



LICENCIATURA ENGENHARIA INFORMÁTICA E MULTIMÉDIA

SISTEMAS MULTIMÉDIA PARA A INTERNET

Trabalho Laboratorial - 3ª Parte

DATA: 5 DE JULHO DE 2020

Realizado por:

Docente:

Eng. Carlos Gonçalves

Rafael Vitorino - 43409

Miguel Duarte - 43756

Luís Fonseca - 45125

Grupo:13

Índice de Conteúdos

1	Objetivo	6
2	SGC - Sistema de Gestão de Conteúdos	6
	2.1 O que é um SGC?	6
	2.2 Vantagens de usar um SGC	6
	2.3 Desvantagens de um SGC	7
	2.4 SGC Escolhido	7
3	Plataforma E-Learning	7
4	Funcionalidades	7
	4.1 Funcionalidades dos Utilizadores do Sistema	7
	4.2 Simpatizante	8
	4.3 Administrador	8
	4.4 Funcionalidades da plataforma	8
	4.5 Curso	9
	4.6 Registo	9
5	Conteúdos Multimédia	9
6	Modelo EA	10
	6.1 Utilizador	10
	6.2 Curso	11
	6.3 Conteúdo	11
7	Base de Dados	11
	7.1 Registos	11
	7.2 Validação de Conta	11
	7.3 Atribuição de Roles	11
	7.4 Roles existentes	12
	7.5 Conteúdo	12
	7.6 Curso	12
	7.7 curso_conteudo	13
	7.8 email_accounts	13
	7.9 Imagem	14
	7.10 Video	14
	7.11 Inscrição	15

8	Produto Final	15
8.1	Página Principal	15
8.2	About	18
8.3	Página Geral com os Cursos	20
8.4	Login	45
8.5	Registo	59
8.6	Conteúdos	71

Índice de Figuras

1	Modelo EA	10
2	Tabela Base de Dados 'conteudo'	11
3	Tabela Base de Dados 'auth-challenge'	11
4	Tabela Base de Dados 'auth-permissions'	12
5	Tabela Base de Dados 'auth-roles'	12
6	Tabela Base de Dados 'conteudo'	12
7	Tabela Base de Dados 'curso'	13
8	Tabela Base de Dados 'curso'	13
9	Tabela Base de Dados 'curso'	14
10	Tabela Base de Dados 'images'	14
11	Tabela Base de Dados 'video'	14
12	Tabela Base de Dados 'inscricao'	15
13	Página Principal	15
14	About acerca dos autores do projeto	18
15	About com o google maps e as respetivas redes sociais	18
16	Design da página dos Cursos em geral, com perspetiva no utilizador .	21
17	Design da página dos Cursos em geral, com perspetiva do administrador	22
18	Janela para criar um curso, na página geral dos cursos	23
19	Prompt para apagar um curso	24
20	Página com alteração role do utilizador	25
21	Janela criada para alterar a visibilidade dos cursos	26
22	Página para alterar o acesso à base de dados do projeto	26
23	Design da página de login	45
24	Design da página de login após incorreta introdução de dados	46
25	Design da página de submeter o seu email	46
26	Email a confirmar a mudança da nova palavra pass	46
27	Design da página de submeter a nova pass	47
28	Design da página de login	59
29	Design da página de login após incorreta introdução de dados	59
30	Design da página de Conteúdo	71
31	Adicionar novo Conteúdo, Imagem, Video	72

1 Objetivo

Pretende-se um Sistema de Gestão de Conteúdos–SGC (Content Management System – CMS) que de forma simples e robusta implemente uma plataforma que permita a “partilha de conteúdos multimédia”, tais como fotografias, vídeos ou sons. O sistema a desenvolver será acedido única e exclusivamente utilizando um browser (quer por parte dos convidados, utilizadores, simpatizantes ou administradores).

2 SGC - Sistema de Gestão de Conteúdos

2.1 O que é um SGC?

Um SGC é um sistema que permite a gestão de sítios web, seja na forma de um sítio que apresenta uma organização, suporta uma rede social, um blog ou um wiki. Um SGC é um esqueleto de um sítio, que através de algumas configurações fica pronto a ser utilizado. Pretende-se um Sistema de Gestão de que de forma simples e robusta implemente um sítio que permita a “partilha de conteúdos multimédia”, tais como fotografias, vídeos ou sons. O sistema a desenvolver será acedido única e exclusivamente utilizando um browser.

2.2 Vantagens de usar um SGC

Minimizar o gasto da empresa em desenvolvimento/programação de recursos para gestão de conteúdos, formulários, base de dados,etc;

Descentralizar a responsabilidade sobre a atualização de conteúdos em um website, permitindo que pessoas sem conhecimento técnico especializado em programação na internet atualizem as páginas do website.

Acompanhar estatisticamente o acesso aos conteúdos, permitindo obter informações sobre o público-alvo de um website, quais os seus interesses e as suas necessidades; permitir a atuação de forma preditiva ao apresentar conteúdos de interesse do cliente durante a sua experiência de navegação.

O utilizador não precisa de prever características básicas para o site, como login de utilizadores, criação, edição e publicação de conteúdo, publicação de “banners” de propaganda, pois esses recursos já estão pré-programados e prontos para serem utilizados.

Funcionalidades extras, tais como mecanismos de pesquisa, calendários, webmail, e outros componentes podem ser construídos no próprio ambiente, ou disponibilizados por terceiros como plug-ins.

Economizar tempo que seria, normalmente gasto no desenvolvimento ou na tentativa de integrar componentes.

2.3 Desvantagens de um SGC

Como em todos os framework, trabalhar com um “modelo” pré-concebido implica aceitar algumas características, restrições ou limitações desse “modelo”.

A aparência do site, apesar de poder ser alterada, também fica de certa forma limitada pelo sistema de templates do CMS.

2.4 SGC Escolhido

O SGC que foi escolhido como base foi o ATutor (referenciado na bibliografia). Esse SGC tem como funcionalidade ser um sistema de e-learning online e permite criar e gerir cursos online. Este SGC tem diversas funções tais como uma galeria de fotos, criação de *external tools*, login e registo de utilizadores, visualizar e pesquisar cursos, visualizar os módulos existentes entre outros.

3 Plataforma E-Learning

O tema do CMS a implementar será sobre um website E-Learning que tem como finalidade ensinar diversos cursos numa plataforma online simples e robusta.

4 Funcionalidades

4.1 Funcionalidades dos Utilizadores do Sistema

Convidado

Este perfil permite apenas visualizar e pesquisar os cursos disponíveis no catálogo de cursos.

Utilizador

Este perfil permite após ter a sua conta criada efetuar login e além de visualizar e pesquisar cursos pode ainda se inscrever nos cursos que desejar. O Utilizador pode ainda dar a sua opinião acerca de um curso através de um comentário ou dando uma atribuição numérica (1 a 5).

4.2 Simpatizante

Este perfil tem várias funcionalidades, sendo elas, criação de cursos, obter conteúdos do curso, enviar conteúdos multimédia para o curso e alterar a visibilidade de cada conteúdo (público ou privado).

4.3 Administrador

Este perfil engloba todas as funcionalidades anteriores e ainda permite editar, modificar e apagar cursos. Além disso pode ainda configurar o serviço de correio electrónico permitindo enviar emails com novidades ou informações importantes a cada utilizador registado na plataforma. O administrador pode também criar novas categorias de cursos no catálogo de cursos e pode ainda gerir os utilizadores registados.

4.4 Funcionalidades da plataforma

Registo e Login

Esta funcionalidade do sistema serve para os convidados poderem-se registar para pertencerem à plataforma do sistema.

Criação de Cursos

Permitirá o administrador e o simpatizante criar novos cursos e disponibilizar novos conteúdos para os utilizadores pertencentes à plataforma poderem usufruir.

Inscrição em Cursos

Permitir aos utilizadores inscreverem-se aos cursos que desejarem.

Modificar ou Apagar Cursos

Nesta secção o administrador é o único que pode fazer ambas as funcionalidades de modificar e apagar cursos, tem livre acesso a ambos. O simpatizante apenas pode modificar os curso, alterando categorias secundárias, etc.

Notificações de novos Cursos

Esta funcionalidade permite que utilizadores recebam notificações por email de novos cursos recentemente criados numa categoria que os mesmos subscrevam.

Avaliação e Comentários de cada Cursos

Esta funcionalidade permite fazer uma avaliação (usando uma classificação de 1 a 5 estrelas) e, se achar necessário, colocar um comentário, positivo ou negativo, acerca da sua opinião, fazer uma abordagem do curso que realizou ou apontar os defeitos e o que tem de ser melhorado. Esta funcionalidade é apenas usada pelos utilizadores, visto que, é aquele que pode efetuar o login na plataforma E-Learning.

4.5 Curso

Num curso serão partilhados diversos conteúdos multimédia para os utilizadores poderem aprender. O simpaticante pode editar o curso como foi referido anteriormente e o utilizador poderá fazer download dos conteúdos multimédia disponibilizados no curso para que possa assisti-los sem estar na plataforma caso assim o deseje.

4.6 Registo

No processo de registo serão pedidas diversas informações ao utilizador que serão posteriormente armazenadas na base de dados. No caso do utilizador querer se registar com redes sociais como a Google o mesmo também será permitido. No final do registo será também feito a validação "Não sou um robot". Para o email será enviado uma confirmação de registo após ser concluído todo o registo.

5 Conteúdos Multimédia

Como o nosso website, irá incidir em criar uma plataforma de e-learning agrupado em cursos, iremos colocar conteúdos multimédia dentro do nosso website, sendo esses conteúdos, vídeos e imagens.

Os vídeos iram funcionar como aulas, onde os diferentes utilizadores irão aprender como funcionam as diferentes linguagens. Apenas os utilizadores que apresentem um registo no nosso website, é que irão usufruir dessas aulas, e ter acesso a esses respectivos vídeos.

As imagens irão funcionar como sendo uma opção de dar upload de documentos, sendo que o conteúdo desses documentos, poderá conter exercícios, e páginas de livros ou documentos acerca do conteúdo leccionado, dando ao utilizador mais conhecimentos para além do que as aulas por vídeo possam fornecer.

6 Modelo EA

Durante a fase de planeamento, foi elaborado o seguinte modelo EA.

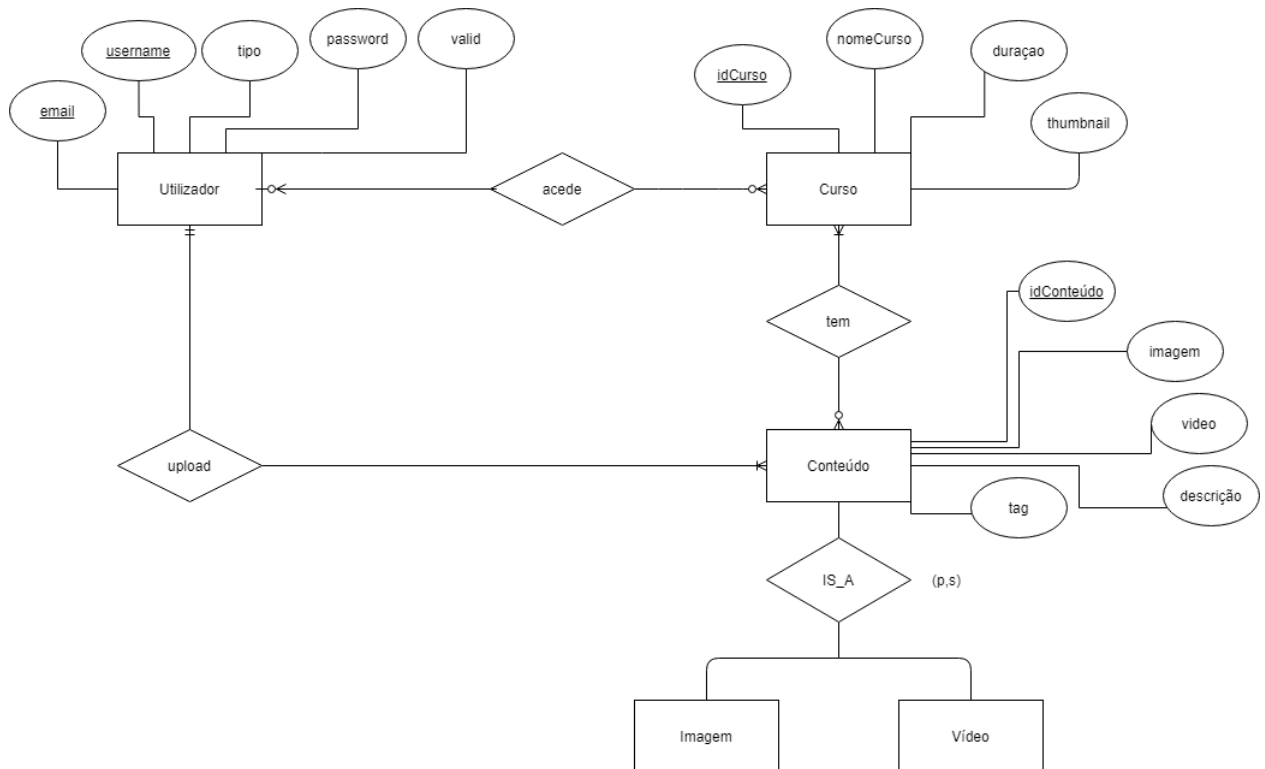


Figura 1: Modelo EA

Como se pode observar, é possível identificar 5 entidades que serão agora explicadas individualmente.

6.1 Utilizador

O utilizador é quem utiliza o website. É identificado por um email, um username, um tipo (role que tem no website), password e valid(identifica se a sua conta já foi ou não validada por email). Para o utilizador obter roles superiores é necessário o administrador atribuir os mesmos.

6.2 Curso

Um curso é identificado por um id, tem um nome, uma duração e uma thumbnail que descreve de forma gráfica o mesmo. Cada curso contém diversos conteúdos que poderão ser imagens ou vídeos.

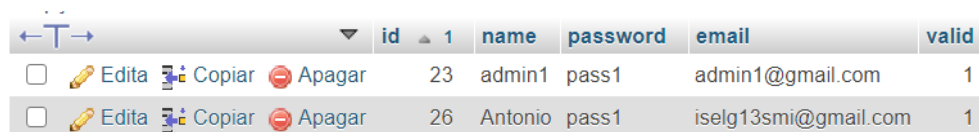
6.3 Conteúdo

Um conteúdo é então identificado por um id, uma imagem, um vídeo, uma descrição e uma tag que irá permitir a pesquisa pelo mesmo de forma rápida.

7 Base de Dados

7.1 Registos

A tabela 'auth-basic' contém todos os registos de contas existentes na plataforma. Nas colunas contém um id (identifica a conta), um nome (username), uma password e um email. Por fim, contém também um campo valid que distingue as contas já validadas e as não validadas.

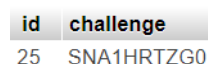


	id	name	password	email	valid
<input type="checkbox"/> Editar Copiar Apagar	23	admin1	pass1	admin1@gmail.com	1
<input type="checkbox"/> Editar Copiar Apagar	26	Antonio	pass1	iselig13smi@gmail.com	1

Figura 2: Tabela Base de Dados 'conteudo'

7.2 Validação de Conta

A tabela 'auth-challenge' mantém registo das contas que ainda estão por confirmar. Para tal, mantém um par de colunas (id-challenge) em que id é o id do user que está por confirmar e o challenge é o código que irá identificar esse pedido de validação.



id	challenge
25	SNA1HRTZG0

Figura 3: Tabela Base de Dados 'auth-challenge'

7.3 Atribuição de Roles

Na tabela 'auth-permissions' estão guardados todos os roles que cada conta tem. Para tal, é mantido um duplo (id-role) em que a cada id se associa um role.

role	id
1	23
3	26

Figura 4: Tabela Base de Dados 'auth-permissions'

7.4 Roles existentes

Na tabela 'auth-roles' estão identificados todos os papéis existentes no website juntamente com uma descrição textual dos mesmos.

		role	friendlyName
<input type="checkbox"/>	 Edita  Copiar  Apagar	1	admin
<input type="checkbox"/>	 Edita  Copiar  Apagar	2	simpatizante
<input type="checkbox"/>	 Edita  Copiar  Apagar	3	utilizador
<input type="checkbox"/>	 Edita  Copiar  Apagar	4	convidado

Figura 5: Tabela Base de Dados 'auth-roles'

7.5 Conteúdo

A tabela 'conteudo' tem 3 colunas, uma para o 'id_conteudo', uma para o título e outra para a descrição. Na coluna do 'id_conteudo' fica guardado um id para cada conteúdo, sendo que cada um desses ids é único. O 'titulo' vai guardar uma string referente ao titulo desse conteúdo e a 'descricao' guarda também uma string que contém um texto referente ao titulo do conteúdo.

+ Opções			
		id_conteudo	titulo
<input type="checkbox"/>	 Edita  Copiar  Apagar	1	Variáveis em Python Uma variável é um rótulo ou um nome dado a um dete...

