

**LER O ENUNCIADO ATÉ AO FIM ANTES DE COMEÇAR!**

---

O objectivo do exame é adicionar uma nova funcionalidade à aplicação web que foi construída ao longo dos laboratórios da disciplina.

É dado acesso a uma implementação do site web com as funcionalidades correspondentes ao LAB 10 (realizado em *CodeIgniter* e *Smarty*). A base de dados já se encontra implementada (informação sobre a base de dados encontra-se em ANEXO).

Apenas se solicita ao aluno a alteração dos templates e o código controlador PHP necessário para implementar a nova funcionalidade.

**PRELIMINARES**

A. Faça login por ssh (com o PuTTY, por exemplo) no servidor com o IP 10.10.23.183

```
a12345@daw: ~$
```

B. Faça download para a pasta “recurso” já existente, do código do site web arquivado em “Github” com o comando “git”, e ponha as permissões correctas.

```
a12345@daw: ~$  
git clone https://github.com/jmatbastos/recurso29.01.2018.git  
~/public_html/recurso
```

(nota: o comando “git” é todo só numa linha!)

```
a12345@daw: ~$ cd public_html  
a12345@daw: ~/public_html$ find recurso -type d -exec chmod g+rx {} \;  
a12345@daw: ~/public_html$ find recurso -type f -exec chmod g+r {} \;  
  
a12345@daw: ~/public_html$ cd recurso/application/views  
a12345@daw: ~/public_html/recurso/application/views$ chmod -R g+w templates_c
```

C. Actualize a base URL in config.php

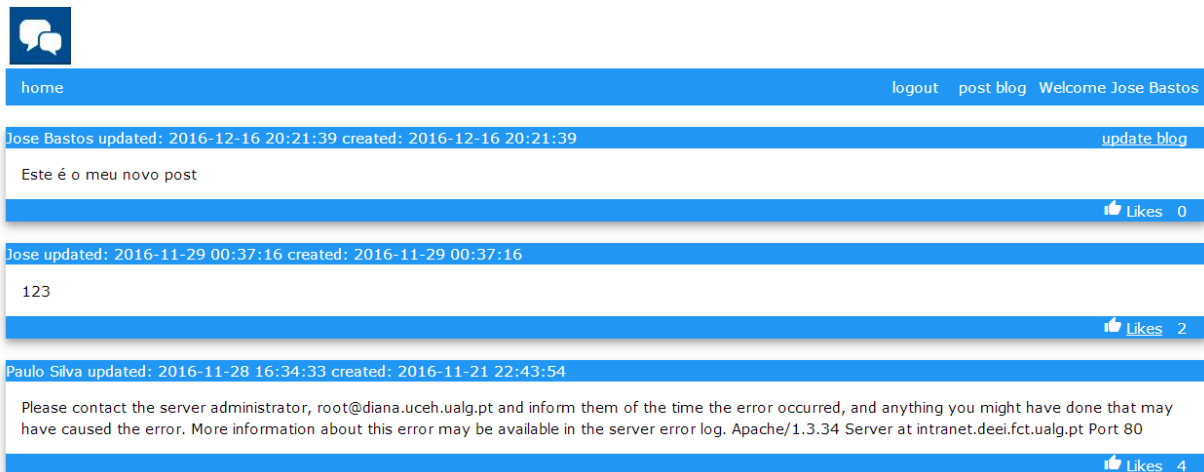
```
a12345@daw: ~/public_html/recurso/application/config$ nano config.php  
  
$config['base_url'] = '~/a12345/recurso';
```

(nota: substitua “12345” pelo seu número de aluno!)

## PARTE A - FUNCIONALIDADE “LIKES” [7 valores]

Pretende-se nesta secção implementar a funcionalidade “likes” na página de rosto do site

1. Actualize o template `index_template.tpl` para ter um placeholder para um link “Likes”



- caso o utilizador esteja registado
  - activar o hyperlink “Likes”  
`<a href="{ $like_url } { $blog.id } ">Likes</a>`

para todos os posts excepto os seus (são proibido auto-likes)

NOTA: o placeholder `{ $like_url }` é definido no controlador `Blog.php` função `index()` como

```
$data['like_url'] = site_url('blog/likes/')
```

