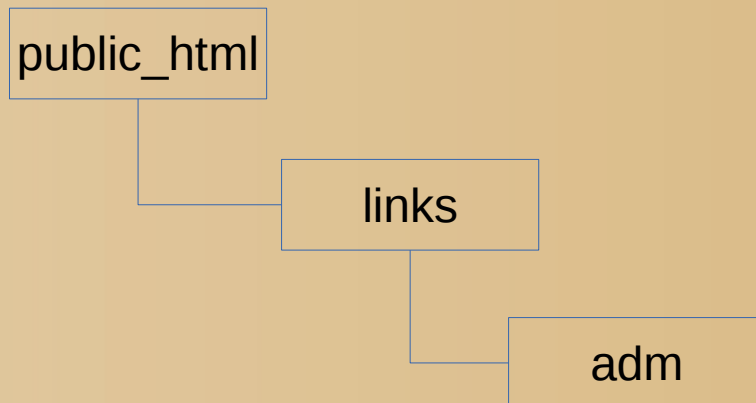


# Controle de acesso

Construiremos um sistema simples para cadastro de links. O sistema terá uma página pública (onde serão exibidos os links) e uma área administrativa, onde os usuários cadastrados poderão incluir novos links. Os usuários serão agrupados em níveis, sendo os usuários de nível 0 os administradores que poderão cadastrar outros usuários.

# Preparando o ambiente

Dentro do seu diretório virtual (public\_html, www, htdocs) crie uma pasta chamada **links** – esta pasta será a raiz do nosso sistema. Dentro dela, crie uma pasta chamada **adm**

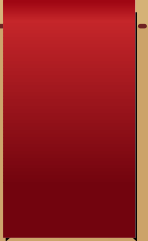


# Preparando o ambiente

Nosso próximo passo é criar o banco de dados, como não é nosso foco tal matéria, vou me limitar a passar os comandos necessários (e ter esperança que você sabe onde deve digita-los.) - Qualquer dúvida, utilize os comentários.

- create database sLinks;
- Use sLinks;
- create table usuarios (usuario varchar(200) primary key, senha varchar(200), level int default 1, nome varchar(200));

# Preparando o ambiente



Banco de dados pronto, agora vamos cadastrar dois usuários para podermos efetuar os primeiros testes. Você pode alterar os dados livremente.

- insert into usuarios values ('admin','admin',0,'José adolfo filho'), ('usuario','usuario',1,'Anderson Salomé');

```
mysql> select * from usuarios;
+-----+-----+-----+-----+
| usuario | senha   | level | nome           |
+-----+-----+-----+-----+
| admin   | admin   | 0     | José adolfo filho |
| usuario | usuario | 1     | Anderson Salomé   |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```



# Agora, vamos ao código:

Para facilitar o entendimento, dividi o código no maior número de páginas possíveis, o nome da página está sempre como comentário na primeira linha.

A princípio, faremos a parte administrativa, ou seja, a área onde o administrador poderá cadastrar novos usuários e os usuários poderão cadastrar novos links (todas as páginas devem ser salvas dentro da pasta adm).

# Cabeçalho

Para facilitar a utilização (futura) de scripts e css, criei uma página com o cabeçalho HTML padrão:

```
<!-- cabecalho.php -->
<!DOCTYPE html>
<html lang="pt-br">
<head>
<META HTTP-EQUIV="CACHE-CONTROL" CONTENT="NO-CACHE">
<META HTTP-EQUIV="PRAGMA" CONTENT="NO-CACHE">
<meta name="keywords" content="" />
<meta name="description" content="" />
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>SisLinks </title>
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon" />

</head>

<body>
```

# Cabeçalho

Como a página só possui código HTML básico, não vou comentá-la aqui (presencialmente eu devo acabar falando alguma coisa... afinal, eu sou bonzinho (vote larback)).

# Formulário de login

```
1 <?php
2 //formLogin.php
3 if ($_SESSION['errosL']>=5)
4     die("Você fez tentativas demais, aguarde alguns segundos antes de tentar novamente.");
5 ?>
6 <form action="index.php" method="post">
7     <input type="hidden" name="acao_interna" value="login">
8     <table align=center >
9         <tr bgcolor=#eeeeee ><td colspan=2 align=center>SisLinks - Controle de acesso</td></tr>
10        <tr><td>Usuário</td><td><input type="text" name="usuario" /></td></tr>
11        <tr><td>Senha</td><td><input type="password" name="senha" /></td></tr>
12        <tr><td colspan=2 align=right ><input type="submit" value="Acessar" /></td></tr>
13    </table>
14 </form>
15
```

A única novidade aqui é o controle de tentativas, na página index.php, toda vez que um login é tentado sem sucesso, a variável de sessão errosL é incrementada. No formulário de login, antes de montarmos o formulário, conferimos se o usuário já esgotou as tentativas. Caso tenha esgotado, paramos a execução do script e solicitamos ao mesmo que aguarde.



# Conexão com o banco de dados

