

# LAB 6 – Autenticação em PHP

---

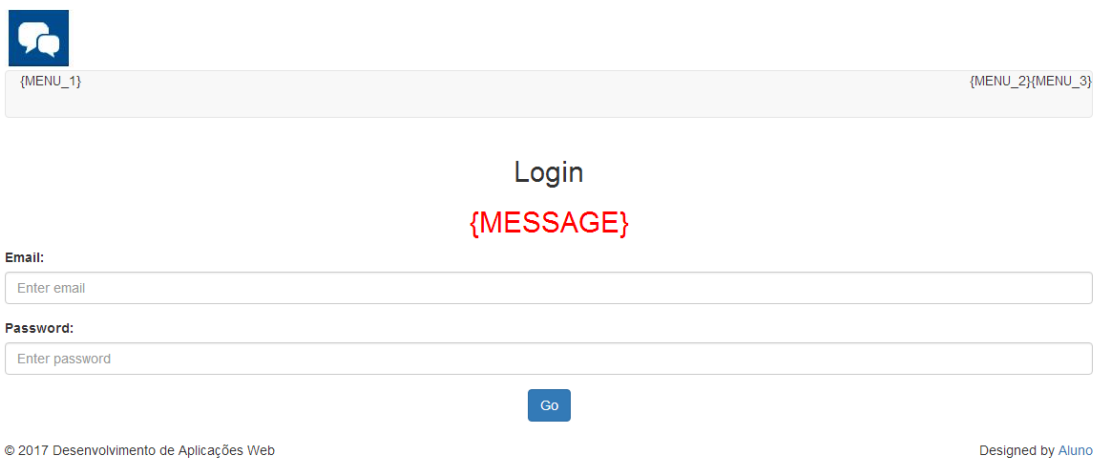
1. Assuma-se neste lab que concluiu com sucesso o LAB4 e o LAB5 (e sabe utilizar a *template engine* SMARTY) . Copie todos os ficheiros da pasta “LAB5” para a pasta “LAB6”:

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

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

```
a12345@daw:~/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. At the top left is a blue speech bubble icon. Below it is a horizontal bar containing the placeholder `{MENU_1}` on the left and `{MENU_2}{MENU_3}` on the right. In the center, the word "Login" is displayed above a red placeholder `{MESSAGE}`. Below this are two input fields: "Email:" with the placeholder "Enter email" and "Password:" with the placeholder "Enter password". A blue "Go" button is centered below the password field. At the bottom left, it says "© 2017 Desenvolvimento de Aplicações Web" and at the bottom right, "Designed by Aluno".

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"`.

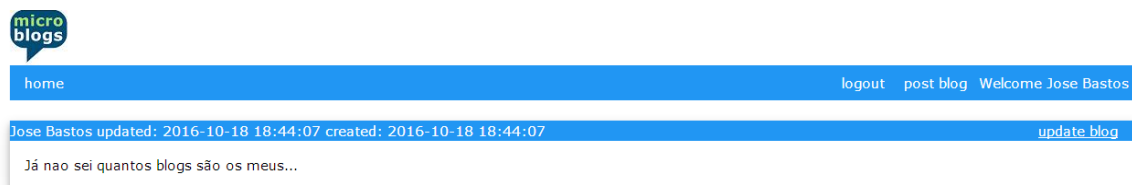
2. Construa o controlador `login_action.php` que vai consultar a base de dados para validar o login do utilizador:

- 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}<sup>1</sup> 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`

para a pasta “LAB6”

Faça o upload dos templates

- `login_template.tpl`
- `message_template.tpl`
- `index_template.tpl`<sup>2</sup>

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

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

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

no seu site web pessoal

## **REFERÊNCIAS:**

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

## **ANEXO 1** Acesso à base de dados MySQL

- O acesso à base de dados MySQL na rede UALG pode ser feita utilizando um cliente gráfico à sua escolha (por exemplo <http://www.heidisql.com/>),

ou em linha de comando

```
a12345@daw:~$mysql -u a12345 -p -h 10.10.23.183 db_a12345
```

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

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

na rede UALG e na **INTERNET**

## 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,  
  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;
```