

LAB 6 – Autenticação em PHP

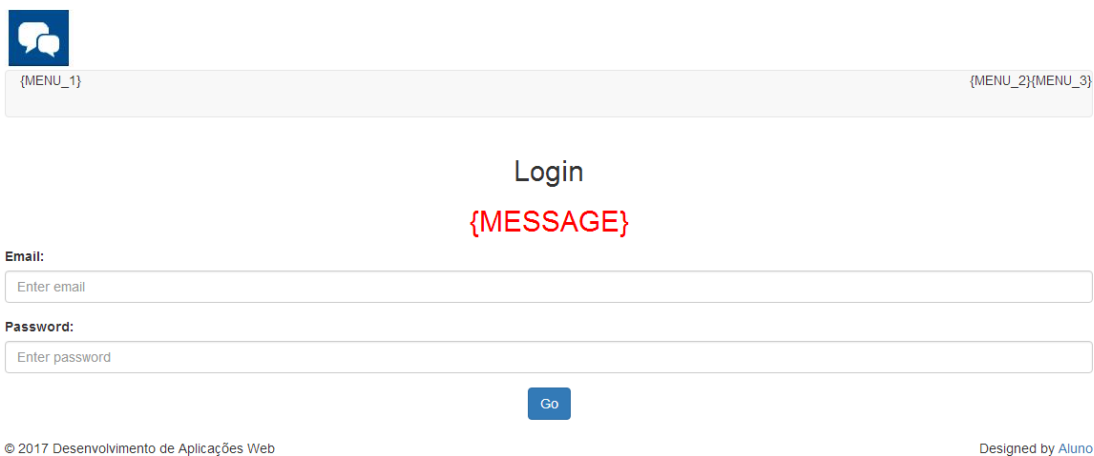
1. Assuma-se neste lab que concluiu com sucesso o LAB4 e o LAB5 (e sabe utilizar a *template engine* SMARTY) . Copie todos os ficheiros da pasta “LAB5” para a pasta “LAB6”:

```
a12345@daw2:~$ cd public_html
```

```
a12345@daw2:~/public_html$ cp -R LAB5 LAB6
```

```
a12345@daw2:~/public_html$ chmod g+w LAB6/templates_c
```

2. Construa o formulário (“form”) `login_template.tpl` a partir do template `login_template.html` (que realizou no LAB2 ou no LAB3 à escolha):



The image shows a login form template with the following elements:

- A blue speech bubble icon in the top left corner.
- A navigation bar with three placeholders: `{MENU_1}`, `{MENU_2}`, and `{MENU_3}`.
- The word "Login" centered below the navigation bar.
- A red placeholder `{MESSAGE}` for an error message.
- An "Email:" label followed by a text input field with the placeholder text "Enter email".
- A "Password:" label followed by a text input field with the placeholder text "Enter password".
- A blue "Go" button below the password field.
- Copyright text "© 2017 Desenvolvimento de Aplicações Web" in the bottom left.
- Text "Designed by Aluno" in the bottom right.

Note que o template tem quatro “placeholders”

- `{MENU_1}` é um link (“home”) para “index.php”
 - `{MENU_2}` é um self-link (“login”)
 - `{MENU_3}` é um link (“register”) para “register.php”
 - `{MESSAGE}` é um placeholder para a mensagem em caso de erro (“Wrong email or password”)
-
- O código PHP que controla este template deverá encontrar-se em “login.php”
 - A acção deverá ser `action="login_action.php"` e o método deverá ser `method="post"`.

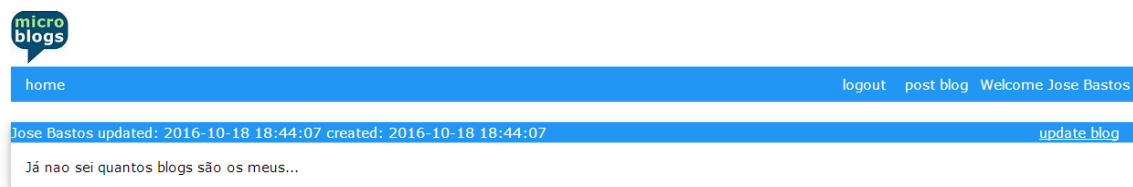
2. Construa o controlador `login_action.php` que vai consultar a base de dados para validar o login do utilizador:

- em caso de sucesso:
 - cria um **“cookie” de sessão** e regista no array correspondente `$_SESSION` os dados “id”, “email” e “name” do utilizador retirados da base de dados
 - re-direcciona para o controlador `message.php`, e este envia para o browser a mensagem "Welcome back!". Utilize a variável de sessão `message` para comunicar a mensagem.
- em caso de insucesso
 - re-direcciona novamente para a página `login.php`.
 - envia o email que foi introduzido de volta ao controlador `login.php` através da variável de sessão `email`
 - envia a mensagem "Wrong email or password" de volta ao controlador `login.php` através da variável de sessão `message`

SUGESTÃO: A partir deste momento pode reescrever o seu código nos controladores `register.php` e `register_action.php` para também eles utilizarem variáveis de sessão para passar a informação entre eles

NOTA: A partir deste lab **não é permitido** utilizar dados embutidos no URL para passar informação entre controladores; **é obrigatório** utilizar variáveis de sessão!

