


|   |                                       |  |
|---|---------------------------------------|--|
|  | <b>INSTITUTO DE EDUCAÇÃO SUPERIOR</b> |  |
|   | <b>CURSO</b>                          | CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA A INTERNET |
|   | <b>DISCIPLINA</b>                     | TÓPICOS AVANÇADOS EM TI (PHP)                            |
|   | <b>PERÍODO: P7</b>                    | <b>Semestre: 2009.1</b>                                  |
|   | <b>PROFESSOR</b>                      | <b>Geraldo Rodrigues de Melo Neto</b>                    |

## AJAX:

---

### Criando a tabela

```
CREATE TABLE `pessoa` (
  `id_pessoa` int(10) unsigned NOT NULL auto_increment,
  `nome` varchar(45) default NULL,
  PRIMARY KEY (`id_pessoa`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

### Inserindo dados no banco

```
INSERT INTO `pessoa` (`id_pessoa`,`nome`) VALUES (1,'ALEXANDRE VIEIRA DE OLIVEIRA'),
INSERT INTO `pessoa` (`id_pessoa`,`nome`) VALUES (2,'ALEXANDRE TESTANDO'),
INSERT INTO `pessoa` (`id_pessoa`,`nome`) VALUES (3,'WAGNER DOS SANTOS'),
INSERT INTO `pessoa` (`id_pessoa`,`nome`) VALUES (4,'VIRGINIA'),
INSERT INTO `pessoa` (`id_pessoa`,`nome`) VALUES (5,'PEDALA ROBINHO'),
INSERT INTO `pessoa` (`id_pessoa`,`nome`) VALUES (6,'MASSA DEMAIS');
```

### Página index.php

```
<html>
<head>
<title>BUSCA DINÂMICA</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script src="script.js"></script>
<script>
function pesquisa(valor)
{
//FUNÇÃO QUE MONTA A URL E CHAMA A FUNÇÃO AJAX
url="busca_nome.php?valor="+valor;
ajax(url);
}
```

```

}
</script>
</head>

<body>
<table width="657" border="0" bgcolor="#f7f7f7">
<tr>
<td align="center" bgcolor="#CCCCCC"><strong><font color="#FF0000"
size="1" face="Verdana, Arial, Helvetica, sans-serif">&lt;-
BUSCA DIN&Acirc;MICA -&gt;</font></strong></td>
</tr>
<tr>
<td><font size="1" face="Verdana, Arial, Helvetica, sans-
serif"><strong>DIGITE
AQUI O NOME DA PESSOA</strong></font>:
<!-- NOTE Q A CADA TECLA PRESSIONADA É CHAMADA A FUNÇÃO
PESQUISA PASSANDO O QUE O USUARIO TEM DIGITADO ATÉ O
MOMENTO -->
<input type="text" name="nome" onKeyPress="pesquisa(this.value)"></td>
</tr>
<tr>
<td>
<!-- AQUI SERÁ APRESENTADO O RESULTADO DA BUSCA DINÂMICA..
OU SEJA OS NOMES -->
<div id="pagina"></div></td>
</tr>
</table>

</body>
</html>

```

Arquivo de javascript usado: script.js

```

// JavaScript Document
// FUNÇÃO RESPONSÁVEL DE CONECTAR A UMA PAGINA EXTERNA NO
NOSSO CASO A BUSCA_NOME.PHP
// E RETORNAR OS RESULTADOS

```

```

function ajax(url)
{

//alert(nick);
//alert(dest);
//alert(msg);

req = null;
// Procura por um objeto nativo (Mozilla/Safari)
if (window.XMLHttpRequest) {
req = new XMLHttpRequest();

```

