed, [callback])`
- `toggle()`
- Sliding:**
- `slideDown(speed, [callback])`
- `slideUp(speed, [callback])`
- `slideToggle(speed, [callback])`
- Fading:**
- `fadeIn(speed, [callback])`
- `fadeOut(speed, [callback])`
- `fadeTo(speed, opacity, [callback])`
- Custom:**
- `animate(params, [duration], [easing], [callback])`
- `animate(params, options)`
- `stop()`
- `queue(callback) / queue(queue)`
- `dequeue()`

**CSS**

- CSS:**
- `css(name) / css(properties)`
- `css(name, value)`
- Positioning:**
- `offset()`
- Height and Width:**
- `height() / height(val)`
- `width() / width(val)`

Yoshiomi, KURISU - <http://dev.chrisryu.com>

Na imagem acima podemos ver a API do *jQuery*, onde é possível distinguir os tipos da biblioteca, bem como os métodos separados por finalidade.

O código *jQuery* apenas é executado depois da estrutura DOM estar pronta e totalmente carregada.

Todas as ações e efeitos são executados sequencialmente e até terminarem, podendo ser encadeados.

---

# FERRAMENTAS

---

A edição de código *jQuery* pode ser feita em qualquer simples editor de texto sendo que para que funcione apenas é necessário a inclusão da biblioteca, local ou remotamente.

No entanto para a construção dos exemplos usados na apresentação e do projecto escolhemos usar o IDE *NetBeans* que proporciona verificação ortográfica para *JavaScript* bem como reconhecimento e sugestão de elementos para a biblioteca *jQuery* e para o desenvolvimento HTML.

---

## EXEMPLO 1 DE JQUERY (EVENTOS E EFEITOS)

---

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Sistemas Distribuídos E Tolerância A Falhas - Exemplo 1</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    <!-- carrega a biblioteca jQuery -->
    <script type="text/javascript" src="js/jquery-1.5.1.js"></script>

    <script type="text/javascript">
      $(document).ready(function(){
        /* fade in */
        $("#bt1").click(function(){
          $("#ubi").fadeIn("slow");
        });
        /* fade out */
        $("#bt2").click(function(){
          $("#ubi").fadeOut("slow");
        });
        /* mover direita */
        $("#bt3").click(function(){
          $("#ubi").animate({"margin-left": "+=50px"}, "slow");
        });
        /* mover esquerda */
        $("#bt4").click(function(){
          $("#ubi").animate({"margin-left": "-=50px"}, "slow");
        });
      });
    </script>
  </head>
  <body>
    <div style="text-align: center;">
      <div style="display: inline-block; margin-right: 20px;">
        <input type="button" value="Fazer Fade In" />
      </div>
      <div style="display: inline-block; margin-right: 20px;">
        <input type="button" value="Fazer Fade Out" />
      </div>
      <div style="display: inline-block; margin-right: 20px;">
        <input type="button" value="Mover Direita" />
      </div>
      <div style="display: inline-block;">
        <input type="button" value="Mover Esquerda" />
      </div>
    </div>
    <div style="text-align: center; margin-top: 20px;">
      <div style="display: inline-block; margin-right: 20px;">
        <input type="button" value="Mostrar" />
      </div>
      <div style="display: inline-block; margin-right: 20px;">
        <input type="button" value="Esconder" />
      </div>
      <div style="display: inline-block; margin-right: 20px;">
        <input type="button" value="Mover Direita" />
      </div>
      <div style="display: inline-block;">
        <input type="button" value="Mover Esquerda" />
      </div>
    </div>
  </body>
</html>
```

```
/*mudar de cor*/
$("#vermelho").click(function(){
    $("h1").css('color', 'red');
    $(this).css('color', 'black');
    $("#azul").css('color', 'white');
    $("#verde").css('color', 'white');

});

$("#azul").click(function(){
    $("h1").css('color', 'blue');
    $(this).css('color', 'black');
    $("#vermelho").css('color', 'white');
    $("#verde").css('color', 'white');
});

$("#verde").click(function(){
    $("h1").css('color', 'green');
    $(this).css('color', 'black');
    $("#azul").css('color', 'white');
    $("#vermelho").css('color', 'white');
});
});
</script>

<style type="text/css">
    /*.blue { color: blue; } /*faz um elemento azul quando aplicada*/
    /*.green { color: green; } /*faz um elemento verde quando aplicada*/
    /*.red { color: red; } /*faz um elemento vermelho quando aplicada*/
```