Figura 6: Tabela Base de Dados 'conteudo'

7.6 Curso

A tabela 'curso' contém 8 tabelas, essas colunas são: o id, nome, duração, imagem, a tag, um vídeo e a visibilidade do curso. A coluna 'id_Curso' corresponde ao id de cada curso, sendo único para cada curso, guardando um id para cada conteúdo. O nome do curso irá ser guardado em formato String, contendo o título do curso. A duração do curso, que corresponde ao tempo que dura esse curso, é guardado em

formato String também. A coluna 'img', guarda uma imagem do curso(neste caso o logótipo do curso), nos mais diferentes formatos(png, jpg,etc). A required_role corresponde à role de cada curso, quando é criado um curso, apenas com a role certa é que um dado utilizador pode aceder a esse curso. Um exemplo simples é caso o curso criado tenha a role 3, mas o utilizador contém a role 2, ele não pode ver os conteúdos que o curso contém. Esta coluna é guardada em int(10), ou seja, é um int. A 'tag' permite associar os diferentes cursos, sendo muito útil para apagar um curso, e até mesmo alterar a sua visibilidade. A coluna 'vídeo' permite guarda vídeos, é guardado em formato String. A última coluna, de nome 'visible' permite alterar a visibilidade do curso, sendo público ou privada. É guardado em formato de int(10), sendo que, foi tratado como um booleano, no qual, 0 corresponde a um curso estar privado, e o valor de 1, corresponde ao curso esta em público.



	id_curso	nome_curso	duracao	img	required_role	tag	video	visible
<input type="checkbox"/> Edit <input type="checkbox"/> Copiar <input type="checkbox"/> Apagar	3	java	00:30:00	java_logo.png	2	java	java_intro.mp4	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copiar <input type="checkbox"/> Apagar	2	Python	01:30:00	python_logo.png	3	python	python_1.mp4	1

Figura 7: Tabela Base de Dados 'curso'

7.7 curso_conteudo

Esta tabela apenas contém duas colunas, essas colunas são: 'id_curso' e 'id_conteudo'. Como ambos os ids de cada coluna são únicos, é mais fácil associar um curso ao seu respetivo conteúdo, distinguindo o id do curso, ao id do seu conteúdo. Estes dois ids são guardados em formato de int.



id_curso	id_conteudo
3	2
2	3

Figura 8: Tabela Base de Dados 'curso'

7.8 email_accounts

Esta tabela permite guardar as informações acerca do email, sendo possível enviar emails com este email aqui colocado. Esta tabela contém 10 colunas, essas colunas são: um id, accountName, smtpServer, port, useSSL, timeout, loginName, password, email e displayName. A coluna corresponde ao id do email, ficando guardado o id correspondente a este email, sendo este id único. A coluna 'accountName' corresponde ao nome da conta, neste caso foi dado o nome de "Gmail-SMI". A coluna 'smtpServer' corresponde ao servidor que irá ser usado. Nesta caso foi usado

o protocolo smtp, que corresponde ao protocolo de transferência simples de emails, podendo ser transferido entre dois dispositivos computacionais. A coluna 'port' corresponde ao porto que irá ser usado para efetuar a transferência dos emails, para o nosso projeto, o port escolhido foi o 465. A coluna 'useSSL', querendo dizer Secure Sockets Layer, é um protocolo de segurança projetado para fornecer segurança nas comunicações. A coluna 'loginName' corresponde ao email de login, neste caso, é este o email que será usado para efetuar o login. A coluna 'password' corresponde à pass do nosso email. A coluna 'email' corresponde ao email, onde irá ser enviado um email. A última coluna 'displayName' corresponde ao assunto do email. neste uma mensagem dentro do email.

id	accountName	smtpServer	port	useSSL	timeout	loginName	password	email	displayName
1	Gmail - SMI	smtp.gmail.com	465	1	30	iselg13smi@gmail.com	isel1920	iselg13smi@gmail.com	Sistemas Multimédia para a Internet

Figura 9: Tabela Base de Dados 'curso'

7.9 Imagem

A tabela 'images' tem também 3 colunas, uma que guarda o 'id', id esse que é único, tem uma coluna 'id_conteudo', que vai fazer referência aos conteúdos, ou seja coloca a imagem nesse conteúdo e uma coluna onde grava uma string com o nome da imagem, denominada 'img'.

+ Opções			
← T →			
<input type="checkbox"/>	Edita	Copiar	Apagar
1		1	python.png

Figura 10: Tabela Base de Dados 'images'

7.10 Video

A tabela 'video' é igual à tabela das imagens, tem um 'id', que é único, tem uma coluna 'id_conteudo' que vai fazer a ligação entre video e conteudo e tem também uma coluna para guardar o nome de cada video que tem o nome de 'video'.

+ Opções			
← T →			
<input type="checkbox"/>	Edita	Copiar	Apagar
1		1	python_1.mp4

Figura 11: Tabela Base de Dados 'video'

7.11 Inscrição

A tabela 'inscricao' tem duas colunas, uma 'id' para guardar um id único para cada inscrição e um 'idCurso' que faz uma referência entre uma inscrição e um curso, por esse id.



id	idCurso
1	1

Figura 12: Tabela Base de Dados 'inscricao'

8 Produto Final

8.1 Página Principal

Como primeiro passo no nosso projeto, criamos uma página principal, dando uma introdução ao utilizador, do que o tema do nosso website trata. Como já foi referido anteriormente, o nosso website consiste em aprender diferentes linguagens, dando ao utilizador a possibilidade de escolher que linguagem aprender, das mais diversas linguagens que existem.

Design

Em baixo, segue-se uma figura do aspeto da nossa página principal:

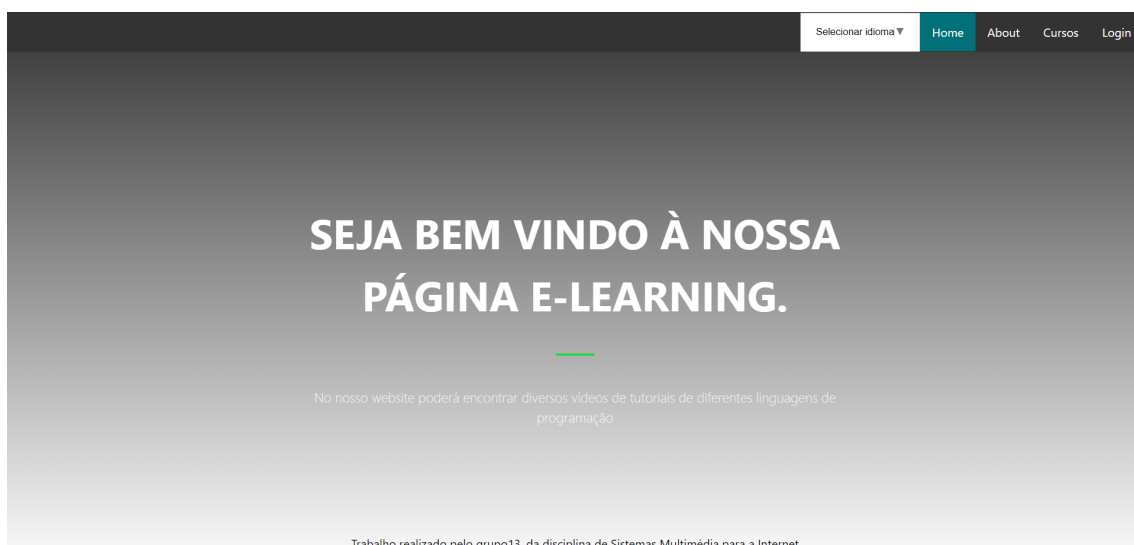


Figura 13: Página Principal

Como é possível de visualizar existe um banner, que contém uma ligação para as diferentes páginas, assim como uma opção para mudar o idioma. Os idiomas aqui presentes são: português(por pré-defeito), inglês, espanhol e francês.

Implementação

Para a implementação desta página, recorreremos a `html(nav,ul,li,a href)`, para navegar nas diferentes páginas e apresentar o respetivo conteúdo na nossa página principal. Foi também aplicado `css`, para dar estilo à nossa página principal. O que é aqui de salientar nesta página, é o facto de que quando o utilizador efetuou o login, ou seja, o que queremos dizer com isto é que, de início quando o utilizador entra no nosso website, aparece a opção "login", mas após efetuar o login (assumindo que fez o registo), o texto é mudado para logout, e irá ficar assim no resto das nossas páginas. Para isso usamos a função do php "SESSION", verificando quando é que o utilizador efetuou o login. Caso tenha efetuado o login, o texto do banner onde diz "log in" é alterado para "log out", redirecionando também para o ficheiro "logout" para efetuar o logout do website. Caso o utilizador tenha clicado na opção "log out", a sessão é terminada, mudando o texto para "log in", sendo redirecionado para a página de log in, caso pretenda efetuar novamente o registo de log in.

Como última implementação desta página, iremos explicar como efetuamos a mudança de idioma. Para isso recorreremos à linguagem javascript.

Para fazer a tradução das páginas em várias linguagens, foi feita uma script de ligação à API do Google Tradutor, que permite fazer a tradução integral de toda a página apresentada. É selecionada a linguagem predefinida da página, assim como as linguagens que se pretende que sejam apresentadas na combo box, para ser feita a tradução. Quando uma linguagem diferente da predefinida é selecionada, é feita a tradução automaticamente. A referência "notranslate", serve para não traduzir algumas partes da página que não são traduzíveis.

O código que se segue permite verificar esta explicação:

```
<?php

if (!isset($_SESSION)) {
    session_start();
}

$_SESSION['locationAfterAuth'] = "../Conteudos/paginaConteudos/
conteudosWebsiteCursos.php";

$text = "";
$link = "";
```



```
if (isset($_SESSION['username']) ) {
    $text = "Logout";
    $link = "../Login/logout.php";
}
else{
    $text = "Login";
    $link = "../Login/processFormLogin.php";
}

<ul>
    <?php echo "<li><a href='". $link. "'><span class='nottranslate'>". $text. "
</span></a></li>"?>
    <li>
        <a href="../Conteudos/paginaConteudos/contedosWebsiteCursos.php">
            Cursos</a>
        </li>
    <li>
        <a href="../About/about.php"><span class='nottranslate'>About</span></a>
        </li>
    <li>
        <a class="active" href="../PaginaPrincipal/PaginaPrincipal.php">
            <span class='nottranslate'>Home</span></a></li>
    <li><a id="translate"><div id="google_translate_element"></div></a></li>
</ul>

<script type="text/javascript">
    function googleTranslateElementInit() {
        new google.translate.TranslateElement(
            {pageLanguage: 'pt', includedLanguages: 'en,es,fr,pt',
             layout: google.translate.TranslateElement.InlineLayout.SIMPLE},
            'google_translate_element');
    }
</script>
<script type="text/javascript" src="//translate.google.com/translate_a/
element.js?cb=googleTranslateElementInit"></script>

?>
```

Listing 1: paginaPrincipal.php

8.2 About

Foi também criada uma página de nome "About" que contém as informações acerca dos autores deste projeto, assim como a localização(em google maps) da faculdade estudada, assim como interação com redes sociais, tais como: uma página de facebook, que nos leva à página de grupo da faculdade, e outra que corresponde ao website da respetiva faculdade.

Design

Nas figuras seguintes é possível de visualizar a página about que foi construída:

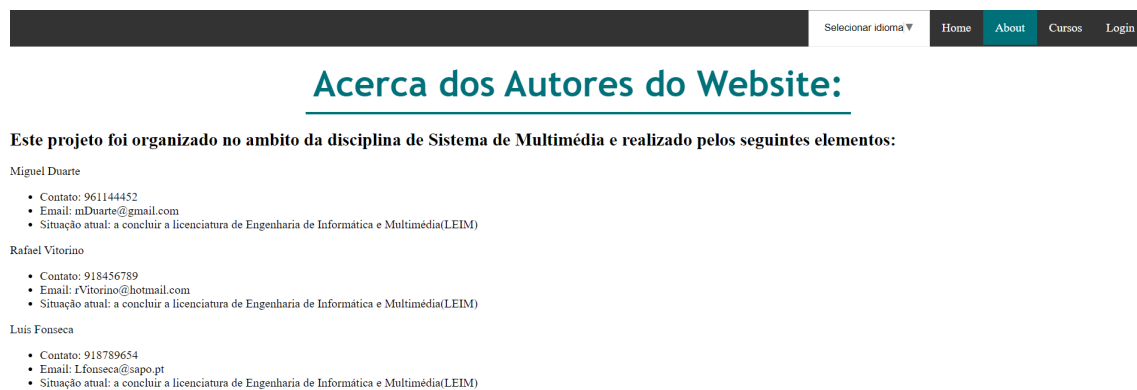


Figura 14: About acerca dos autores do projeto

Na imagem de cima podemos verificar as informações acerca dos autores do projeto.



Figura 15: About com o google maps e as respetivas redes sociais

Na imagem de cima podemos verificar a localização da faculdade, assim como o logo da faculdade e do facebook.

Implementação

Para a implementação desta página, recorreremos a html(nav,ul,li,a href), para navegar nas diferentes páginas e apresentar o respetivo conteúdo na nossa página principal. Foi também aplicado css, para dar estilo à nossa página de about. O que é de salientar é o facto de como foi efetuado a implementação do google maps. Para efetuar este ponto, foi criado um método, em javascript, de nome myMap(), este método permite ler as coordenadas(latitude e longitude) de uma localização dentro do google Maps.

Em baixo, é possível de ver o scripy associado a esta página:

```
<h2> Este projeto foi organizado no  
ambito da disciplina de Sistema de Multim dia  
e realizado pelos seguintes elementos:</h2>  
<p>Miguel Duarte</p>  
<ul>  
  <li>Contato: 961144452</li>  
  <li>Email: mDuarte@gmail.com</li>  
  <li>Situa o atual: a concluir a licenciatura de Engenharia de  
    Inform tica e Multim dia(LEIM)</li>  
</ul>  
  
<p>Rafael Vitorino</p>  
<ul>  
  <li>Contato: 918456789</li>  
  <li>Email: rVitorino@hotmail.com</li>  
  <li>Situa o atual: a concluir a licenciatura de Engenharia de  
    Inform tica e Multim dia(LEIM)</li>  
</ul>  
  
<p>Lu s Fonseca</p>  
<ul>  
  <li>Contato: 918789654</li>  
  <li>Email: Lfonseca@sapo.pt</li>  
  <li>Situa o atual: a concluir a licenciatura de Engenharia de  
    Inform tica e Multim dia(LEIM)</li>  
</ul>  
  
<p>Caso tenha interesse em nos visitar , pode ser visto a localiza o  
da faculdade na imagem de baixo:</p>  
<div id="googleMap" style="width:50%;height:400px;float:_center"></div>  
<p>Para mais informac es acerca da faculdade, clique na imagem em baixo,  
ou ent o selecione a imagem do facebook, caso tenha alguma d vida!</p>
```

```
<div id="isel" class="row">
  <div class="column">
    <a href="https://www.isel.pt/en/">
      
    </a>
  </div>

  <div class="column">
    <a href="https://www.facebook.com/isel.pt/">
      
    </a>
  </div>

</div>

<script>
function myMap() {
var mapProp= {
  center:new google.maps.LatLng(38.756107, -9.116923),
  zoom:17,
};
var map = new google.maps.Map(document.getElementById("googleMap"),mapProp);
}

function googleTranslateElementInit() {
  new google.translate.TranslateElement(
    {pageLanguage: 'pt', includedLanguages: 'en,es,fr,pt',
      layout: google.translate.TranslateElement.InlineLayout.SIMPLE},
    'google_translate_element');
}
</script>
<script type="text/javascript" src="//translate.google.com/translate_a/
element.js?cb=googleTranslateElementInit"></script>

<script src="https://maps.googleapis.com/maps/api/
js?key=AIzaSyBE58e0tKkmZoBqHtKZekaY_QSUMuURaKY&callback=myMap"></script>
```

Listing 2: about.php

8.3 Pagina Geral com os Cursos

Iremos passar agora para a explicação de como foi efetuada a página referente aos cursos apresentados.

Funcionalidades

Esta página contém várias utilidades, tanto para o utilizador, como para o admin.