3. Actualize o controlador `index.php` para



- ter acesso ao cookie da sessão e, em caso de login com sucesso,
- actualize o placeholder `{WELCOME}` em `index_template.tpl` com o nome do utilizador
- actualize o placeholder `{MENU_2}` em `index_template.tpl` com um link para `logout_action.php`
- actualize o placeholder `{MENU_3}` em `index_template.tpl` com um link para `blog.php`

- todos os blogs da autoria do utilizador (e apenas estes) deverão ter o placeholder {UPDATE}¹ com um link para `blog.php?micropost_id=xx` (onde “xx” é o id do blog)

4. Desenhe o controlador “`message.php`” e o template “`message_template.tpl`”



{MESSAGE}

© 2016 Desenvolvimento de Aplicações Web

Designed by Aluno

Note que este template tem um placeholder {MESSAGE} e tem um “meta tag” que faz redirect para a página “`index.php`” do portal ao fim de 3 segundos.

Construa o controlador `logout_action.php` que

- encerra (destroi) a sessão
- coloca em {MESSAGE} “See you back soon!”

5. Faça o upload dos ficheiros

- `index.php`
- `login.php`
- `login_action.php`
- `logout_action.php`
- `message.php`
- `model.php` (opcional)

para a pasta “LAB6”

Faça o upload dos templates

- `login_template.tpl`

¹ Se desejar, em lugar do placeholder, pode usar a estrutura SMARTY `{if} {else} {/if}`

- message_template.tpl
- index_template.tpl²

para a pasta "LAB6/templates"

Considere o lab concluído quando tiver reproduzido a funcionalidade de

http://all.deei.fct.ualg.pt/~a999990/smarty_exam2/login.php

no seu site web pessoal no servidor "daw2" (<http://daw.deei.fct.ualg.pt>)

REFERÊNCIAS:

- <http://intranet.deei.fct.ualg.pt/DAW/auth-db-sessions/files.html>

² Só no caso se alterou o template para introduzir a estrutura SMARTY {if} {else} {/if}

ANEXO 1: Acesso à base de dados MySQL

- O acesso à base de dados MySQL na rede UALG pode ser feita em linha de comando

```
a12345@daw2:~$mysql -u a12345 -p -h 10.10.23.184 db_a12345
```