```
1 <?php
2 //conSLinks.php
3 mysql_connect("127.0.0.1","root","*****") or die(mysql_error());
4 mysql_select_db("sLinks") or die(mysql_error());
5 ?>
```

Esta página é responsável por fazer a conexão com o banco de dados... mas você já tinha visto isso ao ler o código... eu poderia comentar que o `mysql_connect` faz a conexão com o servidor mysql e que o `mysql_select_db` seleciona o banco de dados que iremos utilizar... mas sinceramente, é desnecessário né? Então vou aproveitar o espaço e fazer uma propaganda sobre o 'print screen com o mouse' – você já viu? Se ainda não viu assim que terminar seus estudos, acesse o link: <http://www.youtube.com/watch?v=KaTZ92Xjn3Y>

# Arquivo de funções

```
1 <?php
2 //funcoes.php
3 function confereLogin($u,$s) {
4     $rs = mysql_query("select * from usuarios where usuario='$u' and senha='$s'");
5     if (mysql_num_rows($rs)>0){
6         $ln = mysql_fetch_assoc($rs);
7         $_SESSION['usuario'] = $ln['usuario'];
8         $_SESSION['nome'] = $ln['nome'];
9         $_SESSION['level']=$ln['level'];
10        return true;
11
12    }else
13        return false;
14 }
15
16 function conferePermissao() {
17     if (!$_SESSION['logado'])
18         die ("Acesso negado");
19 }
20 ?>
```

# Arquivo de funções

O arquivo de funções possui duas funções.

A primeira é responsável por conferir se o usuário e senha informados como parâmetros estão cadastrados no banco de dados. Se estiverem, as informações do usuário são gravadas em variáveis de sessão e é retornado o valor lógico true. Se os dados não corresponderem a nenhuma ocorrência da tabela usuários, é retornado o valor lógico false.

A segunda função confere se a variável de sessão 'logado' é diferente de verdadeiro, se esta condição for verdadeira (ou seja, a variável 'logado' é falsa) o script é encerrado com a função die exibindo a mensagem “Acesso negado”.

A primeira função é utilizada no login do usuário (a chamada será feita na página index.php) e a segunda será usada em todas as subpáginas, para evitar que usuários não logados tenham acesso ao sistema.

# Página principal – index.php

Esta página será o centro da nossa área administrativa. Todos os links apontarão para ela e ela fará as execuções necessárias. Seu código é um pouco extenso (e vai aumentar quando dermos prosseguimento as demais funções do sistema) e por isso eu vou separá-la em várias partes. Não prossiga sem ter entendido completamente.

```
1 <?php
2 //index.php
3 error_reporting(0);
4 session_start();
5 if (!isset($_SESSION['errosL']))
6     $_SESSION['errosL']=0;
7
```

A linha 3 desativa a exibição de mensagens de erro. O ideal é que essa linha seja comentada até o projeto está pronto e funcional (pois se ocorrer um erro, este não será exibido). Se você desconhece a função `error_reporting`, leia:

[http://php.net/manual/pt\\_BR/function.error-reporting.php](http://php.net/manual/pt_BR/function.error-reporting.php)

A linha 4 inicia a sessão. Sem iniciar a sessão não se pode utilizar variáveis de sessão (e nós utilizamos muitas).

A linha 5 confere se a variável de sessão `errosL` não existe, e se ele não existir, inicializa a mesma com o valor 0 (zero) – É esta variável que utilizamos no formulário de login.

# Página principal – index.php

```
8 include "conSLinks.php";
9 include "funcoes.php";
10 include "cabecalho.php";
11
12 if(!isset($_REQUEST['acao_interna']))
13     $acao_interna="inicio";
14 else
15     $acao_interna=$_REQUEST['acao_interna'];
```

As linhas 8, 9 e 10 incluem os arquivos inicialmente necessários (eles já foram vistos – se necessário volte alguns slides).

A linha 12 confere se não existe uma variável chamada `acao_interna` em alguma requisição HTTP. O `$_REQUEST` é utilizado para que o PHP procure nas três formas de envio `$_GET`, `$_POST` e `$_COOKIE`. Se não existir, a variável local de mesmo nome, recebe o valor `inicio`. Se existir, a variável recebe o valor que foi enviado.

É esta variável `acao_interna` que será utilizada para sabermos qual ação deve ser executada pela página `index.php`

Mais sobre o `$_REQUEST` em:

[http://php.net/manual/pt\\_BR/reserved.variables.request.php](http://php.net/manual/pt_BR/reserved.variables.request.php)

# Página principal – index.php

```
16  
17 if ((!$_SESSION['logado']) && ($acao_interna!="login"))  
18     $acao_interna = "telaLogin";  
19
```

Como estamos na área administrativa do site, o usuário só pode fazer qualquer ação se estiver logado. A única ação que ele pode executar se não estiver logado, é o login.

Assim, na linha 17 temos um SE para garantir isso. Se a variável de sessão logado não for true – ela recebe true no momento em que o usuário faz o login – e a ação\_interna for diferente de login, então a acao\_interna recebe telaLogin.

Na continuação do código você verá que este valor “telaLogin” na variável acao\_interna redireciona a aplicação para o formulário de login, evitando que pessoas não logadas utilizem o sistema.



# Página principal – index.php