Para o utilizador ela permite:

- Subscriver a um curso;
- Visualizar os conteúdos dentro do website que estejam públicos;
- Ao clicar num curso, tem de efetuar o login para aceder ao seu conteúdo(caso nao tenha conta, tem de efetuar o registo);
- Pesquisar os diferentes cursos que existem;

Para o administrador ela permite:

- Criar um curso;
- Apagar um curso;
- Alterar a "role" de um utilizador;
- Alterar a visibilidade do curso(público ou privado);
- Alterar o acesso à Base de Dados

Design

Nas figuras seguintes é possível de visualizar a página Cursos que foi construída:

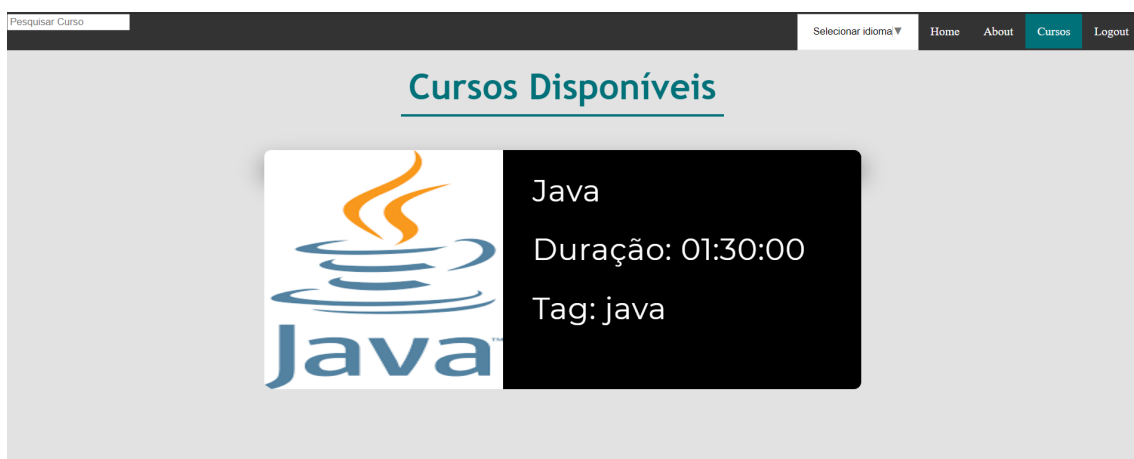


Figura 16: Design da página dos Cursos em geral, com perspetiva no utilizador

Como podemos observar na figura de cima, quando estamos na perspectiva de utilizador, apenas é possível de ver o curso que está público, essa visibilidade dos cursos pode ser controlada pelo administrador, que mais para a frente vai ser explicada com mais pormenor.



Figura 17: Design da página dos Cursos em geral, com perspectiva do administrador

Como podemos observar na figura de cima, quando estamos na perspectiva de administrador, podemos ver o menu de lado, com as 4 seções, que iram ser explicadas mais para a frente. Também é possível de ver quais são os cursos que estão públicos ou privados.

De seguida iremos apresentar as diferentes janelas que o administrador contém para efetuar as 4 tarefas que foram mencionadas:

Para o administrador criar uma seção, foi criada uma janela com a seguinte figura:

Criar Secção

Nome Curso:

Duracao

Imagem

 Nenhum ficheiro seleccionado

Role

Tag

Video

 Nenhum ficheiro seleccionado

☐ Público

☐ Privado

Criar

Fechar

Figura 18: Janela para criar um curso, na página geral dos cursos

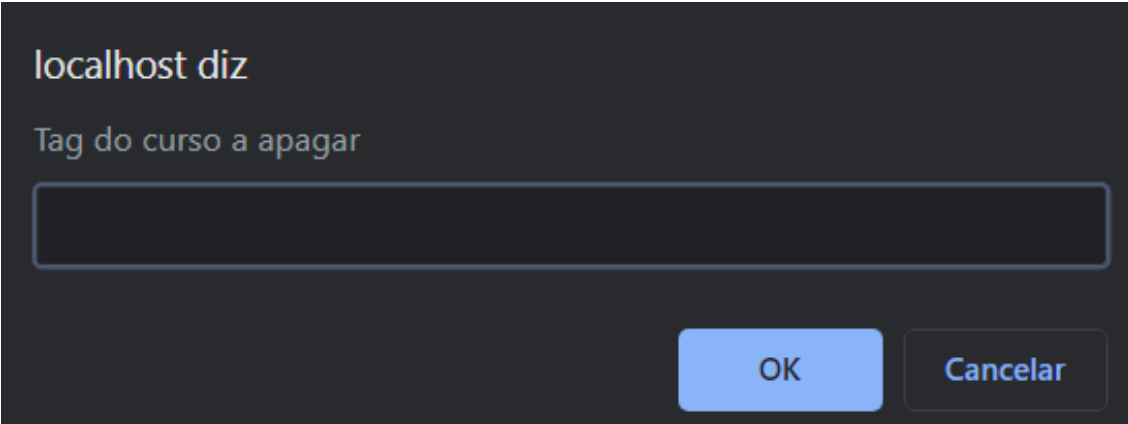
Como é possível de ver na figura de cima, criamos esta janela para permitir ao administrador criar uma seção, seção essa que permite criar um curso novo. Essa seção contém os seguintes elementos:

- Nome Curso: permite dar um título ao curso;
- Duração: corresponde ao tempo que o curso irá conter;
- Imagem: permite dar upload a uma imagem(neste caso um logo da linguagem

correspondente);

- Role: permite atribuir uma role a esse curso. Essa role, caso seja 1 ou 2, ou seja, caso a role seja do utilizador, ele consegue ver, ou não certos cursos, mas não consegue ver a visibilidade desse curso, caso esteja público ou privado. Caso a role seja 3 ou 4, ou seja, a do administrador, ele consegue ver todos os cursos que foram criados, até mesmo a visibilidade desse curso, caso seja público ou privado;
- Vídeo: permite fazer um upload dum vídeo;
- Visibilidade: permite modificar a visibilidade do curso, sendo privado ou público;

De seguida iremos apresentar como procedemos para o administrador apagar um curso nesta página:



The image shows a dark-themed web interface. At the top, it says 'localhost diz'. Below that is the label 'Tag do curso a apagar' in a light blue font. Under the label is a large, empty rectangular text input field with a light blue border. At the bottom right of the form are two buttons: a blue button with the text 'OK' and a grey button with the text 'Cancelar'.

Figura 19: Prompt para apagar um curso

Como se pode ver na figura de cima, usamos o prompt(irá ser explicado a seguir). Caso o administrador deseje apagar esse curso, necessita apenas de colocar a tag desse respetivo curso.

De seguida iremos apresentar como efetuamos a página para o administrador, alterar a role de cada utilizador:

Admin Dashboard

Email da Conta:

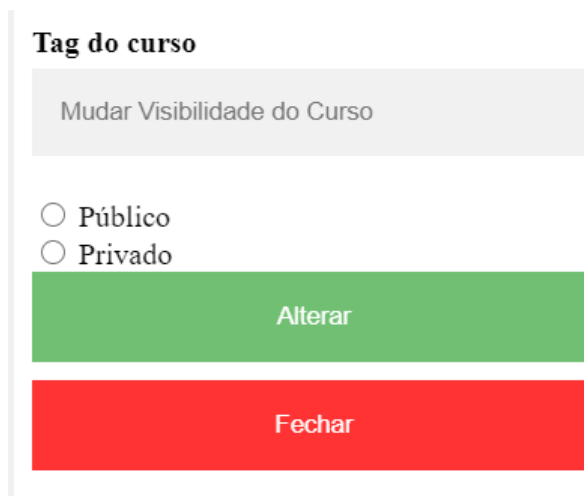
Role a atribuir:

Alterar

Figura 20: Página com alteração role do utilizador

Como podemos ver na figura de cima, a maneira que o nosso grupo achou mais prático, foi alterar a role desse utilizador através do email. Ao escrever o email desse utilizador, é então possível alterar a role desse respetivo utilizador.

No final desta página, mostramos como foi feita a alteração da visibilidade dos cursos criados nesta página:



The image shows a web form titled "Tag do curso". Below the title is a light gray button labeled "Mudar Visibilidade do Curso". Underneath this button are two radio button options: "Público" and "Privado". Below the radio buttons are two large, rectangular buttons: a green one labeled "Alterar" and a red one labeled "Fechar".

Figura 21: Janela criada para alterar a visibilidade dos cursos

Como se pode ver na figura de cima, podemos verificar que para alterar a visibilidade do curso, apenas é necessário escrever a tag do curso, e ao escrever a tag desse curso, pode ser alterada a sua visibilidade, tanto para público como para privado.

De seguida, apresenta-mos a página do administrador poder alterar o acesso à base de dados na figura abaixo:



The image shows a web form titled "Alterar Acesso da Base de dados". Below the title are five input fields, each with a label and a placeholder text: "LocalHost:" with "Introduzir LocalHost", "Porto:" with "Introduzir Porto", "Database:" with "Introduzir Base Dados", "Username:" with "Introduzir Username", and "Password:" with "Introduzir Password". Below these fields is a dark gray button labeled "Alterar".

Figura 22: Página para alterar o acesso à base de dados do projeto

Como se pode verificar na figura de cima, temos as 5 opções que estão de acordo com o ficheiro ".htconfig.xml". Sendo esse ficheiro, um ficheiro xml, que apresenta 5 linhas, que são:

- LocalHost: o nome do host;
- Porto: o número do porto;
- Database: o nome da base de dados associada ao phpMyAdmin;
- Username: o nome do utilizador associado a essa base de dados;
- Password: a pass associada a esse utilizador;

Implementação

A implementação desta página consiste num conjunto vago de scripts para a execução de cada uma das tarefas aqui faladas.

Foi então criado e implementado o ficheiro `conteudosWebsiteCursos.php`. Como passo inicial, fizemos os imports necessários, e usar os métodos do php para fazer a ligação à base de dados.

De seguida foi feito o processo de associar um id a uma role. Para isso usamos a condição `if(isset($_SESSION[id]))`. Caso a sessão correspondente a esse id seja verdadeira, é criada uma query, sendo assim mais fácil de manipular as roles e associar a cada utilizador. Para isso é selecionado na base de dados as roles de cada utilizador (função `FROM`) e de seguida selecionar a coluna que corresponde à role de cada utilizador (função `SELECT`), no final é necessário retirar o id, para isso localizamos o id dentro da nossa base de dados (pela função do SQL `WHERE`), no final desta query, foi necessário usar a função do php de nome `SESSION`, com um respetivo id, sendo assim possível de saber qual o id de sessão associado. No final são usados os métodos do php: `mysqli_query` que aplica uma query na base de dados, ou seja, faz a ligação à base de dados e o método `mysqli_fetch_array`, que irá retornar o resultado dessa linha, no final é adicionado a role a cada a utilizador, à coluna da base de dados.

De seguida, procedemos à realização de como podemos distinguir o administrador e o simpatizante, dos outros utilizadores. Para isso fazemos a condição `if($userRole <= 2)`, caso esta condição seja verdadeira, então aparece um menu (criado em html, com `navs`, e `ul`), com as quatro opções já apresentadas (que pode ser visto na figura 10).

De seguida foi criado a função em php `runmyFunction($requiredRole, $nextPage, $idC)` que permite verificar a role de cada utilizador (se é um administrador, ou um utilizador, e caso seja o utilizador, avisar que não tem acesso a esse curso). Para isso a implementação é idêntica à de cima, a única diferença é que neste caso verificamos a role do utilizador. Pela condição `if($userRole <= $requiredRole)`, é verificado qual a role do utilizador:

- role = 1 ou role = 2, o utilizador não tem acesso aos conteúdos do respetivo curso;
- role = 3 ou role = 4, o utilizador e o administrador podem visualizar os conteúdos do respetivo curso, sendo depois redirecionado para a página do conteúdo desse curso;

Depois deste método criado, iremos buscar os valores do role, do "nxt", neste caso da próxima página, e do seu respetivo, primeiro usando o método isset do php, para verificar se estes valores não retornam nada, e o método GET do php, para ir buscar esse respetivo valor.

Iremos agora passar para a explicação de como procedemos para criar uma nova seção(ou seja, ter a possibilidade do administrador poder criar e adicionar um novo curso)Para isso, criamos uma base de dados de nome "curso" que irá conter todas as informações acerca do curso a ser criado. Mais uma vez necessitamos os métodos do php para ligar à base de dados, e criar uma query capaz de ler a tabela, lendo toda a informação lá nela contida. De seguida, é usada o método mysqli-fetch-array para adicionar as novas informações na nossa base de dados. De seguida, são lidas todas as informações da nossa base de dados, desde o id do curso, até à imagem. No final é tratada da visibilidade do curso. Para isso são criados duas variáveis de nomes: *aceder* e *isVisible*, sendo tratadas como booleanos. Estas variáveis iram ser-nos muito úteis para fazer a representação da janela, quando o administrador necessitar de alterar a visibilidade do curso. Caso a variável *visible* seja igual a 1, o valor do *aceder* é alterado para 1(valor verdadeiro). É também verificado em que situação é que temos a falar do utilizador ou do administrador,e para isso é feita a condição `if(userRole <=2)`. Caso *userRole* seja igual a três ou quatro, incrementa-se as variáveis *aceder* e *isAsmin*. No final fazemos a condição `if(aceder == 1)`. Caso seja verdadeira a condição, então o menu de alterar a visibilidade do curso é apresentada.

Para a criação das páginas de criar uma nova seção e alterar a visibilidade do curso, foram usados labels, com input, para registar o que é escrito.

Como última explicação desta página, iremos agora explicar a implementação dos métodos criados em javascript

- `novaSeccao()` e `hiddenCourse()` que permitem colocar a janela da nova seção e de mudar a visibilidade do curso em bloco;
- `closeForm()` e `closeFormVisible()` que permitem esconder a janela associada a essa seção.
- `deleteForm()` que permite ligar o prompt, fazendo aparecer uma mensagem

”Tag do curso a apagar);

- `changeEvent()` que permite esconder várias categorias, quando é apenas selecionado um curso. Por exemplo temos três cursos: Java, HTML e Python. No motor de busca, caso seja escolhido Java, os outros dois cursos serão escondidos;
- `myFunction()` que permite pesquisar pelos cursos todos que estejam dentro da página geral dos cursos, pela sua respetiva tag.

```
$_SESSION['locationAfterAdd'] = "conteudosWebsiteCursos.php";

<?php

require_once( "../Lib/lib.php" );
require_once( "../Lib/db.php" );

if (!isset($_SESSION)) {session_start();}

$_SESSION['locationAfterAdd'] = "conteudosWebsiteCursos.php";

dbConnect(ConfigFile);
$dataBaseName = $GLOBALS['configDataBase']->db;
mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );

$userRole = 4;

if(isset($_SESSION['id'])){
    $query = "SELECT `role` FROM `{$dataBaseName}`.
    `auth-permissions` where `id`=" . $_SESSION['id'];
    $queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
    if ($queryResult) {
        while ($registro = mysqli_fetch_array( $queryResult )){
            $userRole = $registro['role'];
        }
    }
}

if($userRole <= 2){
    echo '<div class="sidenav">';
    echo '<a onclick="novaSeccao()">-Nova Seção </a>';
    echo '<a onclick="deleteForm()">-Apagar Conteúdo </a>';
    echo '<a href="../Dashboard/dashboard.php">
    -Alterar Role Utilizador</a>';
```

```

        echo '<a_onclick="hiddenCourse()">-Alterar_Visibilidade_do_Curso</a>';
        echo '</div>';
    }
    dbDisconnect();

function runMyFunction(int $requiredRole, String $nxtPage, String $idC) {
    include( "../ensureAuth.php");

    $_SESSION['idC'] = $idC;

    dbConnect(ConfigFile);
    $dataBaseName = $GLOBALS['configDataBase']->db;
    mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );
    $query = "SELECT `role` FROM ` $dataBaseName ` .
    ..... `auth-permissions` where `id`=_ " . $_SESSION['id'];

    $userRole = 4;
    $queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
    if ($queryResult) {
        while ($registro = mysqli_fetch_array( $queryResult )){
            $userRole = $registro['role'];
        }
    }
    dbDisconnect();
    if($userRole <= $requiredRole){
        echo "<script_type='text/javascript'>
    .....window.location.href=_ '../conteudosDiversos/
    .....course_page.php';
    .....</script>";
    }else{
        echo "<script_type='text/javascript'>alert
    .....('N o_tem_authorized_para_aceder');</script>";
    }
}

if (isset($_GET['role']) && isset($_GET['nxt']) && isset($_GET['idC'])) {
    runMyFunction($_GET['role'], $_GET['nxt'], $_GET['idC']);
}

$text = "";
$link = "";

if (isset($_SESSION['username']) ) {

```

```

    $text = "Logout";
    $link = "../../Login/logout.php";
} else {
    $text = "Login";
    $link = "../../Login/processFormLogin.php";
}
?>

<head><title>Cursos</title></head>
<h1 id="title"> Cursos Dispon veis </h1>
<?php
    $uploadDir = "C:/Temp/upload/";
    dbConnect(ConfigFile);
    $dataBaseName = $GLOBALS['configDataBase']->db;
    mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );
    $query = "SELECT_*_FROM_`$dataBaseName`_.._`curso`";
    $img = "";
    $queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
    if ( $queryResult ) {
        while ( $registro = mysqli_fetch_array( $queryResult ) ) {
            $idCurso = $registro['id_curso'];
            $nome_curso = $registro['nome_curso'];
            $duracao = $registro['duracao'];
            $img = $registro['img'];
            $required_role = $registro['required_role'];
            $video = $registro['video'];
            $nxtPage = "course_page";
            $tag = $registro['tag'];
            $visible = $registro['visible'];
            $aceder = 0;
            $isAdmin = 0;
            if ( $visible == 1 ) {
                $aceder = 1;
            }
            if ( $userRole <= 2 ) {
                $aceder = 1;
                $isAdmin = 1;
            }
            if ( $aceder == 1 ) {
                echo "<div_id='". $tag ."'_class='card'>";
                echo "<div_class='card_left'>";
                echo "<a_href='./conteudosWebsiteCursos.php?";
                echo ".....role=".$required_role."";