- caso o utilizador seja anónimo o hyperlink não existe

2. Actualize o template `index_template.tpl` para ter um placeholder que mostra o numero de “likes” de cada micropost (tabela microposts, coluna likes)

3. Construa no controlador `Blog.php` a função `likes($blog_id = FALSE)` que recebe o “id” do micropost e

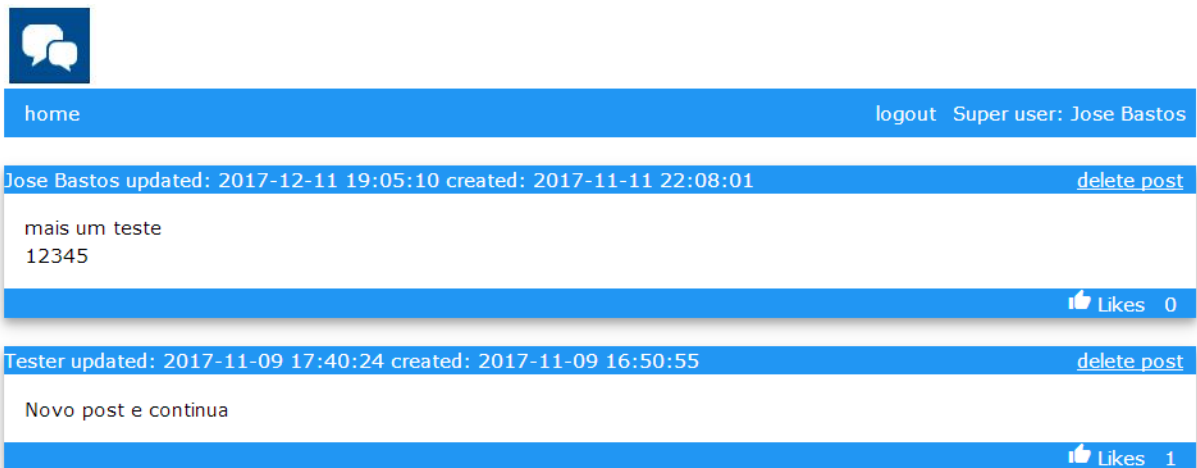
- verifica se a sessão tem um utilizador registado

- verifica na base de dados se NÃO é um auto-like (chamando a função `function get_blog($blog_id)` já definida no controlador `Blog_model.php`)
- incrementa de “1” o valor de “likes” do micropost (tabela `microposts`, coluna `likes`) chamando a função `update_likes($blog_id)` no controlador `Blog_model.php`

4. Construa no controlador `Blog_model.php` a função `update_likes($blog_id)`

## PARTE B - FUNCIONALIDADE “ADMIN” [6 valores]

Pretende-se nesta secção implementar a funcionalidade “admin” na página de rosto do site: os utilizadores com o campo “admin=1” na tabela “users” são considerados super utilizadores.



5. Altere a função `login_action()` para adicionar mais uma variável de sessão `$_SESSION['admin']`

Actualize o template `index_template.tpl` para que, para super utilizadores, todos os posts tenham um hyperlink que permita apagar selectivamente o post

```
<a href="{ $delete_url } { $blog.id } ">delete post</a>
```

NOTA: o placeholder `{ $delete_url }` é definido no controlador `Blog.php` função `index()` como

```
$data['delete_url'] = site_url('blog/delete/');
```

6. Construa no controlador `Blog.php` a função `delete($blog_id = FALSE)`

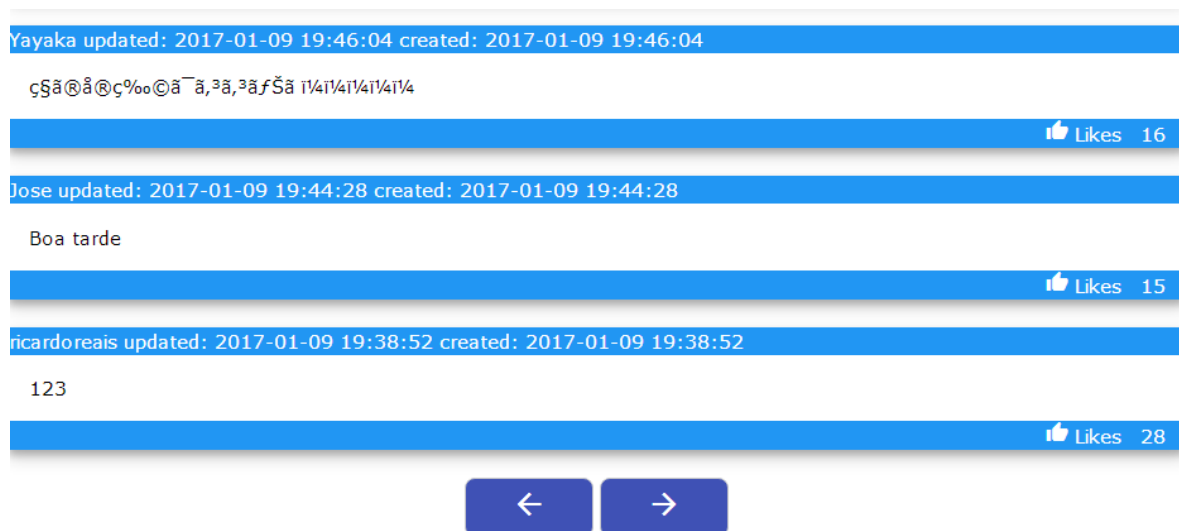
Esta função, exclusivamente para super utilizadores, chama a função `delete_blog($blog_id)` a definir no controlador `Blog_model.php`

7. Construa no controlador `Blog_model.php` a função `delete_blog($blog_id = FALSE)`

Esta função remove o post da tabela “microposts”.

### PARTE C - FUNCIONALIDADE “PAGINATION” [7 valores]

Pretende-se nesta secção implementar a funcionalidade “paginação” na página de rosto do site:



© 2016 Desenvolvimento de Aplicações Web

Designed by [Aluno](#)

Esta funcionalidade separa os posts em grupos de dez, permitindo mostrar apenas dez posts de cada vez

8. Actualize o template `index_template.tpl` para ter dois placeholders para dois hyperlinks no fim da pagina. Os hyperlinks podem ser apenas texto (por exemplo “next page” “previous page”), ou dois icons como se mostra na figura.

9. Atualize no controlador `Blog.php` a função `index($page = FALSE)`

de forma a atualizar os hyperlinks

Por exemplo se a página de rosto fôr a pagina três:

“previous page”:

`http://all.deei.fct.ualg.pt/~a12345/recurso/index.php/blog/index/2`

“next page”:

`http://all.deei.fct.ualg.pt/~a12345/recurso/index.php/blog/index/4`

- Caso a página de rosto seja a primeira página, apenas o hyperlink “next page” deve ser visível
- Caso a página de rosto seja a ultima página, apenas o hyperlink “previous page” deve ser visível

10. Atualize no controlador `Blog_model.php` a função `get_posts($page = FALSE)`,

de forma a apenas retornar dez posts de cada vez.

#### NOTAS:

- Coloque o seu código entre as linhas “PRINCIPIO DO CODIGO” e “FIM DO CODIGO” (o seu código pode ter qualquer numero de linhas; adicione as linhas que precisar)
- Caso considere absolutamente necessário escrever código noutra local, comente o seu código e justifique-o num ficheiro “README.TXT”. Pode utilizar esse ficheiro também para outra informação que julgue pertinente.
- Caso tenha trabalhado no seu portatil, **é obrigatório fazer o upload de todos os ficheiros** para a pasta “recurso” no seu site web pessoal

`http://all.deei.fct.ualg.pt/~a12345/recurso`

(onde 12345 é o seu número de aluno). Utilize `scp` (Linux) ou `WinSCP` (Windows) para fazer a cópia. **Verifique que o site fica operacional** (com as permissões necessárias).

- Caso tenha problemas como seu código e o browser apresente uma página web em branco, lembre-se que pode consultar o log do servidor web com o comando

```
a12345@daw:~$ tail -f /var/log/apache2/error.log
```

## ANEXO 1 Acesso à base de dados MySQL

- O acesso à base de dados MySQL 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 a999999 -p -h 10.10.23.183 db_a999999
```

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

<http://all.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 (`user_id`),  
  CONSTRAINT FOREIGN KEY (`user_id`) REFERENCES `users` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```