

Plataforma de Gestão e Disponibilização de Recursos Educativos

Projeto Final da UC Desenvolvimento de Aplicações WEB

Paulo Barbosa (PG40160)

Janeiro 2021

Resumo

O presente relatório tem como objetivo apresentar uma Plataforma de Gestão e Disponibilização de Recursos Educativos, assim como todas as suas funcionalidades.

Na Introdução são apresentadas as condições nas quais a mesma foi desenvolvida e algumas escolhas conceptuais presentes no modelo final.

Na Estrutura de Ficheiros é explicada a diretoria da aplicação pasta por pasta.

No Funcionamento do Sistema é exibido o sistema em ação.

Por último, é recapitulado o conteúdo do artigo, na Conclusão, bem como apresentado trabalho futuro possível.

Keywords: UM, MiEI, DAW2020, ExpressJS, MongoDB, JavaScript, EJS

1 Introdução

No âmbito da Unidade Curricular Desenvolvimento de Aplicações Web do Mestrado Integrado em Engenharia Informática da Universidade do Minho, foi desenvolvida uma Plataforma de Gestão e Disponibilização de Recursos Educativos. O presente relatório serve como suporte à aplicação desenvolvida. A plataforma em questão poderá ser transferida em <https://github.com/PJBarbosa98/DAW2020> (na pasta *Project*). Antes de explicitar a estrutura do sistema, bem como o seu funcionamento, é importante mencionar algumas características nas quais a sua construção foi baseada.

A plataforma tem como objetivo disponibilizar recursos educativos de vários tipos: livros, artigos, aplicações e trabalhos de alunos. Ao longo deste relatório, os recursos educativos mencionados acima serão designados simplesmente como artigos por uma questão de simplicidade.

Utilizadores interagirão com a plataforma. Para estes serão designados **três níveis de acesso: administrador, consumidor e produtor**. A diferença entre estes é definida pelo que cada um pode ver e interagir com.

O administrador do sistema tem acesso a toda a informação presente na plataforma. Todos os outros podem apenas observar informação que é disponibilizada como sendo de acesso público. Quando um utilizador carrega um artigo, é o seu produtor. Ainda que este seja marcado como privado, terá acesso a este recurso. Por outro lado, qualquer outro utilizador da plataforma (à exceção do administrador), será designado como consumidor deste e não lhe poderá aceder.

Cada utilizador da plataforma poderá adicionar novos recursos. Estes serão classificados por ano, tipo, tema, *tags*, etc. Interação entre utilizadores também é possível através de comentários e avaliações de recursos carregados na plataforma.

2 Estrutura de Ficheiros

É, agora, explicitada a estrutura dos ficheiros referentes à aplicação. Ao navegar até à diretoria da plataforma poderá verificar a seguinte estrutura:

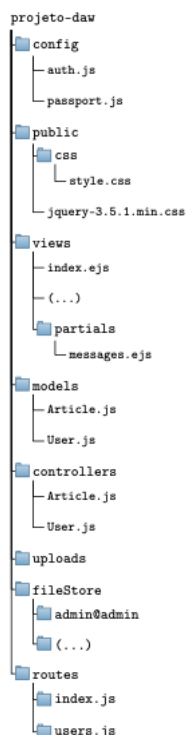


Figura 1: Árvore da diretoria. Nesta imagem, root é a pasta onde a plataforma se encontra no seu sistema.

Na pasta **config** são definidas as estratégias de autenticação de utilizadores.

Em **public** encontram-se os ficheiros estáticos como, por exemplo, *stylesheets* (*bootstrap*) e bibliotecas (*jQuery*).

Os *templates* para cada uma das páginas web apresentada aos utilizadores encontram-se em **views**. Estes foram escritos com recurso à biblioteca **EJS** (mais informação em [1]).

Os modelos (pasta **models**) contêm a estrutura de objetos pertencentes a uma coleção na base de dados. Para cada um destes, existe um controlador respetivo em **controllers**. Em ambas as diretorias, são visíveis dois ficheiros. Um para os utilizadores (**User.js**) e outro para os recursos educativos (**Article.js**).

Uploads é uma zona temporária para os ficheiros carregados para o sistema. Cada um destes será, posteriormente, redirecionado para **FileStore**. Nesta última, os ficheiros são armazenados. É criada uma subpasta para cada utilizador (pelo seu e-mail).

Por último, em **routes**, encontram-se as rotas que um navegador pode aceder (em *index.js* para a zona não autenticada e em *users.js*, caso contrário).

3 Funcionamento do Sistema

A aplicação foi criada em **ExpressJS** ([2]).

O motor de base de dados escolhido foi o **MongoDb** ([4]).

A comunicação entre estes módulos é feita através de uma biblioteca de *JavaScript* designada por **Mongoose** ([3]).

