

## Módulo 3 - PHP+MySQL | Exercício 3b – área administração

### 1ª parte do sistema administração

#### Temas abordados

- Criação de um sistema de administração

Versão resolvida:

[http://labmm.clients.ua.pt/deca\\_17L4/deca\\_17L4\\_p/PHP\\_MySQL\\_ex\\_03b\\_resolvido/admin](http://labmm.clients.ua.pt/deca_17L4/deca_17L4_p/PHP_MySQL_ex_03b_resolvido/admin)

Login: admin

Pass: demo

**Neste exercício pretende-se iniciar o desenvolvimento da área de administração. Caso ainda não possua uma versão avançada de HTML+CSS para o projeto final, sugere-se que utilize os ficheiros fornecidos para este exemplo e, posteriormente, faça a sua adaptação.**

Esta proposta de website está dividida em duas componentes distintas.

- Na pasta “public” que trabalhou na aula prática anterior;
- Na pasta “admin” pode encontrar os ficheiros utilizados para construir a área administração do website. Esta área só de encontra disponível aos utilizadores autenticados e com perfil de administrador. Para este exercício foi utilizada uma versão simplificada do seguinte template de bootstrap.

**Template: SB Admin 2** <http://startbootstrap.com/template-overviews/sb-admin-2>

Recomenda-se que explore as funcionalidades adicionais disponibilizadas no template.

Os ficheiros disponibilizados neste exercício devem ser copiados para a pasta do projeto no PhpStorm criada na última aula.

#### Siga os seguintes passos:

#### PARTE 1 – LIGAÇÃO À BD

1. Estude a estrutura geral da pasta “admin”. Nesta parte do exercício só irá trabalhar nessa pasta e todas as referências deste enunciado são para pastas dentro da pasta “admin”;
2. Estude a estrutura da página “index.php”;
3. Atualize os dados de ligação à BD no ficheiro “connections/connection.php”;
4. Experimente invocar a função de ligação à base de dados, e verifique se a ligação é efetuada com sucesso.

## PARTE 2 – Condicionar acesso à área de administração

Nesta parte pretende-se implementar um mecanismo para condicionar o acesso dos utilizadores à área de administração, garantindo que só utilizadores com as devidas credenciais conseguem aceder. É importante ter em atenção que esconder o link para a área de administração não é proteção suficiente!

1. No componente “check\_admin.php”, adicione o código necessário para validar se existe um utilizador autenticado com privilégios de administração. Caso não exista, redirecione automaticamente o utilizador para a área de login na parte pública;
2. Todas as páginas da área da administração (pasta “pages”) devem incluir as barras de navegação que se encontram centralizadas no componente “navbars.php”;
3. Nesse componente das barras de navegação, adicione o componente “check\_admin.php”. Desse modo, garanta que todas as páginas da administração que tenham as barras de navegação estão protegidas de acessos indevidos;
4. No entanto, outras páginas que venha a desenvolver onde não seja necessário incluir as barras de navegação também devem ser protegidas de acessos indevidos.

## PARTE 3 – LOGOUT

Nesta parte pretende-se implementar as mudanças necessárias na barra de topo depois de ser efetuado um login com sucesso.

5. Na página “logout.php” insira o código necessário para terminar a sessão do utilizador. Garanta que todas as variáveis de sessão são destruídas<sup>1</sup>;
6. Redirecione o utilizador para a área pública;
7. Verifique o funcionamento efetuando o login/logout no website.

## PARTE 4 – LISTAGEM DE UTILIZADORES

Nesta parte pretende-se implementar a área de gestão de utilizadores.

8. Na página “users.php” adicione o código necessário para listar todos os utilizadores da BD, incluindo os seguintes dados: id, username, data de criação da conta, email e perfil de utilizador;
9. Na base de dados adicione um campo “active”, do tipo booleano, na tabela “users”. Este campo, por defeito, deve ter o valor verdadeiro e indica se um utilizador registado foi ou não desativado;
10. Altere a lista de utilizadores na página “users.php” passando a estar identificados os utilizadores desativados. Os ícones necessários podem ser aqui consultados: <http://fontawesome.io/icons/>
11. Altere a página de login para garantir que só é permitido fazer login aos utilizadores que se encontram ativos.

---

<sup>1</sup> <http://php.net/manual/en/function.session-destroy.php>

## **PARTE 5 – EDIÇÃO DE UTILIZADORES**

**Nesta parte pretende-se implementar a página de edição dos dados de um utilizador.**

12. Para cada utilizador da lista deve existir a opção de edição através do respetivo ícone.  
Garanta que esse tem a ligação para a página “user\_edit.php” e o valor da chave primária do utilizador é passado através da query string;
13. Programe a página “user\_edit.php” para permitir editar os dados do utilizador selecionado, com exceção da password;
14. O formulário deve apresentar os dados anteriores do perfil do utilizador e sugere-se que execute este passo em duas versões:
  - a. Uma versão mais simples com a edição do perfil do utilizador através de um campo de texto;
  - b. Uma versão mais completa com a edição do perfil do utilizador através de um `select` do HTML;
15. A ação de submeter o formulário deve direcionar para o sript “users\_update.php”;
16. Na página “users\_update.php”, escreva o código necessário para atualizar os dados do utilizador na base de dados;
17. Alguns detalhes de aperfeiçoamento que pode incluir na sua solução:
  - a. Garantir que o utilizador que está na área de administração não pode desativar e a sua própria conta nem pode retirar os privilégios de administração.