```

```

.....&nxt=". $nxtPage."
.....&idC=". $idCurso." '>\n";
            echo "<img_width='225'_height='320'
.....src=_ './getThumbnail.php?
.....idC=" . $idCurso. "'>";

            echo "</a>\n";
            echo "</div>";
            echo "<div_class='card_right'>";
            echo "<h1>". $nome_curso. "</h1>";
            echo "<h1>Dura o:_" . $duracao . "</h1>";
            echo "<h1>Tag:_" . $tag . "</h1>";
            if($isAdmin){
                $tipo = "Privado";
                if($visible == 1){$tipo = "P blico";}
                echo "<h1>Tipo:_" . $tipo . "</h1>";
            }
            echo "</div>";
            echo "</div>";
        }
    }
}
dbDisconnect();
?>

<link rel="stylesheet" href="cssMain/sideMenu.css">
<link rel="stylesheet" href="cssMain/style.css">
<script>theVideo = document.getElementById( "my_video" );</script>
<link rel="stylesheet" href="cssMain/mainPage.css">

<ul>
<?php echo "<li><a_href=' " . $link. "'>
.....<span_class='nottranslate'>". $text. "
.....</span></a></li>" ?>
    <li><a class="active">Cursos</a></li>
    <li><a href=".../About/about.php">
    <span class='nottranslate'>About
    </span></a></li>
    <li><a href=".../PaginaPrincipal/PaginaPrincipal.php">
    <span class='nottranslate'>Home</span>
    </a></li>

```



```
<li class="li" <a id="translate">
<div id="google_translate_element">
</div></a></li>
<input type="text" id="pesquisar_curso"
placeholder="Pesquisar_Curso"
onkeyup="myFunction()">
</ul>

<div class="form-popup" id="myForm">
  <form action="./newCourse.php" class="form-container"
    enctype="multipart/form-data"
    method="POST"
    name="FormUpload">
    <h1>Criar Sec  o </h1>
    <label><b>Nome Curso:</b></label>
    <input type="text" placeholder="Introduzir_Nome_do_Curso"
name="nome_curso" required>

    <label><b>Duracao</b></label>
    <input type="text" placeholder="Introduzir_Dura  o_do_Curso"
name="duracao" required>

    <label><b>Imagem</b></label>
    <input type="file" size="64"
placeholder="Introduzir_Imagem_do_Curso"
name="imageFile" required>

    <label><b>Role</b></label>
    <input type="text" placeholder="Introduzir_Role_do_Curso"
name="role" required>

    <label><b>Tag</b></label>
    <input type="text" placeholder="Introduzir_Tag_do_Curso"
name="tag" required>

    <label><b>Video</b></label>
    <input type="file" size="64"
placeholder="Introduzir_Video_do_Curso"
name="videoFile" required>

    <input type="radio" name="privacy" value="1">
    <label for="public"> P blico </label><br>
```

```
<input type="radio" name="privacy" value="0">
<label for="private"> Privado</label><br>

<button type="submit" class="btn">Criar</button>
<button type="button" class="btn_cancel"
onclick="closeForm()">Fechar</button>
</form>
</div>

<div class="form-popup" id="myFormVisible">
  <form action="/changeVisibleCourse.php" class="form-container"
    enctype="multipart/form-data"
    method="POST"
    name="FormUpload">
    <label><b>Tag do curso</b></label>
    <input type="text" placeholder="Mudar_Visibilidade_do_Curso" name="
      <input type="radio" name="privacy" value="1">
        <label for="public"> P blico </label><br>
      <input type="radio" name="privacy" value="0">
        <label for="private"> Privado</label><br>
    <button type="submit" class="btn">Alterar</button>
    <button type="button" class="btn_cancel"
    onclick="closeFormVisible()">Fechar</button>
  </form>
</div>

<script>
function novaSeccao() {
  document.getElementById("myForm").style.display = "block";
}

function closeForm() {
  document.getElementById("myForm").style.display = "none";
}

function deleteForm() {
  var input = prompt("Tag_do_curso_a_apagar", "");
```

```
        if (input != null) {
            window.location="./delete.php?deleteTag="+input;
        }
    }

function closeFormVisible(){
    document.getElementById("myFormVisible").style.display = "none";
}

function hiddenCourse(){
    document.getElementById("myFormVisible").style.display = "block";
}

function changeEvent(){
    var courses = document.getElementsByClassName("Card");
    for(var i=0; i<courses.length; i++){
        if(courses[i].id !== document.getElementById
            ("pesquisar_curso").value)courses[i].style.display='none';
    }
}

function myFunction() {
    var input, filter, a, i, txtValue;
    input = document.getElementById("pesquisar_curso");
    filter = input.value.toUpperCase();
    var courses = document.getElementsByClassName("Card");
    for(var i=0; i<courses.length; i++){
        a = courses[i];
        txtValue = a.id;
        if (txtValue.toUpperCase().indexOf(filter) > -1) {
            courses[i].style.display = "";
        } else {
            courses[i].style.display = "none";
        }
    }
}

function googleTranslateElementInit() {
    new google.translate.TranslateElement
    ({pageLanguage: 'pt', includedLanguages: 'en,es,fr,pt',
      layout: google.translate.TranslateElement.InlineLayout.SIMPLE},
      'google_translate_element');
}
```

```
</script>
<script type="text/javascript"
src="//translate.google.com/translate_a/
element.js?cb=googleTranslateElementInit">
</script>
```

Listing 3: conteudosWebsiteCursos.php

Para ocorrer as mudanças da visibilidade do curso, foi necessário alterar também o parâmetro da visibilidade do curso dentro da base de dados. Para isso foi criado o script `changeVisibleCourse.php`. Como procedimento habitual, inicializamos com os imports necessários, e a ligação à base de dados. De seguida usamos a função do SQL `UPDATE` que permite atualizar o valor da coluna "visible", neste caso, a nossa base de dados será um valor booleano, pois 0 corresponde ao valor de um curso estar privado, e valor 1 do curso estar público. No final são criadas variáveis, que permitem ler o porto, para a sessão atual. É usada a função *header*, que permite fazer a submissão dos dados e transitar para a próxima página.

```
<?php

if (!isset($_SESSION)) {session_start();}

require_once( "../Lib/lib.php" );
require_once( "../Lib/db.php" );

dbConnect(ConfigFile);
$dataBaseName = $GLOBALS['configDataBase']->db;
mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );

if (isset($_POST['tag'])) {
    $query = "UPDATE `curso` SET `visible` =
    '$_POST['privacy']'
    where `tag` = '$_POST['tag']'";
    mysqli_query($GLOBALS['ligacao'], $query);
}
dbDisconnect();
$serverName = "localhost";
$serverPort = 80;
$name = webAppName();
$baseUrl = "http://" . $serverName . ":" . $serverPort;
$baseNextUrl = $baseUrl . $name;
$nextUrl = $_SESSION['locationAfterAdd'];
header("Location:_" . $baseNextUrl . $nextUrl);?>
```

Listing 4: changeVisibleCourse.php

Com a parte da página geral dos cursos tratada, iremos agora passar para uma explicação mais em detalhe de como inserimos os valores de criar um curso novo dentro da nossa base de dados.

Foi criado o script newCourse.php. Como procedimento habitual, inicializamos com os imports necessários, e a ligação à base de dados. De seguida usamos mais uma vez a função do php isset, neste caso iremos verificar os valores da tabela: nome-curso, duracao, role e tag. De seguida, usamos o método do SQL INSERT, que permite inserir valores da nossa tabela da base de dados. Para associar o valor da linha da base de dados, para ler os valores que são inseridos, tanto para a tag, como para os vídeos que são usados: POST, que retorna o valor do label e FILES, que permite verificar se os ficheiros que estão a ser selecionados são imagens ou vídeos. No final são criadas variáveis, que permitem ler o porto, para a sessão atual. É usada a função *header*, que permite fazer a submissão dos dados e transitar para a próxima página.

```
<?php

if (!isset($_SESSION)) {
    session_start();
}

require_once( "../Lib/lib.php" );
require_once( "../Lib/db.php" );

dbConnect(ConfigFile);
$dbName = $GLOBALS['configDataBase']->db;
mysqli_select_db( $GLOBALS['ligacao'], $dbName );

if (isset($_POST['nome_curso'])
    && isset($_POST['duracao'])
    && isset($_POST['role'])
    && isset($_POST['tag'])) {

    $query = "INSERT INTO `curso`
    (`nome_curso`,`duracao`,`img`,`required_role`,`tag`,`video`,`visible`)
    VALUES('".$_POST['nome_curso'].",'
    ".$_POST['duracao'].",'
    ".$_FILES['imageFile']['name'].",'
    ".$_POST['role'].",'
    '".$_POST['tag'].",'
    ".$_FILES['videoFile']['name'].','
    ".$_POST['privacy'].')";
```

```
        mysqli_query($GLOBALS['ligacao'], $query);
    }
    dbDisconnect();
    //

    include_once './imageUpload.php';
    include_once './videoUpload.php';

    //
    $serverName = "localhost";
    $serverPort = 80;
    $name = webAppName();
    $baseUrl = "http://" . $serverName . ":" . $serverPort;
    $baseNextUrl = $baseUrl . $name;
    $nextUrl = $_SESSION['locationAfterAdd'];
    header("Location:_" . $baseNextUrl . $nextUrl);
?>
```

Listing 5: newCourse.php

De seguida, iremos passar para uma explicação mais em detalhe de como apagamos um dado curso na nossa base de dados.

Foi criado o script delete.php. Como procedimento habitual, inicializamos com os imports necessários, e a ligação à base de dados. De seguida usamos a função do SQL DELETE que permite apagar uma linha da base de dados. Nesta caso, quando é lida um valor da tag inserida, a linha correspondente a essa tag é apagada, No final são criadas variáveis, que permitem ler o porto, para a sessão atual. É usada a função *header*, que permite fazer a submissão dos dados e transitar para a próxima página.

```
<?php

require_once( "../Lib/lib.php" );
require_once( "../Lib/db.php" );

if (!isset($_SESSION)) {
    session_start();
}

dbConnect(ConfigFile);
$dataBaseName = $GLOBALS['configDataBase']->db;
mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );
```

```
$query = "DELETE_FROM_`curso`_WHERE_tag='". $_GET["deleteTag"]. "'";
$queryResult = mysqli_query( $GLOBALS['ligacao'], $query );

$serverName = "localhost";
$serverPort = 80;
$name = webAppName();
$baseUrl = "http://" . $serverName . ":" . $serverPort;
$baseNextUrl = $baseUrl . $name;
$nextUrl = "conteudosWebsiteCursos.php";

header("Location:_" . $baseNextUrl . $nextUrl);

?>
```

Listing 6: delete.php

De seguida, iremos explicar a funcionalidade do administrador poder mudar a role de cada utilizador. Para isso foram criados dois scripts:

Um de nome dashboard.php, que apresenta uma página html, com dois inputs, o primeiro como sendo a escrita do email do dado utilizador, e outro, uma lista com os quatro tipos de funções.

```
<?php
    $text = "";
    $link = "";

    if (isset($_SESSION['username']) ) {
        $text = "Logout";
        $link = "../Login/logout.php";
    }
    else{
        $text = "Login";
        $link = "../Login/processFormLogin.php";
    }
?>
<link rel="stylesheet" href="cssMain/mainPage.css">
<head>
    <title>Admin Dashboard</title>
</head>

<ul>
    <?php echo "<li><a_href='". $link. "'>". $text. "</a></li>" ?>;
    <li><a>Cursos</a></li>
    <li><a href="../About/about.php">About</a></li>
    <li><a href="../PaginaPrincipal/PaginaPrincipal.php">Home</a></li>
</ul>
<h1 id="title">Admin Dashboard</h1>
<div class="cont">
<form action="../changeAccount.php" method="POST">
    <label><b>Email da Conta:</b></label>
    <input type="email"
        placeholder="Introduzir_Email_da_Conta"
        name="email_conta" required>
    <br>
    <br>
    <label><b>Role a atribuir:</b></label>
    <select name="role" id="cars">
        <option value="1">Admin</option>
        <option value="2">Simpatizante</option>
        <option value="3">Utilizador</option>
    </select>
    <button type="submit">Alterar</button>
</form>
</div>
```

Listing 7: dashboard.php

O outro script de nome `changeAccount.php`, que permite alterar a role do dado utilizador dentro da base de dados. Como procedimento habitual, inicializamos com os imports necessários, e a ligação à base de dados. Neste caso foi criada uma query, que irá dar-nos jeito para associar o id ao respetivo email. Para ir buscar o id, usamos a função `SELECT`, e o `WHERE` para localizar o email, igualando ao método `POST` do php, para ir buscar o valor do `POST` passado no html no label. De seguida, é feito `mysqli-fetch-array` para adicionar o valor à base de dados. No final é criada outra query, mas neste caso é usada a função `UPDATE` para fazer a alteração da role na base de dados, a função `SET` para aplicar esse valor, igualando ao valor de `POST` usado no label, e mais uma vez a função `WHERE`, que necessitamos de localizar o id associado a essa base de dados. No final são criadas variáveis, que permitem ler o porto, para a sessão atual. É usada a função *header*, que permite fazer a submissão dos dados e transitar para a próxima página.

```
<?php
```

```
if (!isset($_SESSION)) {
    session_start();
}

require_once( "../Lib/lib.php" );
require_once( "../Lib/db.php" );

dbConnect(ConfigFile);
$dataBaseName = $GLOBALS['configDataBase']->db;
mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );

$query = "SELECT `id` FROM `".$dataBaseName.``.`auth-basic` where `email`=_(' " .
$id = -1;
$queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
if ($queryResult) {
    while ($registro = mysqli_fetch_array( $queryResult )){
        $id = $registro['id'];
    }
}

$query = "UPDATE `auth-permissions`
SET `role`=_(' " . $_POST['role'] .
" ` where `id`=_(' " . $id . " '";
$queryResult = mysqli_query( $GLOBALS['ligacao'], $query );

$serverName = "localhost";
```

```
$serverPort = 80;
$name = webAppName();
$baseUrl = "http://" . $serverName . ":" . $serverPort;
$baseNextUrl = $baseUrl . $name;
$nextUrl = "dashboard.php";

echo $baseNextUrl . $nextUrl;
header("Location:_" . $baseNextUrl . $nextUrl);
?>
```

Listing 8: changeAccount.php

De seguida iremos explicar a última funcionalidade do administrador, que consiste em alterar o acesso à base de dados. Como ponto de partido, foi criado um ficheiro de nome changeDBaccess.php, que leva 5 forms, que permitem editar as 5 configurações que existem no ficheiro ".htconfig.xml".

```
<?php
if (!isset($_SESSION)) {session_start();}
$text = "";
$link = "";
if (isset($_SESSION['username']) ) {
    $text = "Logout";
    $link = "../Login/logout.php";
}else{
    $text = "Login";
    $link = "../Login/processFormLogin.php";
}
?>
<link rel="stylesheet" href="cssMain/mainPage.css">
<head>
    <title>Admin Dashboard</title>
</head>
<ul>
    <?php echo "<li><a_href='" . $link . "'>" . $text . "</a></li>" ?>;
    <li><a href="../Conteudos/paginaConteudos/conteudosWebsiteCursos.php">
    Cursos</a></li>
    <li><a href="../About/about.php">About</a></li>
    <li><a href="../PaginaPrincipal/PaginaPrincipal.php">Home</a></li>
</ul>

<h1 id="title">Alterar Acesso da Base de dados</h1>

<div class="cont">
```

```
<br><br>
<form action="./processChangeDBaccess.php" method="POST">
<label><b>LocalHost:</b></label>
<input type="localhost" placeholder="Introduzir_LocalHost"
name="localhost" required>
<br>
<label><b>Porto:</b></label>
<input type="porto" placeholder="Introduzir_Porto"
name="porto" required>
<br>
<label><b>Database:</b></label>
<input type="db" placeholder="Introduzir_Base_Dados"
name="db" required>
<br>
<label><b>Username:</b></label>
<input type="username" placeholder="Introduzir_Username"
name="username" required>
<br>
<label><b>Password:</b></label>
<input type="pass" placeholder="Introduzir_Password"
name="pass" required>
<button type="submit">Alterar</button>
</form>
</div>
```

Listing 9: changeDBaccess.php

De seguida criamos outro script de nome processChangeDBaccess.php. Este script permite efetuar as alterações, e processar os dados passado no form. Para editar este ficheiro, iremos usar os métodos referentes a ficheiros do php. Como passo inicial necessitamos de ler o ficheiro, para isso usamos o método "fopen()" que leva dois argumentos, o primeiro que consiste em ler a diretoria do ficheiro(ou apenas o ficheiro), e o outro, o tipo de ficheiro, neste caso será "wb" ou seja, "write binary". De seguida foi necessário guardar a nova informação. Para isso foi criada uma variável de nome \$text, que irá ser-nos útil para guardar a nova informação. Neste caso, como queremos que a informação dentro do ficheiro ".htconfig.xml" seja alterada, é necessário ir buscar esse valor, por isso foi usado o método POST do php, para retornar o valor de cada um dos inputs passados no ficheiro changeDBaccess.php. No final usamos o método "fwrite" que permite escrever e guardar as alterações no próprio ficheiro. Quando tudo foi escrito no ficheiro e guardado, usamos o método "fclose" para terminar de escrever num ficheiro.