```

req.onreadystatechange = processReqChange;
req.open("GET",url,true);
req.send(null);
// Procura por uma versão ActiveX (IE)
} else if (window.ActiveXObject) {
req = new ActiveXObject("Microsoft.XMLHTTP");
if (req) {

req.onreadystatechange = processReqChange;
req.open("GET",url,true);

req.send();
}
}
}

function processReqChange()
{

// apenas quando o estado for "completado"
if (req.readyState == 4) {

// apenas se o servidor retornar "OK"

if (req.status ==200) {

// procura pela div id="pagina" e insere o conteudo
// retornado nela, como texto HTML

document.getElementById('pagina').innerHTML = req.responseText;

} else {
alert("Houve um problema ao obter os dados:n" + req.statusText);
}
}
}
}

```

E a página que faz a busca em si: busca\_nome.php

```

<?php
if(!empty($_GET["valor"]))
{
// O CAMPO VALOR CONTERÁ O QUE O USUARIO DIGITOU ATÉ O
MOMENTO..
// CONECTA AO BANCO COLOCA PARAMENTROS IP,USUARIO,SENHA
$conexao=mysql_connect("localhost","root","1010");

//SELECIONA O BANCO DE DADOS QUE VAI USAR
mysql_select_db("pessoas");

```

```
// EXECUTA A INSTRUÇÃO SELECT PASSANDO O QUE O USUARIO  
DIGITOU
```

```
$sql="select * from pessoa where nome like '$_GET[valor]%'";  
$resultado=mysql_query($sql) or die (mysql_error());
```

```
//VERIFICA A QUANTIDADE DE REGISTROS RETORNADOS  
$linhas=mysql_num_rows($resultado);
```

```
if($linhas>0){  
//EXECUTA UM LOOP PARA MOSTRAR OS NOMES DAS PESSOAS  
// VALE LEMBRAR QUE TODOS ESSES RESULTADOS SERAO  
MOSTRADOS DENTRO DA PAGINA INDEX.PHP  
// DENTRO DO DIV 'PAGINA'
```

```
while($pegar=mysql_fetch_array($resultado))  
echo "$pegar[nome] <br>";  
}
```