```
20 switch($acao_interna) {  
21     case "telaLogin":  
22         include "formLogin.php";  
23         break;
```

Para sabermos o que o usuário deseja fazer, conferimos o valor da variável `acao_interna`. Como ela pode possuir vários valores, optei por utilizar a estrutura `switch` (que você deve lembrar (boas lembranças) como estrutura caso).

O primeiro teste confere se o valor é `telaLogin`, se for, é incluído o arquivo `formLogin`.

Repare que no final de cada opção do `switch` é necessário colocar o comando `break`, se esquecermos de colocar essa interrupção, todas as opções à partir da primeira opção verdadeira serão executadas.

Para mais sobre o `switch`, leia:

[http://us2.php.net/manual/pt\\_BR/control-structures.switch.php](http://us2.php.net/manual/pt_BR/control-structures.switch.php)

# Página principal – index.php

```
24     case "login":
25         $usuario = $_POST['usuario'];
26         $senha = $_POST['senha'];
27         $log = confereLogin($usuario,$senha);
28         if ($log){
29             $_SESSION['logado']=true;
30             include "menu.php";
31         }else{
32             $_SESSION['errosL']++;
33             $_SESSION['logado']=false;
34             include "formLogin.php";
35             echo "Usuário ou senha inválidos. Confira e tente
36                 novamente.<br /><br />Tentativa " .$_SESSION['errosL']. " de 5";
37         }
38         break;
```

O segundo teste confere se o valor é login – se você voltar no formulário de login, verá que a variável `acao_interna` é um campo do tipo hidden (oculto) e que possui justamente este valor, ou seja, o valor será login quando a solicitação tiver vindo do formulário de login.



# Página principal – index.php

Conferido o valor, recuperamos as variáveis enviadas pelo formulário de login – volte nele quantas vezes forem necessárias – e as colocamos em variáveis locais (linhas 25 e 26).

A variável \$log recebe o retorno da função confereLogin, passando como parâmetro o usuário e a senha informados. A função confereLogin(user,pass) – escrita no arquivo de funções – faz uma pesquisa na tabela de usuário procurando uma ocorrência correspondente. Ao encontrar, as variáveis de sessão com os dados do usuário são preenchidas e é retornado o valor true. Se nenhuma ocorrência for encontrada, é retornado o valor false.

Na linha 28, um SE confere se o valor de \$log é true, se for, a variável de sessão logado recebe true e é incluído o arquivo “menu.php” - que ainda veremos.

Se o valor de \$log não for true, contamos a tentativa (variável errosL), a variável de sessão logado recebe falso, incluímos o formulário de login (para que o usuário tente novamente) e imprimimos uma mensagem avisando do erro.

# Página principal – index.php

```
38     case "inicio":
39         include "menu.php";
40         break;
41     case "logoff":
42         unset($_SESSION['logado'], $_SESSION['errosL']);
43         echo "Sessão encerrada";
44         break;
45     }
46     ?>
```

Na linha 38, conferimos se o valor é inicio – algo como um botão home – e se for incluimos o menu.php – estamos quase nele.

Na linha 41, conferimos se o valor é logoff – se for, destruimos a variável de sessão logado e errosL – que utilizamos para controlar o login e exibimos uma mensagem avisando que sessão foi encerrada.

# Menu de controle – menu.php

```
1 <?php
2 //menu.php
3 conferePermissao();
4 echo "<table width='100%'>";
5 echo "<tr><td colspan=2 align='right'>Bem vindo ".$_SESSION['nome']."</td></tr>";
6 echo "<tr>";
7 if ($_SESSION['level']==0)
8     echo "<td><a href='#'>Cadastro de usuários</a></td>";
9 echo "<td><a href='#'>Cadastro de links</a></td>";
10 echo "<td><a href='index.php?acao_interna=logoff'>Sair</td>";
11 echo "</tr>";
12 ?>
```

Muito simples (e essencial) este arquivo. Lembra-se que quando o usuário faz o logoff criamos variáveis de sessão para armazenar seus dados? Pois bem, agora pegamos o nome do usuário e imprimimos uma mensagem de boas vindas – antes, na linha 3, conferimos se o usuário está logado (função conferePermissao()).

Estando o usuário logado e devidamente recebido, criamos o menu de opções. Como definimos no escopo que só administradores podem cadastrar outros usuários, este link só aparece se o level for 0.

# Game over

Bom é isso, a parte do login está pronta e eu espero que você tenha entendido. Conforme eu for montando o resto do sistema (em minhas aulas presenciais) eu vou postando e, se tudo der certo, no final teremos um sistema completo – não muito útil, mas completo.

Qualquer dúvida, fiquem a vontade para perguntar.

Espero que seja útil.

Até mais e obrigado pelos peixes.

P.S<sup>1</sup>

Não deixe de seguir os links que foram passados ao longo da aula.

P.S<sup>2</sup>

Se não entendeu o “Até mais e obrigado pelos peixes” leia O Guia dos Mochileiros da Galáxia de Douglas Adams.