ou ainda utilizando o software **phpMyAdmin** disponível no URL

<http://daw.deei.fct.ualg.pt/phpMyAdmin>

ANEXO 2: estrutura da base de dados

```
CREATE TABLE `users` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(255) default NULL,  
  `email` varchar(255) default NULL,  
  `created_at` datetime NOT NULL,  
  `updated_at` datetime NOT NULL,  
  `password_digest` varchar(255) default NULL,  
  `remember_digest` varchar(255) default NULL,  
  `admin` tinyint(1) default NULL,  
  `activation_digest` varchar(255) default NULL,  
  `activated` tinyint(1) default NULL,  
  `activated_at` datetime default NULL,  
  `reset_digest` varchar(255) default NULL,  
  `reset_sent_at` datetime default NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `index_users_on_email` (`email`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
CREATE TABLE `microposts` (  
  `id` int(11) NOT NULL auto_increment,  
  `content` text,  
  `user_id` int(11) default NULL,  
  `created_at` datetime NOT NULL,  
  `updated_at` datetime NOT NULL,  
  `likes` int(11) NOT NULL DEFAULT '0',  
  PRIMARY KEY (`id`),  
  KEY `fk_user_id` (`user_id`),  
  CONSTRAINT `fk_user_id` FOREIGN KEY (`user_id`)  
  REFERENCES `users` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

ANEXO 3: OPCIONAL

Este guião realiza inteiramente o lab no servidor de produção mas, se desejar, pode realizar o lab no seu PC/portátil. As instruções para a instalação de um ambiente de desenvolvimento "XAMPP" no seu PC/portátil encontram-se em

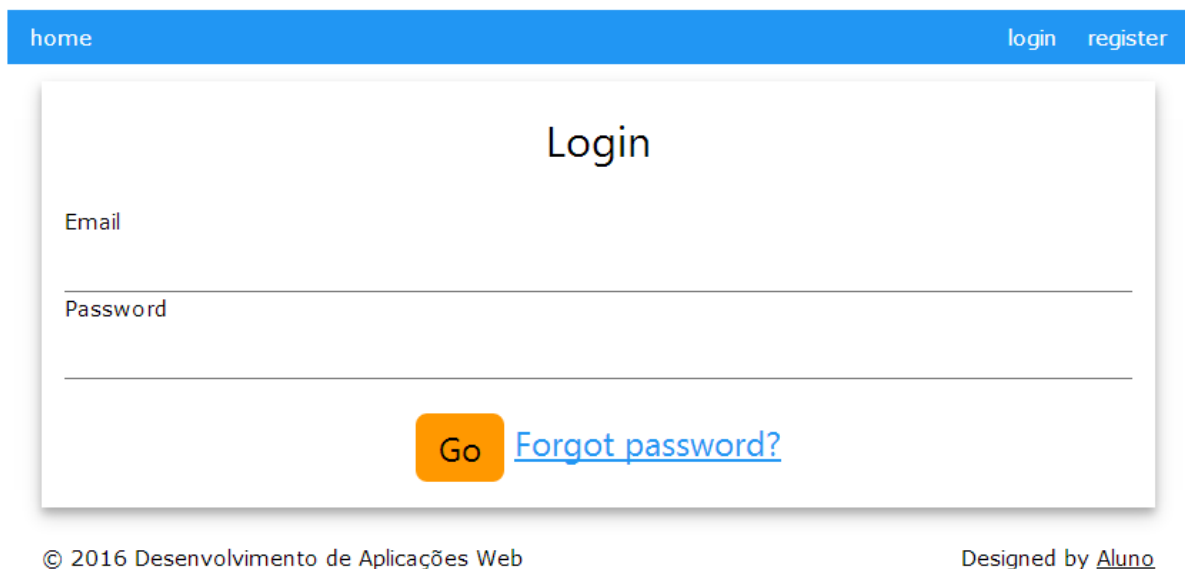
<http://intranet.deei.fct.ualg.pt/IPM/XAMPP.pdf>

ANEXO 4: OPCIONAL

FUNCIONALIDADE “RECOVER PASSWORD”

Pretende-se implementar a funcionalidade “recover password” através do envio de um email contendo um URL (“link”) apontando para uma página web onde se pode escolher uma nova password.

A página de “login.php”



The screenshot shows a web page with a blue header bar containing the text 'home' on the left and 'login register' on the right. Below the header is a white box with the title 'Login' centered at the top. Inside the box, there are two input fields: 'Email' and 'Password'. Below the 'Password' field is an orange button with the text 'Go' and a blue link labeled 'Forgot password?'. At the bottom of the page, there is a copyright notice '© 2016 Desenvolvimento de Aplicações Web' on the left and 'Designed by Aluno' on the right.

contem agora um link,

```
<a href="password_reset.php" >Forgot password?</a>
```

para a página “password_reset.php”,

Password reset

Email

Go

que contem um formulário onde o utilizador vai colocar o email com que se registou no site.

O formulario utiliza o método “POST” para enviar dados ao servidor:

```
<form method="post" action="password_reset_action.php">
```

1. Construa o programa **password_reset_action.php** que:

- verifica se o email introduzido consta da base de dados
- em caso de sucesso
 - Cria um token “md5” do tempo actual

```
$reset_digest = substr(md5(time()),0,32);
```
 - guarda este valor na base de dados (tabela users, coluna reset_digest), bem como o tempo atual (tabela users, coluna reset_sent_at)
 - envia ao utilizador registado um email personalizado com o assunto “Password reset” e com o texto

Olá Sr.(a) José Silva

Para obter uma nova password clique no link

http://daw.deei.fct.ua1g.pt/~a12345/LAB6/new_password.php?token=fad9bc3391f298e6186f8a

Se NÃO pediu uma nova password IGNORE este email.

Cumprimentos,

webmaster!

Página web: <http://intranet.deei.fct.ua1g.pt/DAW/>

E-mail: webmaster@deei.fct.ua1g.pt

NOTA: Não responda a este email, não vai obter resposta!

onde "José Silva" é o nome do utilizador obtido na base de dados e "fad9bc3391f298e6186f8a"

O texto do email contém um link para a página "new_password.php" (substituir 12345 pelo seu número de aluno) com o valor do token embutido

- envia para o browser a mensagem

Password reset activated! Email sent to you :-)

Nota: tem de utilizar um email válido a que tenha acesso, para comprovar o correto funcionamento do programa!

2. O formulário new_password.php

home login register

New password

Password

Password confirmation

Go Clear

© 2016 Desenvolvimento de Aplicações Web Designed by [Aluno](#)

utiliza o método "POST" para enviar a nova password ao servidor, bem como o token recebido (como um input do tipo "hidden")

```
<form method="post" action="new_password_action.php">  
<input type="hidden" name="token" value="{TOKEN}">
```

3. Crie o programa **new_password_action.php** que vai actualizar a base de dados com a nova password do utilizador.

- verifica se o token recebido existe na base de dados
- em caso de sucesso e se não passou mais de uma hora entre a hora actual e a hora de envio do email
 - encripta e atualiza a password na base de dados
 - envia para o browser uma mensagem

Password reset successfully!

- em caso de insucesso envia para o browser a mensagem

ERROR: WRONG TOKEN OR TOKEN EXPIRED, PASSWORD RESET FAILED!