```
}  
?>
```

# Flash + xml + PHP

---

Primeiramente vamos ver quais os elementos e as funções principais de um player:

Um player tem as funções play, stop, pause, avançar, controle de volume e ele exibe também o volume, o nome do artista e o nome da música.

As funções de play, stop, pause podem ser feitas utilizando os principais comandos da classe Sound() que são o play, stop.

Já o controle de volume pode ser ajustado através o getVolume e do setVolume.

E para resgatarmos o nome da música e do artista usaremos o ID3.

## Mas o que é ID3?

ID3 é um conjunto de informações que fica alocado numa mp3 contendo o nome da música o álbum, o artista e etc.

O flash resgata essas informações através de comandos simples:

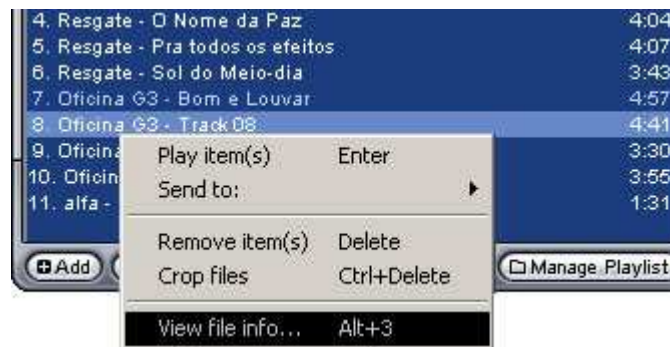
`objeto_da_classe_sound.id3.propriedade;`

| propriedade | descrição        |
|-------------|------------------|
| album       | álbum da música  |
| genre       | gênero da música |
| songname    | nome da música   |
| artist      | nome do artista  |

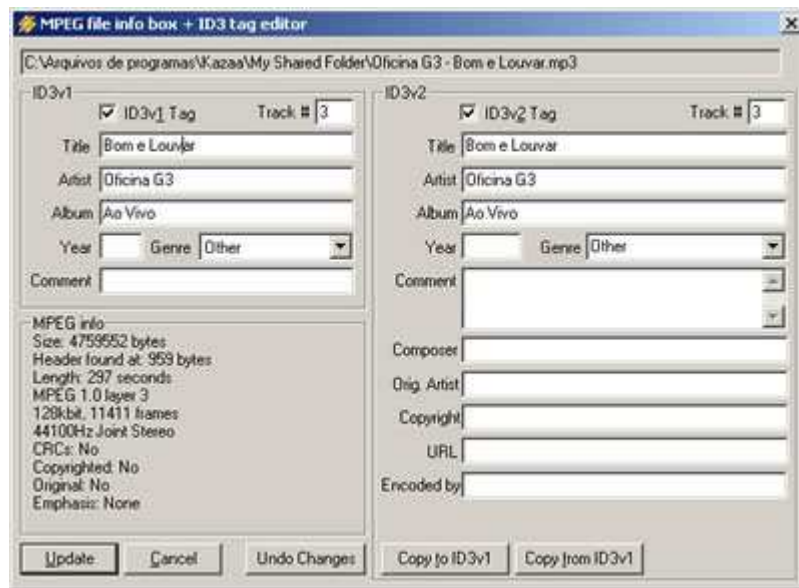
Só tem um problema ao se trabalhar com a id3, se os dados não estiverem configurados, não aparecerá nada no nosso player.

## Como configurar a id3 da minha MP3?

O Winamp, altera e adiciona esses dados a um mp3. Para configurar a id3 de sua mp3, abra o winamp, clique com o botão direito no nome da mp3 que você quer configurar e vá em “view file info (alt+3)” como mostra a figura abaixo:



Daí é só alterar os dados como mostra a figura abaixo e clicar em update para finalizar.



Agora que já está tudo esclarecido sobre id 3 e o que fará o nosso player, vamos aos objetos necessários para o player.

Para criar o player usaremos 6 botões: um para stop, um para play, um para pause, um para avançar e dois para controle de volume. E precisaremos também e 3 campos de texto do tipo dinâmico: para mostrar o volume, o nome da música e o nome do artista. Como mostra a figura abaixo:



Depois instanciamos os botões com as seguintes instâncias:

- botão de play = play
- botão de stop = stop
- botão de pause = pause
- botão avançar = ff
- botão para aumentar volume = v1
- botão para diminuir volume = v2

Para instanciar um botão é só seleciona-lo, ir em propriedades e no campo instance name digitar a instância.

Coloque instâncias nos campos de texto também

- campo para o nome do artista = artista
- campo para o nome da música = musica
- campo para o volume = v3

Bom, depois dos elementos do player criados e instanciados, crie outra camada, renomeie ela para acoes, abra o painel actions (f9) e vamos para a programação.

Vamos começar declarando as variáveis a serem usadas, no formato definido pelo as 2.0, sendo “var variavel:tipo\_da\_variavel”:

```
//objeto que ira manipular o som
var som:Sound = new Sound();
//vetor que ira armazenar o caminho das mp3 a serem tocadas
```

```

var musicas:Array = new Array();
//declarando o resto das variaveis
var i, vol, verifica, pos:Number;
var ligado:Boolean;
var acao:String;
agora vamos atribuir ao vetor o caminho das mp3 a serem utilizadas:
stop();
musicas[0] = "audio/audio1.mp3";
musicas[1] = "audio/audio2.mp3";
musicas[2] = "audio/audio3.mp3";
vamos fazer um if que ira verificar se o player ja esta tocando se ele não estiver ligado,
ele definira o volume como máximo e a posição inicial para a mp3 a ser tocada:
if (ligado==null) {
i = 0;
vol = 100;
ligado = false;
}

```

Na rotina abaixo, especificamos que quando iniciar o frame o volume do som sera o valor setado na variavel vol.

```

this.onEnterFrame = function() {
som.setVolume(vol);

```

No if abaixo verificamos se o player esta ativado e ele estiver desativado, ele ira carregar a musica a ser tocada.