<?php

```
if (!isset($_SESSION)) {session_start();}

require_once( "../Lib/lib.php" );
require_once( "../Lib/db.php" );

dbConnect(ConfigFile);
$dataBaseName = $GLOBALS['configDataBase']->db;
mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );

$myfile = fopen("../Lib/Config/.htconfig.xml", "wb");
$text = '<?xml_version="1.0" _encoding="utf-8"?>'.PHP_EOL;
fwrite($myfile, $text);
$text = '<!DOCTYPE DataBase_SYSTEM_"/>'.PHP_EOL;
fwrite($myfile, $text);
$text = '<Config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" '
        . ' _xsi:noNamespaceSchemaLocation="/>'.PHP_EOL;
fwrite($myfile, $text);
$text = '___<DataBase>'.PHP_EOL;
fwrite($myfile, $text);
$text = '____<host>'.$_POST['localhost'].'</host>'.PHP_EOL;
fwrite($myfile, $text);
$text = '____<port>'.$_POST['porto'].'</port>'.PHP_EOL;
fwrite($myfile, $text);
$text = '____<db>'.$_POST['db'].'</db>'.PHP_EOL;
fwrite($myfile, $text);
$text = '____<username>'.$_POST['username'].'</username>'.PHP_EOL;
fwrite($myfile, $text);
$text = '____<password>'.$_POST['pass'].'</password>'.PHP_EOL;
fwrite($myfile, $text);
$text = '___</DataBase>'.PHP_EOL;
fwrite($myfile, $text);
$text = '</Config>'.PHP_EOL;
fwrite($myfile, $text);
fclose($myfile);

$serverName = "localhost";
$serverPort = 80;
$name = webAppName();
$baseUrl = "http://" . $serverName . ":" . $serverPort;
$baseNextUrl = $baseUrl . $name;
$nextUrl = "changeDBaccess.php";
echo $baseNextUrl . $nextUrl;
```

```
header("Location:_" . $baseNextUrl . $nextUrl);?>
```

Listing 10: processChangeDBaccess.php

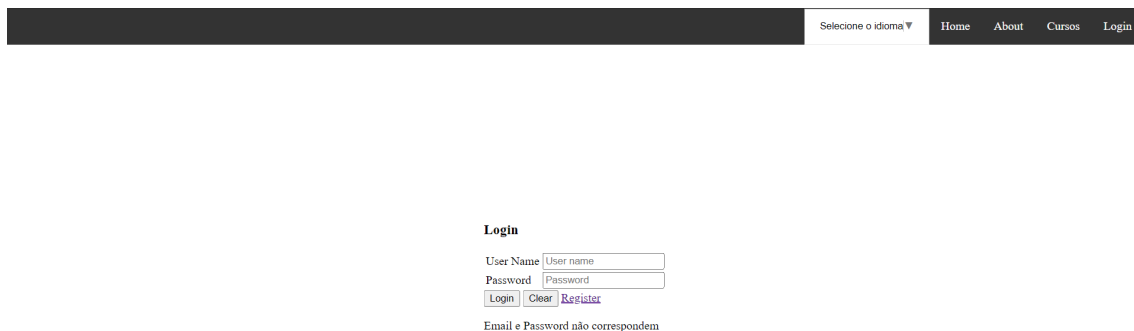
8.4 Login

Design da Página

A página de login tem o seguinte aspeto:

The image shows a web page layout for a login form. At the top, there is a dark horizontal bar. On the right side of this bar, there is a language selector labeled 'Selecione o idioma' with a dropdown arrow, and a navigation menu with links for 'Home', 'About', 'Cursos', and 'Login'. Below the header, the main content area is white. In the center, there is a section titled 'Login'. Under this title, there are two input fields: 'User Name' with a placeholder 'User name' and 'Password' with a placeholder 'Password'. Below these fields are three buttons: 'Login', 'Clear', and 'Register'.

Figura 23: Design da página de login



The image shows a web application header with a dark navigation bar containing links for 'Home', 'About', 'Cursos', and 'Login'. A language selection dropdown is also present. Below the header, a 'Login' section contains two input fields: 'User Name' (containing 'User name') and 'Password' (containing 'Password'). Below these fields are three buttons: 'Login', 'Clear', and 'Register'. A message at the bottom of the login section states 'Email e Password não correspondem' (Email and Password do not match).

Figura 24: Design da página de login após incorreta introdução de dados

Foi também criado uma página para permitir ao utilizador poder mudar a sua palavra pass, ao submeter o seu email.



The image shows a form with the text 'Introduza o seu email' followed by a text input field containing the word 'email'. To the right of the input field is a button labeled 'Alterar'.

Figura 25: Design da página de submeter o seu email

O email com o link associado, ao confirmar a mudança da nova palavra pass pode ser visto na figura de baixo:

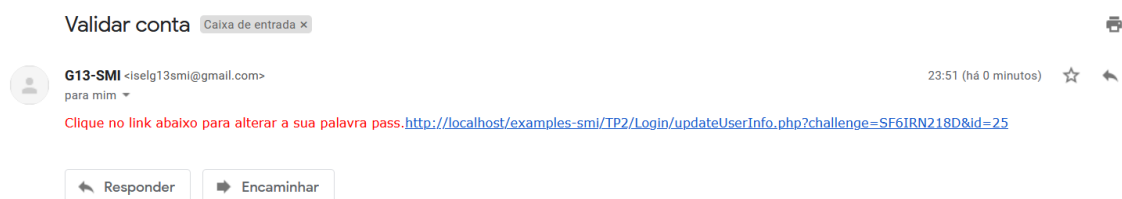


Figura 26: Email a confirmar a mudança da nova palavra pass

No final, é feita mais uma página para o utilizador submeter a nova palavra pass:

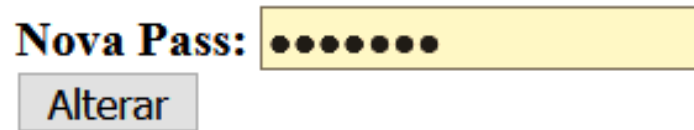


Figura 27: Design da página de submeter a nova pass

Funcionalidades

Na página de login é permitido a quem acede à página que utilize as seguintes funcionalidades:

- Efetuar login no website;
- Transitar para a página de Registo, Home, About e Cursos;
- Fazer alteração da palavra pass, caso o utilizador não se lembre da sua pass;

Para efetuar login, o utilizador deve de introduzir o seu email e password da sua conta que já foi previamente criada. Caso não contenha nenhuma conta, o utilizador deve de ir até á página de registo, onde terá a possibilidade de criar uma nova conta, registando-se assim no website.

Implementação

Na implementação do ficheiro formLogin.php é criada a estrutura html para que o utilizador insira os dados de login e todos os respetivos elementos relativos à navegação entre páginas (navbar, button, etc.). O botão Login irá fazer submit ao form que contém toda a informação introduzida pelo utilizador de forma a que seja possível averiguar se os dados estão corretos.

```
<?php
require_once( "../Lib/lib.php" );

if (!isset($_SESSION)) {
    session_start();
}

$errorMsg = "";

if(isset($_SESSION['loginMsg'])){
```

```
        $errorMsg = $_SESSION['loginMsg'];
        $_SESSION['loginMsg'] = '';
    }

    $serverName = $_SERVER['SERVER_NAME'];
    # $serverName = "localhost";

    $serverPortSSL = 443;
    $serverPort = 80;

    $name = webAppName();

    $nextUrl = "https://" . $serverName . ":" . $serverPortSSL . $name
    . "processFormLogin.php";

    $text = "";
    $link = "";

    if (isset($_SESSION['username']) ) {
        $text = "Logout";
        $link = "../Login/logout.php";
    }
    else{
        $text = "Login";
        $link = "../Login/processFormLogin.php";
    }
?>

<head>
    <link rel="stylesheet" type="text/css" href="CSS/login.css">
    <title>Login</title>

</head>

<link rel="stylesheet" href="cssMain/mainPage.css">

<ul>
<?php echo "<li><a href='". $link. "'><span class='nottranslate'>
". $text. "</span></a></li>" ?>
    <li><a href="../Conteudos/paginaConteudos/contеudos
```



```

<ul><li><a href="#">WebsiteCursos.php</a></li>
<li><a href="#">Cursos</a></li>
<li><a href=" ../About/about.php"><span class='nottranslate'>
About</span></a></li>
<li><a href=" ../PaginaPrincipal/PaginaPrincipal.php">
<span class='nottranslate'>Home</span></a></li>
<li class="li" <a id="translate">
<div id="google_translate_element"></div></a></li>
</ul>

<div class="cont">
<h3>Login</h3>
<form action="<?php echo $nextUrl_?>" method="POST">
<table>
<tr>
<td>User Name</td>
<td><input
type="text"
name="username"
placeholder="User_name"></td>
</tr>
<tr>
<td>Password</td>
<td><input
type="password"
name="password"
placeholder="Password"></td>
</tr>
</table>

<input type="submit" value="Login"> <input type="reset"
value="Clear">
<a href=" ../Register/formRegister.php">Register</a>
<p><?php echo $errorMsg?></p>
<a href="sendEmail.php">
Esqueceu-se da password?
Clique neste link para editar a sua password!
</a>
</form>
</div>

<script type="text/javascript">

```

```
function googleTranslateElementInit() {  
    new google.translate.TranslateElement({pageLanguage: 'pt',  
        includedLanguages: 'en,es,fr,pt',  
        layout: google.translate.TranslateElement.  
            InlineLayout.SIMPLE},  
        'google_translate_element');  
}  
</script>  
<script type="text/javascript"  
src="//translate.google.com/translate_a/element.js?  
cb=googleTranslateElementInit"></script>
```

Listing 11: formLogin.php

Após se submeter o formulário da página formLogin.php, os seus dados são enviados e analisados pelo script processFormLogin.php. Neste script, é efetuada uma ligação à base de dados e compara-se o email e a password. Se estiverem corretos o utilizador será encaminhado até à página dos cursos disponíveis. Caso contrário será mantido na página de formLogin.php mas ser-lhe-á apresentada a informação de que os seus dados estão incorretos.

Para se verificar se os dados estão corretos, obtém o username e a password passados pelo form anterior na forma de POST. Verifica-se através do if(\$userId >0) se a validação deu resultado afirmativo e se sim, coloca-se na sessão o username e o id do mesmo. Isto para que em posteriores páginas, seja possível aceder a elas visto que tanto o id como o username, são elementos essenciais de procura na tabela que contém todos os registos.

```
<?php  
  
require_once( "../Lib/lib.php" );  
require_once( "../Lib/db.php" );  
  
if (!isset($_SESSION)) {  
    session_start();  
}  
  
$serverName = $_SERVER['SERVER_NAME'];  
#$serverName = "localhost";  
  
$serverPort = 80;  
  
$name = webAppName();
```

```
$baseUrl = "http://" . $serverName . ":" . $serverPort;

$baseNextUrl = $baseUrl . $name;

$username = $_POST['username'];
$password = $_POST['password'];

$userId = isValid($username, $password, "basic");
if ($userId > 0) {
    $_SESSION['username'] = $username;
    $_SESSION['id'] = $userId;
    if (isset($_SESSION['locationAfterAuth'])) {
        $baseNextUrl = $baseUrl;
        $nextUrl = $_SESSION['locationAfterAuth'];
        header("Location:_" . ".../Conteudos/paginaConteudos/
conteudosWebsiteCursos.php");
    } else {
        header("Location:_" . ".../Conteudos/paginaConteudos/
conteudosWebsiteCursos.php");
    }
} else {
    $_SESSION['loginMsg'] = "Email_e_Password_nao_correspondem";
    $nextUrl = "formLogin.php";
    header("Location:_" . $baseNextUrl . $nextUrl);
}
?>
```

Listing 12: processFormLogin.php

Além de efetuar login é também possível efetuar logout através do script logout.php. Nesse script é destruída a sessão criada no login.

```
<?php
session_start();
session_destroy();

echo "<META_HTTP-EQUIV=\\"refresh\\"_CONTENT=\\"2;_URL=formLogin.php\\">";
?>

<head>
    <title>Logout...</title>
</head>
```

Listing 13: logout.php

Um dos parâmetros de avaliação deste projeto, era a possibilidade de alterar os dados na base de dados. Neste caso o que fizemos foi o utilizador poder alterar a sua palavra pass, caso se tenha esquecido. Para isso foi usado um "a href", em html, que permite criar uma hiperligação para ser redirecionado para a página de submeter o seu email.

De seguida foi criado o script sendEmail.php, que apenas lê o email de um dado utilizador.

```
<div class="cont">
    <form action="./processSendEmail.php" method="POST">
        <label><b>Introduza o seu email</b></label>
        <input type="email" placeholder="email" name="email" required>
        <button type="submit">Alterar</button>
    </form>
</div>
```

Listing 14: sendEmail.php

A seguir, foi necessário criar um script que pudesse processar os dados. Foi então criado o script processSendEmail.php. Como procedimento habitual, inicializamos com os imports necessários, e a ligação à base de dados. Neste caso foi criada uma query, que irá dar-nos jeito para associar o id ao respetivo email. Para ir buscar o id, usamos a função SELECT, e o WHERE para localizar o email, igualando ao método POST do php, para ir buscar o valor do POST passado no html no label. De seguida, é feito `mysqli_fetch_array` para adicionar o valor à base de dados. No final é criada outra query, mas neste caso é usada a função INSERT para inserir o valor do id e de um challenge passado no link, esses valores iram ser inseridos também com a função VALUES, que permite ler esses valores. No final são criadas variáveis, que permitem ler o porto, para a sessão atual. É usada a função *header*, que permite fazer a submissão dos dados e transitar para a próxima página.

```
<p>Foi enviado um email para alterar a sua password</p>
```

```
<?php
require_once( "../Lib/lib.php" );
require_once( "../Lib/db.php" );

if (!isset($_SESSION)) {
    session_start();
}

dbConnect(ConfigFile);
$dbName = $GLOBALS['configDataBase']->db;
```

```
$serverName = $_SERVER[ 'SERVER_NAME' ];
#$serverName = "localhost";

$serverPort = 80;

$name = webAppName();

$baseUrl = "http://" . $serverName . ":" . $serverPort;

$baseNextUrl = $baseUrl . $name;
mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );

$query = "SELECT `id` FROM ` $dataBaseName ` .
`auth-basic` WHERE `email` = " . $_POST['email'] . "'";
    $idUser = 0;
    $queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
    if ( $queryResult ) {
        while ( $registro = mysqli_fetch_array( $queryResult ) ) {
            $idUser = $registro['id'];
        }
    }

    include './challenge.php';
    $query = "INSERT INTO `confirm_pass`
    ( `id`, `challenge` ) VALUES ( " . $idUser . " , " . $challenge . " )";
    mysqli_query( $GLOBALS['ligacao'], $query );
    $_SESSION['passID'] = $idUser;
    include './confirmNewPassword.php';
    dbDisconnect();
?>
```

Listing 15: sendEmail.php

De seguida foi necessário processar os dados quando é enviado um email para ser alterado a palavra pass. Como procedimento habitual, inicializamos com os imports necessários, e a ligação à base de dados. Neste caso foi criada uma query, que irá dar-nos jeito para associar o id ao respetivo email. Para ir buscar o id, usamos a função SELECT, e o WHERE para localizar o email, igualando ao método POST do php, para ir buscar o valor do POST passado no html no label. De seguida, é feito `mysqli_fetch_array` para adicionar o valor à base de dados. No final é criada outra query, mas neste caso é usada a função INSERT para inserir o valor do id e de um challenge passado no link, esses valores iram ser inseridos também com a

função `VALUES`, que permite ler esses valores. No final são criadas variáveis, que permitem ler o porto, para a sessão atual. É usada a função *header*, que permite fazer a submissão dos dados e transitar para a próxima página.

```
<p>Foi enviado um email para alterar a sua password</p>

<?php
require_once( "../Lib/lib.php" );
require_once( "../Lib/db.php" );

if (!isset($_SESSION)) {session_start();}

dbConnect(ConfigFile);
$dataBaseName = $GLOBALS['configDataBase']->db;
$serverName = $_SERVER[ 'SERVER_NAME' ];

$serverPort = 80;
$name = webAppName();
$baseUrl = "http://" . $serverName . ":" . $serverPort;
$baseNextUrl = $baseUrl . $name;
mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );

$query = "SELECT `id` FROM `".$dataBaseName.``.`auth-basic` WHERE "
        . "`email`='".$_POST['email']."'";
$idUser = 0;
$queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
if ($queryResult) {
    while ($registro = mysqli_fetch_array( $queryResult )){
        $idUser = $registro['id'];
    }
}

include './challenge.php';
$query = "INSERT INTO `confirm_pass`
      (`id`,`challenge`)
      VALUES('".$_idUser."','".$challenge."')";
mysqli_query($GLOBALS['ligacao'], $query);
$_SESSION['passID'] = $idUser;
include './confirmNewPassword.php';
dbDisconnect();
?>
```

Listing 16: sendEmail.php

A seguir, foi tratado o processo de enviar um email a um utilizador. Para isso foi criado o script `confirmNewPassword.php`. Este script permite enviar um email, a ser redirecionado ao utilizador que tem o seu email associado. Para apresentar o conteúdo da mensagem, é criada a variável `$Message`, que contém o corpo da mensagem. Esta variável contém o link que irá ser redirecionado para a página de submeter uma nova password. No final deste link, é associado um challenge, e o id, usando o método do php `SESSION`. De seguida é criada uma query para associar o email a esse respetivo id. Para isso, é usada a função do sql `SELECT`, que permite selecionar a coluna correspondente ao email, a função `FROM` para ler a partir da base de dados, e no final `WHERE`, para detetar o id da base de dados, e associar ao id da sessão. Esta última cláusula é igualada a um `SESSION`, pois queremos sempre saber qual o id da sessão a ser processado. De seguida, é usado método `add_html`, que recebe dois argumentos: a mensagem em html, e a própria mensagem. No final, é usado o método `send()` que permite enviar o respetivo email, passando como argumentos, um `serverSMTP`, um porto, o nome do login, e a pass.

```
<?php
    require_once( "../Lib/lib.php");
    require_once( "../Lib/lib-mail-v2.php" );
    require_once( "../Lib/HtmlMimeMail-class.php" );
    require_once( "../Lib/db.php" );

    $Account = 1;
    $ToName = "";
    $Subject = "Validar_conta";
    $project = explode('/', $_SERVER['REQUEST_URI'])[2];
    $Message = "Clique no link abaixo para alterar a sua palavra pass." .
               "http://localhost/examples-smi/"
               . $project . "/Login/updateUserInfo.php?
    .....challenge=".$_SESSION['$challenge'] .
               "&id=".$_SESSION['passID'];
    $smtpServer = "smtp.gmail.com";
    $mail = new HtmlMimeMail();
    $query = "SELECT `email` FROM
    .....`$dataBaseName`.`auth-basic` WHERE
    .....`id`='".$_SESSION['passID']."'";
    $Email = 0;
    $queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
    if ( $queryResult ) {
        while ( $registo = mysqli_fetch_array( $queryResult ) ) {
            $Email = $registo['email'];
        }
    }
}
```

```
$MessageHTML = <<<EOD
<html>
    <body style="background:_url('background.gif')_repeat;">
        <font face="Verdana,_Arial" color="#FF0000">
            $Message
        </font>
    </body>
</html>
EOD;

$mail->add_html( $MessageHTML , $Message);

$mail->build_message();

$result = $mail->send(
    $smtpServer ,
    1,
    465,
    "iselg13smi@gmail.com",
    "isel1920",
    $ToName ,
    $Email ,
    "G13-SMI",
    "iselg13smi@gmail.com",
    $Subject ,
    "X-Mailer:_Html_Mime_Mail_Class");
?>
```

Listing 17: confirmNewPassword.php

Ao ser selecionado o link do email, o utilizador é redirecionado para uma página, onde pode submeter uma nova palavra pass. Para esta página apenas é criado um form, com uma label, para submeter uma nova palavra pass, usando o input. É de salientar é necessário ir buscar os dados da sessão. A linha `$_SESSION['passID'] = $_GET['id']` permite que seja obtido os dados da sessão anterior, passado no link do email, é também usado o método GET do php, pois é necessário ir buscar o valor do id.

```
<?php
if (!isset($_SESSION)) {session_start();}
$_SESSION['passID'] = $_GET['id'];
```



```
?>

<head>
  <link rel="stylesheet" type="text/css" href="CSS/login.css">
  <title>Update User Info</title>
</head>

<ul>
  <li><a href="../Conteudos/paginaConteudos/conteudosWebsiteCursos.php">
    Cursos</a></li>
  <li><a href="../About/about.php">About</a></li>
  <li><a href="../PaginaPrincipal/PaginaPrincipal.php">Home</a></li>
</ul>

<div class="cont">
  <form action="../processUpdateUserInfo.php" method="POST">
    <label><b>Nova Pass:</b></label>
    <input type="password" placeholder="password" name="password" required>
    <br>
    <button type="submit">Alterar</button>
  </form>
</div>
```

Listing 18: updateUserInfo.php

No final desta implementação, é criado um script para processar os dados. Foi então criado o script processUpdateUserInfo.php. Para fazer a alteração da password na base de dados, é criada uma query, mas antes de criar a query propriamente dita, é necessário verificar o valor do form, usando o método POST do php. De seguida é criado a query para fazer a alteração da palavra pass, para isso são usadas as funções do php UPDATE, que permite ler a base de dados, a função SET que alterar esse valor, e WHERE, para detetar na base de dados a coluna. Neste caso, a coluna correspondente ao id, é igualada ao método do php SESSION, pois é necessário verificar a sessão e quando os dados são processados. No final do script, é usada a função header para navegar para a próxima mensagem, e confirmar a submissão dos novos dados.

```
<?php

if (!isset($_SESSION)) {session_start();}

require_once( "../Lib/lib.php" );
require_once( "../Lib/db.php" );
```

```
dbConnect(ConfigFile);
$dataBaseName = $GLOBALS['configDataBase']->db;
mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );

// PASSWORD ERROR
$password1 = $_POST['password'];
$password2 = $_POST['password2'];
$passError = "";
if($password1 != $password2){
    $passError = "" . '<p>' . "Passwords_n o_correspondem" . '</p>';
}

if (isset($_POST['password'])) {
    $queryPassword = "UPDATE_`auth-basic`
    SET_`password`=_". $_POST['password'] .
    "_WHERE_`id`=_". $_SESSION['passID'] . """;
    $queryResultPassword = mysqli_query
    ( $GLOBALS['ligacao'], $queryPassword );
}

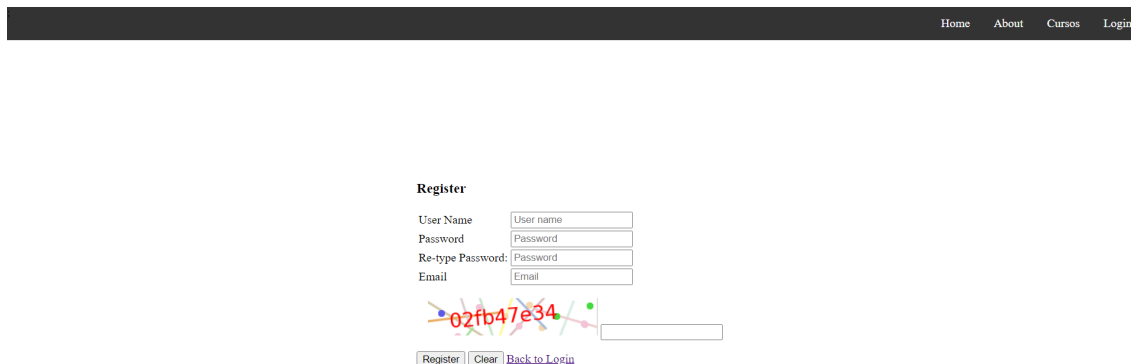
$serverName = "localhost";
$serverPort = 80;
$name = webAppName();
$baseUrl = "http://" . $serverName . ":" . $serverPort;
$baseNextUrl = $baseUrl . $name;
$nextUrl = "messageNewPasswordChanged.php";
header("Location:_ " . $baseNextUrl . $nextUrl);
?>
```

Listing 19: processUpdateUserInfo.php

8.5 Registo

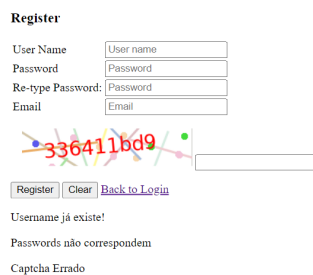
Design da Página

A página de registo tem o seguinte aspeto:



The screenshot shows a registration form titled "Register". It includes four input fields: "User Name", "Password", "Re-type Password", and "Email". Below the "Email" field is a CAPTCHA image displaying the alphanumeric string "02fb47e34". At the bottom of the form are three buttons: "Register", "Clear", and "Back to Login". A dark navigation bar at the top right contains links for "Home", "About", "Cursos", and "Login".

Figura 28: Design da página de login



This screenshot shows the same registration form as Figure 28, but with error messages displayed below the CAPTCHA. The CAPTCHA string is "336411bd9". The error messages are: "Username já existe!", "Passwords não correspondem", and "Captcha Errado". The "Register", "Clear", and "Back to Login" buttons remain at the bottom of the form.

Figura 29: Design da página de login após incorreta introdução de dados

Funcionalidades

Na página de registo é permitido a quem acede à pagina que se registe.

- Efetuar registo no website;
- Transitar para a página de Login, Home, About e Cursos;

Para se registar, o utilizador tem de introduzir um username, uma password e um email. O username e o email têm de ser únicos. As passwords têm de ser iguais e o captcha tem de corresponder com a imagem apresentada.

Implementação

Na implementação do ficheiro formRegister.php é criada a estrutura html para que o utilizador insira os dados de registo e todos os respetivos elementos relativos à navegação entre páginas (navbar, button, etc.). O botão Register irá fazer submit ao form que contém toda a informação introduzida pelo utilizador de forma a que seja possível averiguar se os dados estão corretos.

```
<?php
require_once( "../Lib/lib.php" );

if (!isset($_SESSION)) {
    session_start();
}

$errorMsg = "";

if(isset($_SESSION['loginMsg'])){
    $errorMsg = $_SESSION['loginMsg'];
    $_SESSION['loginMsg'] = '';
}

$serverName = $_SERVER['SERVER_NAME'];
#$serverName = "localhost";

$serverPortSSL = 443;
$serverPort = 80;

$name = webAppName();

$nextUrl = "https://" . $serverName . ":" . $serverPortSSL . $name
. "processFormLogin.php";

$text = "";
```

```

$link = "";

if (isset($_SESSION['username']) ) {
    $text = "Logout";
    $link = "../Login/logout.php";
}
else{
    $text = "Login";
    $link = "../Login/processFormLogin.php";
}
?>

<head>
    <link rel="stylesheet" type="text/css" href="CSS/login.css">
    <title>Login</title>

</head>

<link rel="stylesheet" href="cssMain/mainPage.css">

<ul>
<?php echo "<li><a href='\". $link.\"'><span_class='nottranslate'>
\". $text.\"</span></a></li>" ?>
    <li><a href="../Conteudos/paginaConteudos/contedos
_____WebsiteCursos.php">
    Cursos</a></li>
    <li><a href="../About/about.php"><span class='nottranslate'>
About</span></a></li>
    <li><a href="../PaginaPrincipal/PaginaPrincipal.php">
    <span class='nottranslate'>Home</span></a></li>
    <li class="li" <a id="translate">
    <div id="google_translate_element"></div></a></li>
</ul>

<div class="cont">
    <h3>Login</h3>
    <form action="<?php_echo_ $nextUrl_?>" method="POST">
        <table>
            <tr>
                <td>User Name</td>

```

```
        <td><input
            type="text"
            name="username"
            placeholder="User_name"></td>
    </tr>
    <tr>
        <td>Password</td>
        <td><input
            type="password"
            name="password"
            placeholder="Password"></td>
    </tr>
</table>

<input type="submit" value="Login"> <input type="reset"
value="Clear">
<a href="../Register/formRegister.php">Register</a>
<p><?php echo $errorMsg?></p>
</form>
</div>

<script type="text/javascript">
    function googleTranslateElementInit() {
        new google.translate.TranslateElement({pageLanguage: 'pt',
            includedLanguages: 'en,es,fr,pt',
            layout: google.translate.TranslateElement.
                InlineLayout.SIMPLE},
            'google_translate_element');
    }
</script>
<script type="text/javascript"
src="//translate.google.com/translate_a/element.js?
cb=googleTranslateElementInit"></script>
```

Listing 20: formRegister.php

O captcha é obtido através do script captcha.php. Esse script irá gerar uma string aleatória e irá colocá-la sobre a forma de uma imagem.

```
<?php

// We start a session to access
// the captcha externally!
```

```
session_start();

// Generate a random number
// from 1000-9999
$captcha = @substr(md5(time()), 0, 9);

// The captcha will be stored
// for the session
$_SESSION["captcha"] = $captcha;

$filename = "images/fundocaptch.png";

// Generate a 50x24 standard captcha image
$im = imagecreatefrompng($filename);

// Blue color
$bg = imagecolorallocate($im, 255, 0, 0);

$fontName = "vera.ttf";
$rootDir = realpath($_SERVER["DOCUMENT_ROOT"]);
$fontsDirectory = "./fonts/ttf/";

$fontCaptcha = $rootDir."/../php/extras/fonts/ttf/Vera.ttf";

imagettftext($im, 20, 5, 30, 40, $bg, $fontCaptcha, $captcha);

// Print the captcha text in the image
// with random position & size

// VERY IMPORTANT: Prevent any Browser Cache!!
header("Cache-Control:_no-store,
no-cache,_must-revalidate");

// The PHP-file will be rendered as image
header('Content-type:_image/png');

ob_clean();

// Finally output the captcha as
// PNG image the browser
imagepng($im);
```

```
// Free memory
imagedestroy($im);

?>
```

Listing 21: captcha.php

Após se submeter o formulário no formRegister.php, é necessário avaliar se todos os dados são válidos. Para tal verifica-se se na base de dados se já existe ou não o username introduzido, se as passwords correspondem uma com a outra e se o email é único. Depois confirma-se se o que o utilizador introduziu no captcha corresponde com o valor da mesma, bastando para tal comparar com o que está armazenado na session. Se um ou mais destes dados estiverem errados, será construída uma string contendo uma descrição do que está errado e o utilizador será mantido na página de registo. e ser-lhe-á apresentada a mensagem de erro. Caso tudo esteja correto, será encaminhado até à página de login e será introduzido uma nova entrada na tabela de registos. No entanto, será necessário confirmar o email enviado de forma a obter uma conta válida.

```
<?php
require_once( "../Lib/lib.php" );
require_once( "../Lib/db.php" );

session_start();
dbConnect(ConfigFile);
$dataBaseName = $GLOBALS['configDataBase']->db;
$serverName = $_SERVER[ 'SERVER_NAME' ];
#$serverName = "localhost";

$serverPort = 80;

$name = webAppName();

$baseUrl = "http://" . $serverName . ":" . $serverPort;

$baseNextUrl = $baseUrl . $name;
mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );

$errorMsg = "";
$nextUrl = "";

// EMAIL ERROR
```



```
$query = "SELECT `email` FROM `$dataBaseName`.`auth-basic`";
$mailError = "";
$queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
$mail = $_POST['email'];
if ( $queryResult ) {
    while ( $registro = mysqli_fetch_array( $queryResult ) ){
        $email = $registro['email'];
        if( $email == $_POST['email'] ){
            $mailError = "" . '<p>' . "Email_j _registrado!_Fa a_Login"
                . '</p>';
        }
    }
}

//USER ERROR
$query = "SELECT `name` FROM `$dataBaseName`.`auth-basic`";
$usernameError = "";
$queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
$user = $_POST['username'];
if ( $queryResult ) {
    while ( $registro = mysqli_fetch_array( $queryResult ) ){
        $username = $registro['name'];
        if( $username == $_POST['username'] ){
            $usernameError = "Username_j _existe!";
        }
    }
}

// PASSWORD ERROR
$password1 = $_POST['password1'];
$password2 = $_POST['password2'];
$passError = "";
if( $password1 != $password2 ){
    $passError = "" . '<p>' . "Passwords_n o_correspondem" . '</p>';
}

// CAPTCHA ERROR
$captcha = $_POST['input_captcha'];
$captchaError = '';
if( $captcha != $_SESSION['captcha'] ){
    $captchaError = "" . '<p>' . "Captcha_Errado" . '</p>';
}
```

```
// CRIAR MENSAGEM DE ERRO NA SESSION
$errorMsg = $usernameError . $mailError . $passError . $captchaError;
$_SESSION['errorMsg'] = $errorMsg;

if(strlen($passError)>0 || strlen($usernameError) > 0 ||
strlen($mailError)>0 ||
strlen($captchaError) > 0){
    $nextUrl = 'formRegister.php';
    header("Location:_" . $baseNextUrl . $nextUrl);
} else{
    $query = "INSERT INTO `auth-basic`(`name`,`password`,`email`,`valid`)
VALUES('".$user."','".$password1."','".$mail."','".$0')";
    mysqli_query($GLOBALS['ligacao'], $query);
    /*****
    $baseNextUrl = $baseUrl . 'Login/' ;
    $nextUrl = 'formLogin.php';
    *****/
    $query = "SELECT `id` FROM `".$dataBaseName`.`auth-basic`
WHERE `name`='".$user.'";
    $idCriado = 0;
    $queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
    if ($queryResult) {
        while ($registro = mysqli_fetch_array( $queryResult )){
            $idCriado = $registro['id'];
        }
    }
    include './challenge.php';
    $query = "INSERT INTO `auth-challenge`(`id`,`challenge`)
VALUES('".$idCriado."','".$challenge.'");
    mysqli_query($GLOBALS['ligacao'], $query);
    include './confirmationEmail.php';
    dbDisconnect();
}

?>
```

Listing 22: processFormRegister.php

Para confirmar o email é gerado um código aleatório e é adicionado como parâmetro ao url presente no email que irá levar a uma página que tem como objetivo confirmar a conta na base de dados.

```
<?php
require_once( "../Lib/lib.php");
require_once( "../Lib/lib-mail-v2.php" );
require_once( "../Lib/HtmlMimeMail-class.php" );
require_once( "../Lib/db.php" );

$Account = 1;
$ToName = "";
$ToEmail = "iselg13smi@gmail.com";
$Subject = "Validar Conta";
$Message = "Clique no link abaixo para validar a sua conta." .
           "http://localhost/examples-smi/TP2/Register/validateRegister.php?
           =====challenge=" . $_SESSION['$challenge'];
$smtpServer = "smtp.gmail.com";
$mail = new HtmlMimeMail();

/*
 * HTML component of the e-mail
 */
$MessageHTML = <<<EOD
<html>
  <body style="background:url('background.gif')_repeat;">
    <font face="Verdana,_Arial" color="#FF0000">
      $Message
    </font>
  </body>
</html>
EOD;
/*
 * Add the text, html and embedded images.
 */
$mail->add_html( $MessageHTML, $Message);

/*
 * Builds the message.
 */
$mail->build_message();

/*
 * Sends the message.
 */
$result = $mail->send(
    $smtpServer,
```

```
1,  
465,  
"iselg13smi@gmail.com",  
"isel1920",  
$ToName,  
$ToEmail,  
"G13-SMI",  
"iselg13smi@gmail.com",  
$Subject,  
"X-Mailer:_Html_Mime_Mail_Class");  
  
include '../Login/formLogin.php';  
?>
```

Listing 23: confirmationEmail.php

Para gerar o challenge é utilizado o script challenge.php.

```
<?php  
  
$length = 10;  
$challenge = substr(str_shuffle('0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'),  
1,$length);  
$_SESSION['$challenge'] = $challenge;  
?>
```

Listing 24: confirmationEmail.php

Para validar o registo, o url enviado no email redireciona até ao script validateRegister.php. Para tal é alterado na tabela de registos a conta para válida, removido o challenge da tabela de challenges e adicionado um novo duplo (role, id) na tabela de permissões.

```
<?php  
require_once( "../Lib/lib.php" );  
require_once( "../Lib/db.php" );  
  
dbConnect(ConfigFile);  
$dataBaseName = $GLOBALS['configDataBase']->db;  
$serverName = $_SERVER[ 'SERVER_NAME' ];  
#$serverName = "localhost";  
  
$serverPort = 80;
```

```
$name = webAppName();

$baseUrl = "http://" . $serverName . ":" . $serverPort;

$baseNextUrl = $baseUrl . $name;
mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );

$challenge = htmlspecialchars($_GET["challenge"]);
$col = "ID";
$query = "SELECT count(`id`) as '". $col. "' FROM `auth-challenge`
where `challenge`='". $challenge. "' GROUP by `id`";
$queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
$saida = null;
if ($queryResult) {
    while ($registro = mysqli_fetch_array( $queryResult )){
        $saida = $registro['ID'];
    }
}

if(is_null($saida)){
    echo "<h3>LINK Inv lido</h3>";
}

else{
    echo "<h3>Conta validada!</h3>";

    $query = "SELECT `id` FROM `auth-challenge` where `challenge`='".
    $challenge. "'";
    $idMudar = 0;
    $queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
    if ($queryResult) {
        while ($registro = mysqli_fetch_array( $queryResult )){
            $idMudar = $registro['id'];
        }
    }

    $query = "UPDATE `auth-basic` SET valid=1 where id='". $idMudar. "'";
    $queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
}
```

```
$query = "DELETE_FROM_`auth-challenge`_WHERE_id='". $idMudar. "'";
$queryResult = mysqli_query( $GLOBALS['ligacao'], $query );

$query = "INSERT INTO_`auth-permissions`(`role`,`id`)_VALUES(3,_'".
$idMudar. "')";
mysqli_query($GLOBALS['ligacao'], $query);

//ADICIONAR LINK PARA PAG LOGIN
}
//if($challenge)
dbDisconnect();
?>

<head>
  <title>Validar Conta</title>
</head>
```

Listing 25: validateRegister.php

8.6 Conteúdos

Design da Página

Cada curso contém alguns conteúdos dentro dele. O aspecto desses conteúdos é o seguinte:



Figura 30: Design da página de Conteúdo

Funcionalidade

No interior da página de cada curso é permitido ao utilizador:

- Inscrever-se num curso;
- Visualizar o conteúdo (descrição, imagens e vídeos).

Sendo um administrador é possível ter acesso a outro tipo de funcionalidades:

- Adicionar um novo conteúdo;
- Adicionar uma nova imagem a um conteúdo já existente;
- Adicionar um novo vídeo a um conteúdo já existente;
- Apagar um conteúdo existente;
- Alterar as permissões do utilizador.

Inserir Imagem

Conteúdo:

Introduzir Conteúdo

Imagem

Escolher ficheiro

Nenhum ficheiro seleccionado

Criar

Fechar

Inserir Video

Conteúdo:

Introduzir Conteúdo

Video

Escolher ficheiro

Nenhum ficheiro seleccionado

Criar

Fechar

Criar Secção

Título:

Introduzir Título do Conteúdo

Descricao

Introduzir Descricao do Conteúdo

Imagem

Escolher ficheiro

Nenhum ficheiro seleccionado

Video

Escolher ficheiro

Nenhum ficheiro seleccionado

Criar

Fechar

Figura 31: Adicionar novo Conteúdo, Imagem, Video

Implementação

Na implementação do ficheiro `course_page.php` é feita uma ligação à base de dados para que se possam fazer várias queries, que permitem ir buscar informação à base de dados, é dessa maneira que vamos buscar título de cada curso, conteúdo, é assim que vamos buscar as imagens e os vídeos que aparecem em cada conteúdo. É feita uma query para que se possa obter o role de cada utilizador, depois é feito um if para verificar se o utilizador é administrador ou simpatizante, pois ambos têm acesso a barra de navegação que permite adicionar conteúdos, imagem, vídeos, apagar conteúdos e alterar as permissões. Se forem utilizador ou convidados, apenas têm acesso a visualizar conteúdos, imagens e vídeos, não têm outro tipo de permissões.

```
<?php

require_once( ".../.../Lib/lib.php" );
require_once( ".../.../Lib/db.php" );

if (!isset($_SESSION)) {
    session_start();
}

$_SESSION['locationAfterAdd'] = "course_page.php";
dbConnect(ConfigFile);
$dataBaseName = $GLOBALS['configDataBase']->db;
mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );
$query = "SELECT `role` FROM `{$dataBaseName}`
        `auth-permissions` where `id`=$_" . $_SESSION['id'];

$userRole = 4;
$queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
if ($queryResult) {
    while ($registro = mysqli_fetch_array( $queryResult )){
        $userRole = $registro['role'];
    }
}

if($userRole <= 2){
    echo '<div_class="sidenav">';
    echo '<a onclick="novaSeccao()">-Nova_Sec o </a>';
    echo '<a onclick="novaImagemNav()">-Nova_Imagem</a>';
    echo '<a onclick="novoVideoNav()">-Novo_Video</a>';
    echo '<a onclick="deleteForm()">-Apagar_Conte do</a>';
    echo '<a href_=".../.../Dashboard/'
```

```

.....dashboard.php">-Alterar_Role_Utilizador</a>';
    echo '</div>';
}

    $query = "SELECT `nome_curso` FROM ` $dataBaseName `
.....`curso` where id_curso=" . $_SESSION['idC'];
    $queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
    if ( $queryResult ) {
        while ( $registro = mysqli_fetch_array( $queryResult ) ){
            $nomeCurso = $registro['nome_curso'];
            echo "<title>Curso_de_". $nomeCurso . "</title>";
            echo "<h1_id='title'>". $nomeCurso . "</h1>";

        }
    }

    echo ' <button_id="myBtn">Inscrever_em_Curso</button>';

    $query = "SELECT * FROM ` $dataBaseName `
.....`curso_conteudo` where id_curso=" . $_SESSION['idC'];
    $queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
    if ( $queryResult ) {
        while ( $registro = mysqli_fetch_array( $queryResult ) ){
            $id_cont = $registro['id_conteudo'];
            $query2 = "SELECT * FROM ` $dataBaseName `
.....`conteudo` where id_conteudo=" . $id_cont;
            $queryResult2 = mysqli_query( $GLOBALS['ligacao'], $query2 );
            if ( $queryResult2 ) {
                while ( $registro2 = mysqli_fetch_array( $queryResult2 ) ){
                    $titulo = $registro2['titulo'];
                    $descricao = $registro2['descricao'];
                    $idConteudo = $registro2['id_conteudo'];

                    echo "<h2_class='titulo'>    ". $titulo . "</h2>";

                    $queryImage = "SELECT * FROM ` $dataBaseName `
.....`images` where id_conteudo=" . $idConteudo;
                    $queryResultImage= mysqli_query
                        ( $GLOBALS['ligacao'], $queryImage );
                    if ( $queryResultImage ) {
                        while ( $registroImage =
                            mysqli_fetch_array( $queryResultImage ) ){
                            $image = $registroImage['img'];

```



```

    $text = "Logout";
    $link = "../../Login/logout.php";
}
else{
    $text = "Login";
    $link = "../../Login/processFormLogin.php";
}
dbDisconnect();
?>

<link rel="stylesheet" href="cssMain/sideMenu.css">
<link rel="stylesheet" href="cssMain/mainPage.css">

<ul>
    <?php echo "<li><a href='". $link. "'>
    <span class='nottranslate'>". $text. "</span></a></li>" ?>
    <li><a href="...../About/about.php">About</a></li>
    <li><a href=".../paginaConteudos/conteudosWebsiteCursos.php">
    <span class='nottranslate'>Cursos</span></a></li>
    <li><a href="...../PaginaPrincipal/PaginaPrincipal.php">
    <span class="nottranslate">Home</span></a></li>
    <li class="li" <a id="translate">
    <div id="google_translate_element"></div></a></li>
</ul>

<div class="form-popup" id="myForm">
    <form action="./newContent.php" class="form-container"
        enctype="multipart/form-data"
        method="POST"
        name="FormUpload">
    <h1>Criar Sec  o </h1>

    <label><b>Titulo:</b></label>
    <input type="text" placeholder=
    "Introduzir_Titulo_do_Conte do" name="titulo" >

    <label><b>Descricao</b></label>
    <input type="text" placeholder=
    "Introduzir_Descricao_do_Conte do" name="descricao" >

    <label><b>Imagem</b></label>
    <input type="file" size="64" placeholder=
    "Introduzir_Imagem_do_Conte do" name="imageFile" >

```

```
<label><b>Video</b></label>
<input type="file" size="64" placeholder=
"Introduzir_Video_do_Conte do" name="videoFile" >

<button type="submit" class="btn">Criar</button>
<button type="button" class="btn_cancel"
onclick="closeForm()">Fechar</button>
</form>
</div>

<div class="form-popup" id="myFormImage">
  <form action="./newImage.php" class="form-container"
    enctype="multipart/form-data"
    method="POST"
    name="FormUploadImage">
    <h1>Inserir Imagem</h1>

    <label><b>Conteudo:</b></label>
    <input id="conteudo" type="text"
    placeholder="Introduzir_Conte do" name="titulo" required>

    <label><b>Imagem</b></label>
    <input id="image" type="file" size="64"
    placeholder="Introduzir_Imagem_do_Conte do" name="imageFile" required>

    <button type="submit" class="btn"
    onclick="novaImagem()">Criar</button>
    <button type="button" class="btn_cancel"
    onclick="closeFormImage()">Fechar</button>
  </form>
</div>

<div class="form-popup" id="myFormVideo">
  <form action="./newVideo.php"
  class="form-container"
    enctype="multipart/form-data"
    method="POST"
    name="FormUploadVideo">
    <h1>Inserir Video</h1>
```

```
<label><b>Conteudo:</b></label>
<input id="conteudo" type="text"
placeholder="Introduzir_Conte do" name="titulo" required>

<label><b>Video</b></label>
<input id="video" type="file" size="64"
placeholder="Introduzir_Video_do_Conte do" name="videoFile" required>


<button type="submit" class="btn"
onclick="novoVideo()">Criar</button>
<button type="button" class="btn_cancel"
onclick="closeFormVideo()">Fechar</button>
</form>
</div>

<script>
function novaSeccao() {
    document.getElementById("myForm").style.display = "block";
}

function closeForm() {
    document.getElementById("myForm").style.display = "none";
}

var btn = document.getElementById('myBtn');
btn.addEventListener('click', function() {
    document.location.href = './subscribe.php?idCurso='
    + "<?php_echo_$_SESSION['idC'];?>";
});

function novaImagemNav(){
    document.getElementById("myFormImage").style.display = "block";
}

function novaImagem() {
    var input = document.getElementById("conteudo").value;
    if(input!=null){
        window.location = "./newImage.php?titulo="+input;
    }
}
```

```
function novoVideoNav(){
    document.getElementById("myFormVideo").style.display = "block";
}

function novoVideo() {
    var input = document.getElementById("conteudo").value;
    console.log(input);
    window.location = "./newVideo.php?titulo="+input;
}

function closeFormVideo() {
    document.getElementById("myFormVideo").style.display = "none";
}

function closeFormImage() {
    document.getElementById("myFormImage").style.display = "none";
}

function deleteForm() {
    var input = prompt("T tulo_do_conte do_a_apagar", "");
    if (input != null) {
        window.location="./deleteConteudo.php?deleteConteudo="+input;
    }
}

function googleTranslateElementInit() {
    new google.translate.TranslateElement(
        {pageLanguage: 'pt', includedLanguages: 'en,es,fr,pt',
         layout: google.translate.TranslateElement.InlineLayout.SIMPLE}
        , 'google_translate_element');
}
</script>
<script type="text/javascript"
src="//translate.google.com/translate_a/element.js?
cb=googleTranslateElementInit"></script>
```

Listing 26: course_page.php

A imagem depois de ser feita a query no course_page.php é passada para uma script getThumbnailConteudos.php. Nesse script é criada uma pasta no diretoria "C:/temp/upload/" para que as imagens sejam lá armazenadas.