```
img {padding-right: 20px; float:left}
```

```
</style>
```

```
<style type="text/css">
```

```
.divAbsolute {  
    width:500px;  
    height:150px;  
    position:absolute;  
    top:420px;  
}
```

```
.divInline {  
    width:200px;  
    height:150px;  
    position:absolute;  
    top:20px;  
    left: 410px;  
    display: inline;  
}
```

```
#vermelho {  
    width: 100px;  
    height: 100px;  
    background-color: red;  
    color: white;  
}
```



```
#azul {  
    width: 100px;  
    height: 100px;  
    background-color: blue;  
    color: white;  
}
```

```
#verde {  
    width: 100px;  
    height: 100px;  
    background-color: green;  
    color: white;  
}
```

```
</style>  
</head>  
<body>  
    <div>  
        <div id="imagem">  
            <h1>Exemplo Efeitos jQuery</h1>  
              
        </div>  
        <div class="divAbsolute" id="botoes">  
            <h3>Botões</h3>  
            <input id="bt1" type="button" value="Fade In" name="resposta_1" />  
            <input id="bt2" type="button" value="Fade Out" name="resposta_2" />  
            <input id="bt3" type="button" value="Direita" name="resposta_3" />  
            <input id="bt4" type="button" value="Esquerda" name="resposta_4" />  
        </div>  
        <div class="divInline" id="cores" >
```

```
<h3>Cores</h3>
<div id="vermelho" align="center">
  <p>Vermelho</p>
</div>
<div id="azul" align="center">
  <p>Azul</p>
</div>
<div id="verde" align="center">
  <p>Verde</p>
</div>
</div>
</div>
</body>

</html>
```

---

## EXEMPLO 2 DE JQUERY (ACCORDION WIDGET)

---

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>jQuery Test Project</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    <!-- carrega as bibliotecas jQuery e jQueryUI-->
    <script type="text/javascript" src="js/jquery-1.5.1.js"></script>
    <script type="text/javascript" src="js/jquery.ui.core.js"></script>
    <script type="text/javascript" src="js/jquery.ui.widget.js"></script>
    <script type="text/javascript" src="js/jquery.ui.accordion.js"></script>

    <script type="text/javascript">
      $(document).ready(function(){
        /* faz com que h1 fique azul quando se clica nele */
        /*$("h1").click(function(){
          $(this).addClass("blue").fadeOut("slow");
        })*
        $("#infolist").accordion({
          autoHeight: false
        });
      });
    </script>

    <!-- classe CSS, aplica formatacao à lista e imagens-->
    <style type="text/css">
      /*.blue { color: blue; } */ /*faz um elemento azul quando aplicada*/
```

```
ul {list-style-type: none}
img {padding-right: 20px; float:left}
```

```
#infolist {width:500px}
```

```
</style>
```

```
<!-- folhas de estilos com o tema predefinido do jQueryUI -->
```

```
<link rel="stylesheet" href="css/jquery-ui-1.8.10.custom.css" type="text/css">
```

```
</head>
```

```
<body>
```

```
<!-- <h1>Test1</h1> -->
```

```
<div id="infolist">
```

```
<h3><a href="#">Mary Adams</a></h3>
```

```
<div>
```

```

```

```
<ul>
```

```
<li><h4>Vice President</h4></li>
```

```
<li><b>phone:</b> x8234</li>
```

```
<li><b>office:</b> 102 Bldg 1</li>
```

```
<li><b>email:</b> m.adams@company.com</li>
```

```
</ul>
```

```
<br clear="all">
```

```
</div>
```

```
<h3><a href="#">John Matthews</a></h3>
```

```
<div>
```

```

```

```
<ul>
  <li><h4>Middle Manager</h4></li>
  <li><b>phone:</b> x3082</li>
  <li><b>office:</b> 307 Bldg 1</li>
  <li><b>email:</b> j.matthews@company.com</li>
</ul>
<br clear="all">
</div>
```

```
<h3><a href="#">Sam Jackson</a></h3>
```

```
<div>
  
  <ul>
    <li><h4>Deputy Assistant</h4></li>
    <li><b>phone:</b> x3494</li>
    <li><b>office:</b> 457 Bldg 2</li>
    <li><b>email:</b> s.jackson@company.com</li>
  </ul>
  <br clear="all">
</div>
```

```
<h3><a href="#">Jennifer Brooks</a></h3>
```

```
<div>
  
  <ul>
    <li><h4>Senior Technician</h4></li>
    <li><b>phone:</b> x9430</li>
    <li><b>office:</b> 327 Bldg 2</li>
    <li><b>email:</b> j.brooks@company.com</li>
  </ul>
```

```
<br clear="all">
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```