

# LAB 5 – Formulários com PHP

1. Assuma-se aqui que completou com sucesso o LAB 4. Copie os ficheiros `index_template.tpl`, `index.php`, `db.php`, etc, da pasta “LAB4” para a pasta “LAB5”:

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

```
a12345@daw2:~/public_html$ cp -R LAB4 LAB5
```

```
a12345@daw2:~/public_html$ chmod g+w LAB5/templates_c
```

2. Construa o template SMARTY `register_template.tpl` com o auxílio do template `register_template.html` que desenhou no LAB2 ou no LAB3 (à sua escolha)

NOTA: Os *screenshots* apresentados são ilustrativos. Utilize o seu *layout* que idealizou no LAB2 ou no LAB3; apenas é requerido que os templates tenham (no mínimo) os campos/placeholders que aparecem no *screenshot*

The screenshot shows a web form titled "Sign up" with a blue header bar containing placeholders {MENU\_1}, {MENU\_2}, and {MENU\_3}. Below the header is a red bar with a placeholder {MESSAGE}. The form itself is white and contains the following elements:

- Text label: Name
- Input field: {NAME}
- Text label: Email
- Input field: {EMAIL}
- Text label: Password
- Input field: (empty)
- Text label: Password confirmation
- Input field: (empty)
- Buttons: "Go" (blue) and "Clear" (red)

At the bottom of the page, there is a copyright notice: "© 2015 Desenvolvimento de Aplicações Web" and a design credit: "Designed by Aluno".

Utilizando a linguagem PHP,

construa o ficheiro `register.php` que faz o preenchimento das variáveis (“placeholders”) em `register_template.tpl` com os dados já introduzidos (nome, email) no caso de tentativa de registo falhada. Os dados

já introduzidos são enviados de volta pela “acção” `register_action.php` (mais detalhes no ponto 3 a seguir)

Em particular, deve colocar no *placeholder* `MESSAGE` uma mensagem informando qual o erro no registo (por exemplo email já existe).

### 3. Construa o ficheiro `register_action.php` que:

- (a) em caso de sucesso,
  1. vai actualizar os campos da base de dados no caso de sucesso (`name`, `email`, `created_at`, `updated_at`, `password_digest`). A `password_digest` é a hash utilizando o algoritmo MD5 da password: `substr(md5($_POST['pass1_utilizador']), 0, 32)`. Pode utilizar a função MySQL `NOW()` ou a função PHP `date("Y-m-d H:i:s")` para actualizar os campos `created_at`, `updated_at`.
  2. re-direcciona para a página `register_success.html`

```
header("Location: register_success.html");
```



Sign up completed!

© 2016 Desenvolvimento de  
Aplicações Web

Designed by Aluno

A página `register_success.html` contém uma meta-tag que re-direcciona automaticamente para `index.php` passados 3 segundos:

```
<meta http-equiv="refresh" content="3; url=index.php" />
```

- (b) em caso de erro,

re-direcciona novamente para a página de registo `register.php`. A tentativa de registo considera-se falhada se (a) o email já existe na base de dados (b) o email não é válido (c) a repetição da password não coincide. Comunique os dados válidos e o tipo de erro de

register\_action.php para register.php com um [link com dados embebidos](#)<sup>\*</sup>, por exemplo

```
header("Location:register.php?error=1&name=Joao%20Silva");
```

Considere os seguintes tipos de erro e sugerem-se as seguintes mensagens:

```
error=0 ("Todos os campos devem ser preenchidos")
```

```
error=1 ("Email já existe na base de dados")
```

```
error=2 ("Email tem formato incorrecto")
```

```
error=3 ("Password em branco")
```

```
error=4 ("Passwords não coincidem")
```

## 4. Faça o upload dos ficheiros PHP

- register.php
- register\_action.php
- register\_success.html
- model.php (opcional)

para a pasta "LAB5"

Faça o upload do template

- register\_template.tpl

para a pasta "LAB5/templates"

Considere o LAB5 concluído quando tiver reproduzido a funcionalidade<sup>1</sup> de

---

<sup>\*</sup> **Não utilize cookies**, estes serão utilizados preferencialmente nos próximos laboratórios.

[http://all.deei.fct.ualg.pt/~a999990/smarty\\_exame2/register.php](http://all.deei.fct.ualg.pt/~a999990/smarty_exame2/register.php)

no seu site web pessoal.

---

<sup>1</sup> NOTA: o site exemplo usa variáveis de sessão para passar a informação entre controladores. NÃO É SUPOSTO neste laboratório utilizar variáveis de sessão, mas sim passar a informação através de dados embutidos no URL!

## REFERÊNCIAS:

- <http://intranet.deei.fct.ualg.pt/DAW/forms/files.html>
- [http://intranet.deei.fct.ualg.pt/DAW/auth-db-sessions/signup\\_smartypage.html](http://intranet.deei.fct.ualg.pt/DAW/auth-db-sessions/signup_smartypage.html)
- [http://intranet.deei.fct.ualg.pt/DAW/auth-db-sessions/signup\\_action\\_smartypage.html](http://intranet.deei.fct.ualg.pt/DAW/auth-db-sessions/signup_action_smartypage.html)
- [http://intranet.deei.fct.ualg.pt/DAW/auth-db-sessions/signup\\_template.tpl](http://intranet.deei.fct.ualg.pt/DAW/auth-db-sessions/signup_template.tpl)

# ANEXO 1: Acesso à base de dados MySQL

- O acesso à base de dados MySQL 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 default 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.uaig.pt/IPM/XAMPP.pdf>



# ANEXO 4: OPCIONAL

## FUNCIONALIDADE “PASSWORD ENCRIPADA COM SEMENTE”

Pretende-se melhorar a funcionalidade de encriptar a password que é guardada na base de dados

O método atual de criação da password encriptada

```
$passe = substr(md5($_POST['pass1_utilizador']),0,32);
```

não é seguro porque, se dois ou mais utilizadores utilizarem a mesma password, exactamente o mesmo token/hash é guardado na base de dados. Se houver acesso à base de dados, isto é uma falha de segurança grave.

Pretende-se agora utilizar uma “semente/salt” diferente sempre que se encripta a password.

O algoritmo é o seguinte:

### To Store a Password

1. Generate the random salt.
2. Prepend the salt to the password and hash it with a **standard** password hashing function like md5.
3. Save both the salt and the hash in the user's database record.

### To Validate a Password

1. Retrieve the user's salt and hash from the database.
2. Prepend the salt to the given password and hash it using the same hash function.

Compare the hash of the given password with the hash from the database. If they match, the password is correct. Otherwise, the password is incorrect.

Uma forma muito simples de encriptar a password com semente é a seguinte

```
$seed=substr(time(),-4);  
$passe=substr($seed . md5($seed . $_POST['pass1_utilizador']),0,32);
```

Actualize o programa **register\_action.php**<sup>2</sup> para passar a utilizar o método de encriptar a password acima descrito

---

<sup>2</sup> No LAB6 terá de atualizar o controlador **login\_action.php** para utilizar o mesmo algoritmo para validar a password introduzida pelo utilizador!