```

if (ligado==false) {
som.loadSound(musicas[i], true);

```

Como estamos usando apenas 3 musicas nesse player nosso vetor vai da posição 0 ate 2 por isso fiz a rotina abaixo que faz com que o valor de i, que é a variável que esta controlando a música a ser tocada, não passe de 2.

```

if (i<2)

```

```

i++;

```

```

else

```

```

i = 0;

```

agora definimos que o player esta ligado e fechamos o 1º if

```

ligado = true;

```

```

}

```

Através do id3, pegamos o nome do artista e da música e colocamos esses dados nos campos de texto que instanciamos acima.

```

artista.text = "Artista: " + som.id3.artist;

```

```

musica.text = "Musica: " + som.id3.songname;

```

A rotina abaixo pega o volume do som através da propriedade getVolume(); da classe sound, e verifica se ele é maior que o valor máximo (100), e se for ele deixa o volume com o valor máximo, ele também verifica se esta abaixo do valor mínimo (0) e se estiver ele atribui ao som através do método setVolume o valor mínimo ao volume. E no final ele atribui ao campo de texto v3 o valor do volume atual e fecha a função.

```

verifica = som.getVolume();

```

```

if (verifica>=100) {

```

```

som.setVolume(100);

```

```

}

```

```

if (verifica<=0) {

```

```
som.setVolume(0);
}
v3.text = som.getVolume() + "%";
}
```

Nos comandos abaixo através do evento onSoundComplete eu aviso ao flash que o som já foi finalizado, fazendo com que a próxima música seja tocada.

```
som.onSoundComplete = function() {
ligado = false;
}
```

Agora vamos as funções de cada botão:

Obs: devem serem digitadas juntos com as actions acima no 1º frame da camada ação.

No botão que aumenta o volume (v1) eu pego o valor do volume e adiciono + 5. E no botão que diminui o volume eu faço o contrário, pegando o volume e diminuindo por 5.

//aumenta o volume

```
v1.onRelease = function() {
verifica = som.getVolume();
vol = verifica+5;
}
```

// diminui o volume

```
v2.onRelease = function() {
verifica = som.getVolume();
vol = verifica-5;
}
```

No botão stop eu apenas uso o método stop para parar o som;

```
// para musica
stop.onRelease = function() {
som.stop();
}
```

Na ação do botão pause eu atribuo a variável ação o valor "pause" para eu saber quando está sendo pausado e através da propriedade position eu pego o valor da posição atual em que a música está e armazeno na variável pos que será utilizada futuramente e paro a música através do método stop.

// pausa a musica

```
pause.onRelease = function() {
acao = "pause";
pos = som.position/1000;
som.stop();
}
```

Para o botão play eu usei um if para saber se devemos continuar a música ou se devemos apenas iniciá-la desde o começo. Por isso avia criado a variável ação, se o valor dela for "pause" irei continuar a música desde a parte em que ela foi parada senão irei iniciá-la desde o começo:

```
play.onRelease = function() {
if (acao=="pause") {
som.start(pos);
acao="";
} else {
som.start();
}
}
```

E finalmente o botão avançar. Sua ação é bem simples, apenas paro o som e defino a variável ligado como false, falando para o flash ir para a próxima musica.

```
ff.onRelease = function() {  
ligado = false;  
som.stop();  
}
```

Como viram eu usei 3 mp3 nesse player mas você pode utilizar quantas quiser. Para isso é só adicionar mais posições para o vetor ex:

```
musicas[3] = "audio/audio4.mp3";  
musicas[4] = "audio/audio5.mp3";
```

e na parte em que é verificado se o i (variável que controla a posição do vetor) esta dentro do limite, no caso do player era 2 agora com essa alteração passa a ser 4.

```
if (i<4)  
i++;  
else  
i = 0;
```

Então aqui vocês aprenderam como fazer um player, se aprofundando um pouco mais no as 2.0 vendo como as váriaveis são declaradas nessa nova versão do actionscript. Foi visto também vários métodos da classe sound, inclusive o ID3. Após tudo isso o resultado final deve ser o seguinte: