

LER O ENUNCIADO ATÉ AO FIM ANTES DE COMEÇAR!

O objectivo do exame é adicionar novas funcionalidades 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).

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 “exame” do código do site web arquivado em “Github” com o comando “git” e ponha as permissões de escrita na pasta `template_c`:

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

```
git clone https://github.com/jmatbastos/EXAME.git ~/public_html/EXAME
```

e ponha as permissões de escrita na pasta `templates_c`:

```
chmod -R g+w ~/public_html/EXAME/application/views/templates_c
```

C. Actualize a base URL in `config.php`

```
a12345@daw:~/public_html/EXAME/application/config$ nano config.php
```

```
$config['base_url'] = '~/a12345/EXAME';
```

NOTAS:

- **substitua “12345” pelo seu número de aluno!**
- O seu site deve ser portavel; utilize sempre as funções `site_url()` e `base_url()` para gerar hyperlinks!

D. Verifique que o site é visível no URL

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

FUNCIONALIDADE “FRIENDS”

Pretende-se construir uma página no site que permite a utilizadores registados criar uma lista de amigos (“friends”)

1. [7 valores]

Actualize o template `index_template.tpl` e o controlador `index()` em `Blog.php`

- se o utilizador fez *log in*, para ter uma variável adicional `{ $menu4 }` activa no menu para o *hyperlink*

```
<a href="/~a12345/EXAME/index.php/blog/friends">friends</a>
```



home

friends logout post blog Welcome Jose Bastos

Construa o template `friends_template.tpl`, como se mostra a seguir:



home

logout Welcome Jose Bastos

Friends

Friends (click to unfriend!)	Users (click to make friend!)
Maria Silva(maria.silva@gmail.com)	Jose Bastos(jose.bastos@gmail.com)
Paulo Silva(a22297@ualg.pt)	Maria Silva(maria.silva@gmail.com)
Pedro Graça(kabasizubo@hostcalls.com)	Paulo Silva(a22297@ualg.pt)
	Joao Andrade(a52495@ualg.pt)
	Dário Hermann(a49728@ualg.pt)
	Banana Bananas(bananas@hotmail.com)
	Pedro Graça(kabasizubo@hostcalls.com)
	Minion (minionsforever@hotmail.com)
	Dank Blogger(dank@blogger.ru)

Este template é constituído por duas listas:

- A lista da direita é uma lista com os utilizadores registados no site. Cada elemento da lista é um hyperlink

```
<a href="/~a12345/EXAME/index.php/blog/makefriend/xx">Paulo Silva(a22297@ualg.pt)</a>
```

que permite seleccionar esse utilizador como amigo

- A lista da esquerda é uma lista com os amigos do utilizador. Cada elemento da lista é um hyperlink

```
<a href="/~a12345/EXAME/index.php/blog/unfriend/xx">Paulo Silva(a22297@ualg.pt)</a>
```

que permite remover esse utilizador da lista de amigos

SUGESTÃO: adapte o template `friends_template.html` que vai encontrar dentro da pasta `views/templates`

Construa o controlador `friends()` em `Blog.php` e as queries à base de dados correspondentes em `Blog_model.php`

**NOTA: coloque o seu código no final dos ficheiros a seguir ao comentário
//PONHA O SEU CODIGO ABAIXO DESTA LINHA**

SUGESTÃO: registe-se no site e em seguida coloque manualmente alguns “amigos” na tabela “friends” associados ao seu `user_id`, para aparecerem alguns nomes na sua lista de amigos

2. [3 valores]

- Construa o controlador `makefriend($id)` em `Blog.php` e a query à base de dados correspondente em `Blog_model.php`

que insere um novo amigo na tabela “friends”. O controlador e a query devem garantir que só é possível inserir amigos ao utilizador registado na sessão

- Construa o controlador `unfriend($id)` em `Blog.php` e a query à base de dados correspondente em `Blog_model.php`

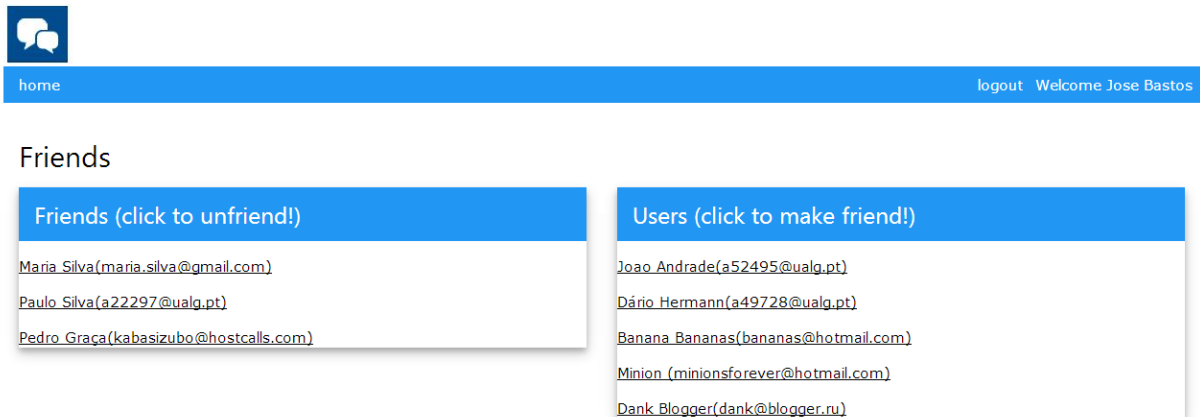
que remove um amigo na tabela “friends”. O controlador e a query devem garantir que só é possível remover amigos do utilizador registado na sessão.

3. [3 valores]

Na pergunta 1 os utilizadores seleccionados como amigos continuam a aparecer na tabela da direita.

Altere o controlador `friends()` em `Blog.php` e a query à base de dados correspondente em `Blog_model.php`

para que os utilizadores seleccionados como amigos, bem como o utilizador registado na sessão, “desapareçam” da lista da direita:



The screenshot shows a web application interface. At the top, there is a blue navigation bar with a home icon on the left and 'logout Welcome Jose Bastos' on the right. Below the navigation bar, the main content area is titled 'Friends'. It contains two side-by-side panels. The left panel, titled 'Friends (click to unfriend!)', lists three users: Maria Silva (maria.silva@gmail.com), Paulo Silva (a22297@ualg.pt), and Pedro Graça (kabasizubo@hostcalls.com). The right panel, titled 'Users (click to make friend!)', lists five users: Joao Andrade (a52495@ualg.pt), Dário Hermann (a49728@ualg.pt), Banana Bananas (bananas@hotmail.com), Minion (minionsforever@hotmail.com), and Dank Blogger (dank@blogger.ru).

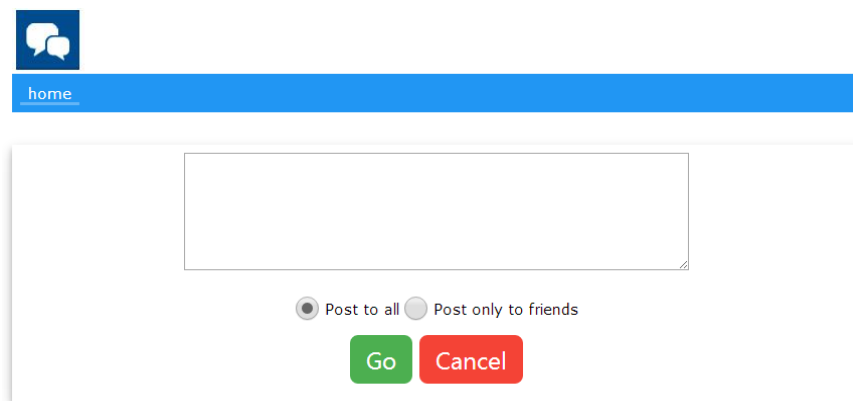
SUGESTÃO: faça uso dos seus conhecimentos de bases de dados e faça a filtragem da lista “Users” na query.

FUNCIONALIDADE “RESTRICTED POSTS”

Pretende-se permitir a utilizadores registados criar posts que só são visíveis aos utilizadores na lista de amigos

4. [2 valores]

Altere o template “`blog_template.tpl`” para ter um “radio button” que permite definir se o post é publico, ou se é restrito ao grupo de amigos



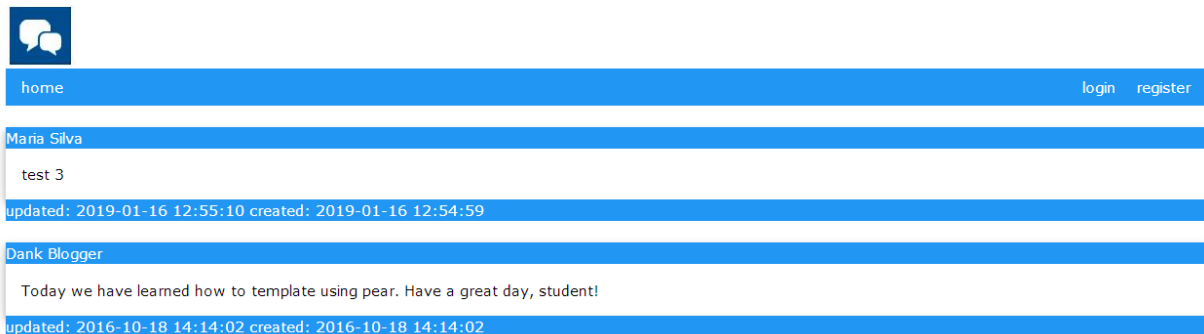
Actualize as queries `new_blog` e `update_blog` no modelo `Blog_model.php` que actualizam a tabela “microposts”

NOTA: o atributo “public” deve ser “1” para posts publicos e “0” para posts de distribuição restrita a amigos

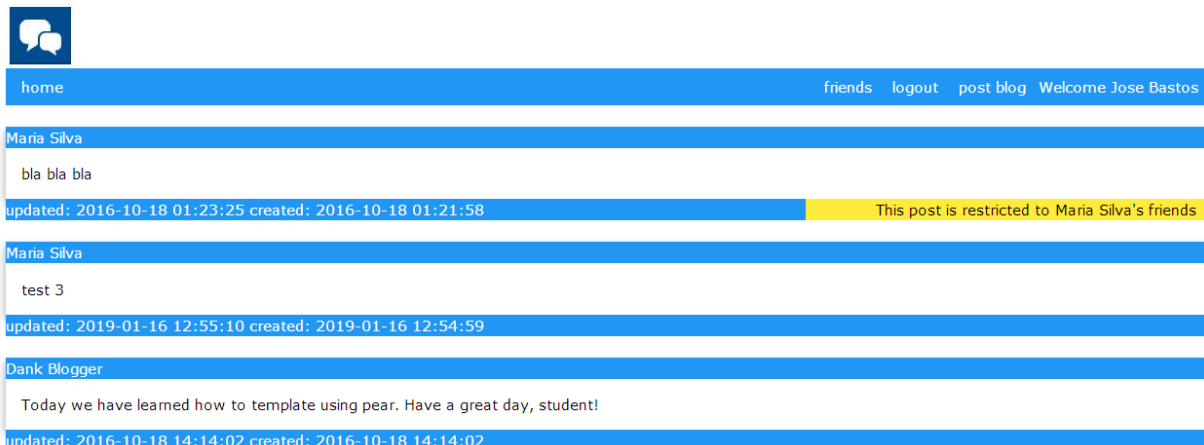
5. [5 valores]

Altere o template `index_template.tpl` o controlador `index()` em `Blog.php` e construa as queries necessárias em `Blog_model.php` para

- se o utilizador é anónimo apenas são visíveis os posts publicos



- se o utilizador fez *log in*, para além dos posts publicos, o utilizador vê no topo da lista tambem os posts de distribuição restrita de utilizadores que o seleccionaram como amigo*



Neste screenshot, para alem do post publico “test 3” da Maria Silva, o utilizador Jose Bastos tambem pode ver o post restrito “bla bla bla” da Maria **porque** ela o seleccionou como amigo.

Os posts restritos têm também um estilo que os identifica, neste caso a cor amarela no fundo do texto “This post is restricted...”

* os posts de distribuição restrita gerados pelo proprio utilizador não são visiveis na lista, a menos que o utilizador se faça amigo de si próprio...

NOTAS:

- Caso tenha trabalhado no seu portatil, **é obrigatório fazer o upload de todos os ficheiros** para a pasta “EXAME” em

```
/users/a12345/public_html/EXAME
```

(onde 12345 é o seu número de aluno). Utilize scp (Linux) ou WinSCP/FileZilla (Windows) para fazer a cópia. **Verifique que o site fica operacional.**

- Não faça o upload de pastas. No upload de ficheiros assegure-se que as permissões por defeito (`rw- r-- --- octal 640`) dos ficheiros estão correctas
- Caso tenha problemas como seu código, 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 a999998 -p -h 10.10.23.183 db_a999998
```

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 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',  
  `public` int(1) NOT NULL DEFAULT '1',  
  PRIMARY KEY (`id`),  
  KEY (`user_id`),  
  CONSTRAINT FOREIGN KEY (`user_id`) REFERENCES `users` (`id`)  
) ENGINE=InnoDB CHARSET=utf8;  
  
CREATE TABLE `friends` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `user_id` int(11) NOT NULL,  
  `friend_user_id` int(11) NOT NULL,  
  `created_at` datetime NOT NULL,  
  `status` int(1) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY (`user_id`),  
  KEY (`friend_user_id`),  
  UNIQUE KEY (`user_id`,`friend_user_id`),  
  CONSTRAINT FOREIGN KEY (`user_id`) REFERENCES `users` (`id`),  
  CONSTRAINT FOREIGN KEY (`friend_user_id`) REFERENCES `users` (`id`)  
) ENGINE=InnoDB CHARSET=utf8;  
  
CREATE VIEW `microposts_total` AS select  
  `microposts`.`user_id` AS `user_id`,  
  count(0) AS `total` from `microposts`  
  group by `microposts`.`user_id` order by count(0);
```