<?php

```
require_once( "../Lib/lib.php" );
require_once( "../Lib/db.php" );
dbConnect(ConfigFile);
$dataBaseName = $GLOBALS['configDataBase']->db;
mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );

$img = $_GET['img'];

if (!file_exists('C:/temp/upload/')) {
    mkdir('C:/temp/upload/', 0777, true);
}

$filename = "C:/Temp/upload/" . $img;

// Generate a 50x24 standard captcha image
$img = imagecreatefrompng($filename);

// The PHP-file will be rendered as image
header('Content-type: image/png');

ob_clean();

// Finally output the captcha as
// PNG image the browser
imagepng($img);

// Free memory
imagedestroy($img);

dbDisconnect();
?>
```

Listing 27: getThumbnailConteudos.php

Um video é apresentado depois de ser feita a query no course_page.php é passado para uma script getVideo.php. Nesse script é criada uma pasta no diretoria "C:/temp/upload/", caso ainda não exista, para que os vídeos sejam lá armazenadas.

```
<?php
require_once( "../Lib/lib.php" );
require_once( "../Lib/db.php" );
dbConnect(ConfigFile);
```



```
$dataBaseName = $GLOBALS['configDataBase']->db;
mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );

$video = $_GET['video'];

if (!file_exists('C:/temp/upload/')) {
    mkdir('C:/temp/upload/', 0777, true);
}

$filename = "C:/Temp/upload/" . $video;
$mimeType = 'video';
$typeFileName = 'mp4';

$fileHandler = fopen($filename, 'rb');

/*echo ("Content-Type: $mimeType/$typeFileName");
echo ("Content-Length: " . filesize($filename));

echo ( "Content-Transfer-Encoding: Binary" );
echo ( "Content-disposition: attachment; filename=\"" . $filename . "\"

header("Content-Type:_" . $mimeType . "/" . $typeFileName);
header("Content-Length:_" . filesize($filename));

header( "Content-Transfer-Encoding:_Binary" );
header( "Content-disposition:_attachment;_filename=\"" . basename($fil

fpassthru($fileHandler);
fclose($fileHandler);
```

?>

Listing 28: getVideo.php

Para que o administrador consiga adicionar novos conteúdos, novas imagens, novos vídeos, apagar conteúdos ou alterar as permissões de cada utilizador, é criado uma estrutura html, onde basta que o administrador introduza um título, uma descrição, uma imagem e um vídeo introduzidos. São feitas várias queries com INSERTS para que todos os elementos introduzidos sejam adicionados à base de dados, caso algum utilizador esteja registado nesse curso é enviado um e-mail a notificar que foram adicionados novos conteúdos a esse curso. Foi feita uma query que seleccionava o id máximo dos ids de conteúdos para que quando fosse adicionado um conteúdo novo, fosse colocado na posição a seguir à última.

```
<?php

if (!isset($_SESSION)) {
    session_start();
}

require_once( "../Lib/lib.php");
require_once( "../Lib/lib-mail-v2.php" );
require_once( "../Lib/HtmlMimeMail-class.php" );
require_once( "../Lib/db.php" );

dbConnect(ConfigFile);
$dataBaseName = $GLOBALS['configDataBase']->db;
mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );

if (isset($_POST['titulo'])
    && isset($_POST['descricao'])) {

    $query = "INSERT INTO `conteudo`(`titulo`,`descricao`)
        . "VALUES('".$_POST['titulo']. "','".$_POST['descricao']. "')";
    mysqli_query($GLOBALS['ligacao'], $query);

    $query = "SELECT MAX(`id_conteudo`) AS
    maxID FROM `".$dataBaseName`.`conteudo`";
    $maxID = "";
    $queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
    if ($queryResult) {
        while ($registo = mysqli_fetch_array( $queryResult )){
            $maxID = $registo['maxID'];
        }
    }
}
```

```
$query = "INSERT INTO `curso_conteudo`(`id_curso`,`id_conteudo`)"
        . "VALUES('".$_SESSION['idC'].",".$maxID."'");
mysqli_query($GLOBALS['ligacao'], $query);

$query = "INSERT INTO `images`(`id_conteudo`,`img`)"
        . "VALUES('".$maxID.",'".$_FILES['imageFile']['name']."'");
mysqli_query($GLOBALS['ligacao'], $query);

//echo $query . "\n";

$query = "INSERT INTO `video`(`id_conteudo`,`video`)"
        . "VALUES('".$maxID.",'
        . " '".$_FILES['videoFile']['name']."'");
mysqli_query($GLOBALS['ligacao'], $query);

//echo $query . "\n";
$query = "SELECT * FROM `".$dataBaseName`.`inscricao`";
$queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
if ($queryResult) {
    while ($registro = mysqli_fetch_array( $queryResult )){
        $id_user = $registro['id'];
        $idCurso = $registro['idCurso'];

        $query2 = "SELECT * FROM `".$dataBaseName`.`
        . " `auth-basic`_where_`id`=_ " . $id_user;
        $queryResult2 = mysqli_query( $GLOBALS['ligacao'], $query2 );
        if ($queryResult2) {
            while ($registro2 = mysqli_fetch_array( $queryResult2 )){
                $nome = $registro2['name'];
                $email = $registro2['email'];

                $curso_nome = "";

                $query3 = "SELECT `nome_curso` FROM `".$dataBaseName`.`
                . " `curso`_where_`id_curso`=_ " . $idCurso;
                $queryResult3 = mysqli_query
                ( $GLOBALS['ligacao'], $query3 );
                if ($queryResult3) {
                    while ($registro3 =
                    mysqli_fetch_array( $queryResult3 )){
                        $curso_nome = $registro3['nome_curso'];
                    }
                }
            }
        }
    }
}
```

```

        $Account = 1;
        $ToName = "";
        $ToEmail = $email;
        $Subject = "E-Learning_-_Novos_Conteudos";
        $Message = "Caro_" . $nome . "_o_curso_de_" .
        $curso_nome . "_recebeu_novos_conte dos .
        _Mantenha-se_a_par!";
        $smtpServer = "smtp.gmail.com";
        $mail = new HtmlMimeMail();
$MessageHTML = <<<EOD
<html>
    <body style="background:_url('background.gif')_repeat;">
        <font face="Verdana,_Arial" color="#FF0000">
            $Message
        </font>
    </body>
</html>
EOD;

        $mail->add_html( $MessageHTML , $Message);
        $mail->build_message();
        $result = $mail->send(
            $smtpServer ,
            1,
            465,
            "iselg13smi@gmail.com",
            "isel1920",
            $ToName ,
            $ToEmail ,
            "G13-SMI" ,
            "iselg13smi@gmail.com",
            $Subject ,
            "X-Mailer:_Html_Mime_Mail_Class");
        }
    }
}

}

}
dbDisconnect();
include_once './imageUpload.php';
include_once './videoUpload.php';

```

```
$serverName = "localhost";
$serverPort = 80;
$name = webAppName();
$baseUrl = "http://" . $serverName . ":" . $serverPort;
$baseNextUrl = $baseUrl . $name;
$nextUrl = $_SESSION['locationAfterAdd'];

echo $baseNextUrl . $nextUrl;

header("Location:_" . $baseNextUrl . $nextUrl);
?>
```

Listing 29: newContent.php

Para se inscrever num curso, basta que dentro de um curso o utilizador carregue no botão "Inscrever em Curso", quando carregado nesse botão, é feita uma análise ao role do utilizador e caso seja do tipo utilizador ou convidado é feita uma query que insere o utilizador nesse curso.

```
<?php

require_once( "../Lib/lib.php" );
require_once( "../Lib/db.php" );

if (!isset($_SESSION)) {
    session_start();
}

dbConnect(ConfigFile);
$dbName = $GLOBALS['configDataBase']->db;
mysqli_select_db( $GLOBALS['ligacao'], $dbName );
$query = "SELECT `role` FROM `{$dbName`.`auth-permissions`}` where `id` = " . $_SESSION['id'];

$userRole = 4;
$queryResult = mysqli_query( $GLOBALS['ligacao'], $query );
if ($queryResult) {
    while ($registro = mysqli_fetch_array( $queryResult )){
        $userRole = $registro['role'];
    }
}

if($userRole == 2 || $userRole == 3){
```

```
        $query = "INSERT INTO `inscricao`(`id`,`idCurso`)
VALUES('".$_SESSION['id']. "','".$_GET['idCurso']. "')";
        mysqli_query($GLOBALS['ligacao'], $query);
        echo 'Foi inscrito neste curso';
    }

    dbDisconnect();

?>
```

Listing 30: subscribe.php

Para que se adicione um nova imagem ou vídeo as scripts são feitas da mesma maneira, a única alteração é a tabela na base de dados que difere o nome. Passando, cada uma delas por duas scripts, imageUpload.php para a newImage.php e videoUpload.php para o newVideo.php. É feita uma ligação à base de dados, depois é feita uma query que dá acesso ao id do conteúdo pretendido consoante o título que foi introduzido na estrutura html anteriormente. Tendo o id do conteúdo, é feito uma query INSERT que permite colocar a imagem ou o vídeo no conteúdo pretendido.

```
<?php

if (!isset($_SESSION)) {
    session_start();
}

require_once( "../Lib/lib.php" );
require_once( "../Lib/db.php" );

dbConnect(ConfigFile);
$dbaseName = $GLOBALS['configDataBase']->db;
mysqli_select_db( $GLOBALS['ligacao'], $dbaseName );

if(isset($_POST['titulo'])){

    $query = "SELECT `id_conteudo` FROM `$dbaseName`
    `conteudo` where `titulo` = " . $_POST['titulo'] . " ";
    $idCImg = "";
    $queryResult = mysqli_query($GLOBALS['ligacao'], $query);

    if($queryResult){
        while($registro = mysqli_fetch_array($queryResult)){
            $idCImg = $registro['id_conteudo'];
        }
    }
}
```

```

        }
    }
    $queryImage = "INSERT INTO `images`(`id_conteudo`,`img`)"
        . "VALUES('" . $idCImg . "',"
        . " " . $_FILES['imageFile']['name'] . "');"
    mysqli_query($GLOBALS['ligacao'], $queryImage);

}
dbDisconnect();
include_once './imageUpload.php';

//
$serverName = "localhost";
$serverPort = 80;
$name = webAppName();
$baseUrl = "http://" . $serverName . ":" . $serverPort;
$baseNextUrl = $baseUrl . $name;
$nextUrl = $_SESSION['locationAfterAdd'];

echo $baseNextUrl . $nextUrl;

header("Location:_" . $baseNextUrl . $nextUrl);
?>

```

Listing 31: newImage.php

```

<?php

// Maximum time allowed for the upload
set_time_limit( 300 );

if ( $_FILES['imageFile']['error'] != 0 ) {
    // Handle the error
    $errorCode = $_FILES['imageFile']['error'];

    echo showUploadFileError($errorCode);

    exit($errorCode);
}

$srcName = $_FILES['imageFile']['name'];

```

```
// Read configurations from data base
$configurations = getConfiguration();
$dstDir = $configurations['destination'];

// Destination for the uploaded file
$src = $_FILES['imageFile']['tmp_name'];
$dst = $dstDir . DIRECTORY_SEPARATOR . $srcName;

$copyResult = copy($src, $dst);

if ( $copyResult === false ) {
    echo "Could_not_write\_\"$src\"_to\_\"$dst\".\n<br>";
    exit();
}

unlink($src);

echo "File_uploaded_with_sucess.\n<br>";

$fileInfo = finfo_open(FILEINFO_MIME);

$fileInfoData = finfo_file($fileInfo, $dst);
print_r($fileInfoData);

$fileTypeComponents = explode(";", $fileInfoData);

$mimeTypeFileUploaded = explode("/", $fileTypeComponents[0]);
$mimeTypeFileUploaded[0];
$typeFileName = $mimeTypeFileUploaded[1];

$thumbsDir = $dstDir . DIRECTORY_SEPARATOR . "thumbs";
$pathParts = pathinfo($dst);

$lat = "";
$lon = "";

$width = $configurations['thumbWidth'];
$height = $configurations['thumbHeight'];

echo "File_is_of_type_$mimeTypeFileUploaded<br>\n";

$imageFileNameAux = null;
```



```
$imageMimeFileName = null;
$imageTypeFileName = null;

$thumbFileNameAux = null;
$thumbMimeFileName = null;
$thumbTypeFileName = null;

switch ($mimeFileName) {
    case "image":
        $exif = @exif_read_data($dst, 'IFD0', true);

        if ($exif === false) {
            echo "No_exif_header_data_found.<br>\n";
        }
        else {
            //Just for Debug - Begin

            echo "<pre>";
            foreach ($exif as $key => $section) {
                foreach ($section as $name => $val) {
                    echo "$key.$name:_$val<br>\n";
                }
            }
            echo "</pre>";

            //Just for Debug - End

            $gps = $exif['GPS'];
            if ( $gps!=NULL ) {
                $latitudeAux = $gps['GPSLatitude'];
                $latitudeRef = $gps['GPSLatitudeRef'];
                $longitudeAux = $gps['GPSLongitude'];
                $longitudeRef = $gps['GPSLongitudeRef'];

                if ( ( $latitudeAux!=NULL ) && ( $longitudeAux!=NULL ) ) {
                    $lat = getCoordAsString($latitudeAux, $latitudeRef);
                    $lon = getCoordAsString($longitudeAux, $longitudeRef);
                    echo "File_latitude:_$lat<br>\n";
                    echo "File_longitued:_$lon<br>\n";
                }
                else {
                    echo "File_does_have_GPS_coordenates<br>\n";
                }
            }
        }
    }
}
```

```

    }
    else {
        echo "File_does_have_GPS_coordinates<br>\n";
    }
}

$imageFileNameAux = $dst;
$imageMimeFileName = "image";
$imageTypeFileName = $typeFileName;

$thumbFileNameAux = $thumbsDir . ~
DIRECTORY_SEPARATOR . $pathParts['filename'] . "." . $typeFileName;
$thumbMimeFileName = "image";
$thumbTypeFileName = $typeFileName;

break;

case "video":
    $size = "$width" . "x" . "$height";

    $imageFileNameAux = $thumbsDir
    . DIRECTORY_SEPARATOR . $pathParts['filename'] . "-Large.jpg";
    $imageMimeFileName = "image";
    $imageTypeFileName = "jpeg";
    echo "Generating_video_1st_image...<br>\n";

    // -itsoffset -1 -> "avan a" o filme 1 segundo
    // -i $dst -> input file
    // -vcodec mjpeg -> codec do tipo mjpeg
    // -vframes 1 -> obter uma frame
    // -s 640x480 -> dimens o do output
    $cmdFirstImage = "_$ffmpegBinary_-itsoffset_-1
    -i_$dst_-vcodec_mjpeg_-vframes_1_-an_-f
    rawvideo_-s_640x480_$imageFileNameAux";

    echo "$cmdFirstImage<br>\n";
    system($cmdFirstImage, $status);
    echo "Status_from_the_generation
    of_video_1st_image:_$status.<br>\n";

    $thumbFileNameAux = $thumbsDir
    . DIRECTORY_SEPARATOR . $pathParts['filename'] . ".jpg";
    $thumbMimeFileName = "image";

```

```

        $thumbTypeFileName = "jpeg";
        echo "Generating_video_thumb...<br>\n";

        $cmdVideoThumb = "ffmpegBinary_-itsoffset
        -5_-i_$dst_-vcodec_mjpeg_-vframes_1_-an
        -f_rawvideo_-s_$size_$thumbFileNameAux";
        echo "$cmdVideoThumb<br>\n";
        system($cmdVideoThumb, $status);
        echo "Status_from_the_generation_of_video_thumb:_$status.<br>\n";
        break;

    case "audio":
        require_once( "Zend/Media/Id3v2.php" );

        $id3 = new Zend_Media_Id3v2($dst);

        $mimeTypeAudioAPIC = explode("/", $id3->apic->mimeType);
        // $mimeTypeAudioAPIC = $mimeTypeAudioAPIC[0];
        $typeAudioAPIC = $mimeTypeAudioAPIC[1];

        $imageFileNameAux = $thumbsDir
        . DIRECTORY_SEPARATOR . $pathParts['filename']
        . "-Large." . $typeAudioAPIC;
        $imageMimeFileName = "image";
        $imageTypeFileName = $typeAudioAPIC;
        $fdMusicImage = fopen($imageFileNameAux, "wb");
        fwrite($fdMusicImage, $id3->apic->getImageData());
        fclose($fdMusicImage);

        $thumbFileNameAux = $thumbsDir . DIRECTORY_SEPARATOR
        . $pathParts['filename'] . "." . $typeAudioAPIC;
        $thumbMimeFileName = "image";
        $thumbTypeFileName = $typeAudioAPIC;
        $resizeObj = new ImageResize($imageFileNameAux);
        $resizeObj->resizeImage($width, $height, 'crop');
        $resizeObj->saveImage($thumbFileNameAux,
        $typeAudioAPIC, 100);
        $resizeObj->close();
        break;

    default:
        $imageFileNameAux = $dstDir . DIRECTORY_SEPARATOR
        . "default" . DIRECTORY_SEPARATOR . "Unknown-Large.jpg";

```

```
$imageMimeFileName = "image";
$imageTypeFileName = "jpeg";

$thumbFileNameAux = $dstDir . DIRECTORY_SEPARATOR
    . "default" . DIRECTORY_SEPARATOR . "Unknown.jpg";
$thumbMimeFileName = "image";
$thumbTypeFileName = "jpeg";
break;
}

?>
```

Listing 32: imageUpload.php

Para que um conteúdo seja apagado, quando o administrador clica no botão de apagar, aparece uma popup que pede para que o título do conteúdo seja introduzido. Quando esse título é introduzido é feita uma query que dá acesso ao id do conteúdo consoante esse título que foi introduzido, feito isso basta fazer 3 query DELETE, uma para o conteúdo, uma para a imagem e outra para o vídeo.

```
<?php

require_once( "../Lib/lib.php" );
require_once( "../Lib/db.php" );

if (!isset($_SESSION)) {
    session_start();
}

dbConnect(ConfigFile);
$dataBaseName = $GLOBALS['configDataBase']->db;
mysqli_select_db( $GLOBALS['ligacao'], $dataBaseName );

$idConteudo = "";

$query = "SELECT_id_conteudo_FROM_$dataBaseName
    _conteudo_WHERE_titulo='". $_GET["deleteConteudo"]. "' " ;

$queryResult = mysqli_query( $GLOBALS['ligacao'], $query );

if($queryResult){
    while ($registro = mysqli_fetch_array($queryResult)){
```

```
        $idConteudo = $registro['id_conteudo'];
    }
}

$query = "DELETE FROM images WHERE id_conteudo=" . $idConteudo;
mysqli_query( $GLOBALS['ligacao'], $query );

$query = "DELETE FROM video WHERE id_conteudo=" . $idConteudo;
mysqli_query( $GLOBALS['ligacao'], $query );

$query = "DELETE FROM conteudo WHERE titulo='"
    . $_GET["deleteConteudo"] . "'";
mysqli_query( $GLOBALS['ligacao'], $query );

$serverName = "localhost";
$serverPort = 80;
$name = webAppName();
$baseUrl = "http://" . $serverName . ":" . $serverPort;
$baseNextUrl = $baseUrl . $name;
$nextUrl = "course_page.php";

header("Location:_" . $baseNextUrl . $nextUrl);

?>
```

Listing 33: deleteConteudo.php

Bibliography

[1] <https://www.opensourcecms.com/atutor/>

[2] <https://tudosobrecms.wordpress.com/vantagens-e-desvantagens-2/>