Para iniciar a aplicação, basta correr os seguintes comandos:

- `cd 'Diretoria da Aplicação'`
- `npm install`
- `npm start`

Quando isto tiver sido feito, basta aceder à página `localhost:7710`. Se tudo estiver a funcionar corretamente, verá o seguinte:

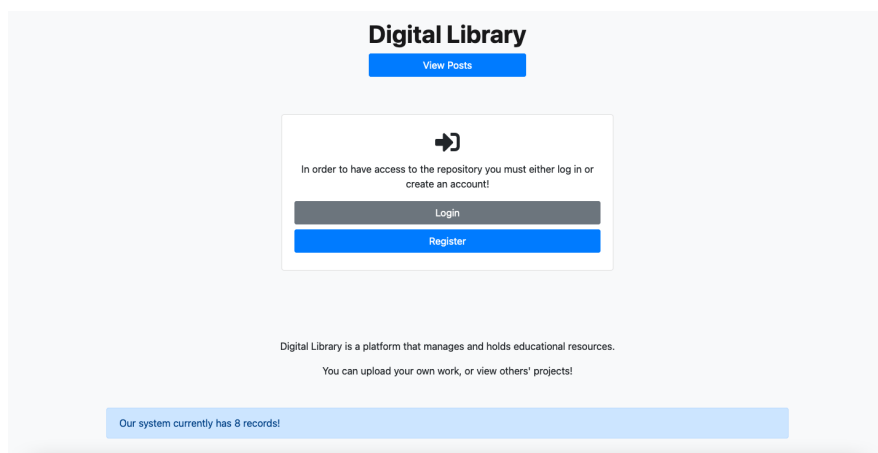
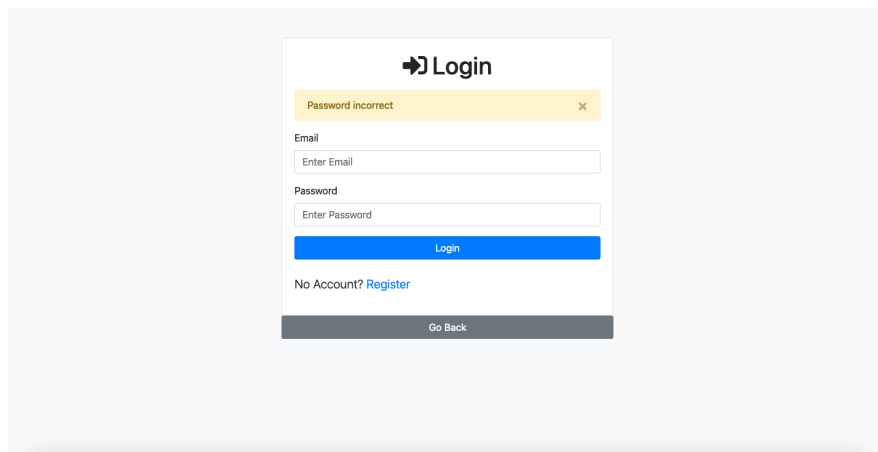


Figura 2: Página principal da aplicação.

É-lhe, agora, possível criar ou fazer *login* na sua conta.

Se algo de errado acontecer em algum destes processos, será notificado com uma mensagem de erro. De seguida, é apresentado um exemplo para quando a palavra-passe inserida é incorreta.

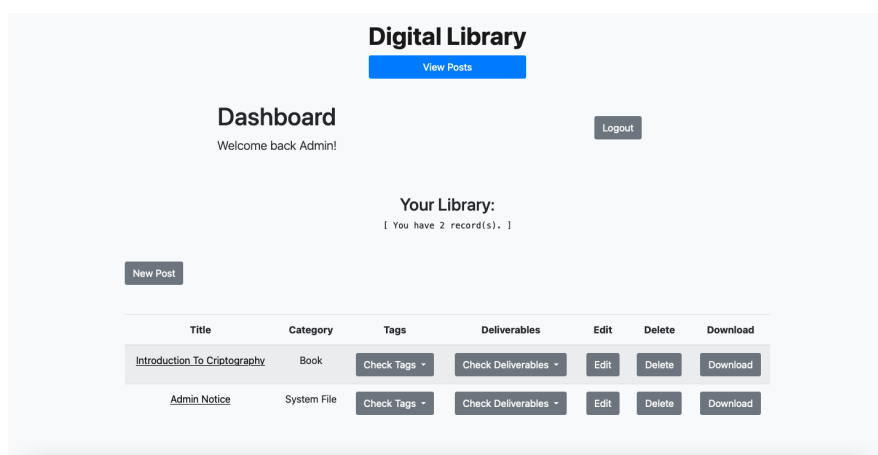


The image shows a login form titled "Login" with a right-pointing arrow icon. At the top, there is a yellow error message box that says "Password incorrect" with a close button (X). Below this, there are two input fields: "Email" with the placeholder text "Enter Email" and "Password" with the placeholder text "Enter Password". A blue "Login" button is positioned below the password field. At the bottom of the form, there is a link "No Account? Register" and a grey "Go Back" button.

Figura 3: Mensagem de erro no login.

Quando um utilizador tiver sido devidamente autenticado, será redirecionado para a *dashboard*. Aqui poderá adicionar, editar, remover e mesmo transferir os seus recursos educativos.

É de notar que vários ficheiros (chamados de *deliverables* na plataforma) podem ser adicionados aquando da adição de um recurso educativo. Adicionalmente, para cada utilizador, cada artigo deverá ter um título diferente (que não pode ser editado).



The image shows an admin dashboard titled "Digital Library" with a "View Posts" button. Below the title, there is a "Dashboard" section with the text "Welcome back Admin!" and a "Logout" button. Underneath, there is a "Your Library:" section with the text "[You have 2 record(s).]". A "New Post" button is located to the left of a table. The table has columns for Title, Category, Tags, Deliverables, Edit, Delete, and Download. The first row shows "Introduction To Cryptography" as the title, "Book" as the category, and "Check Tags" as the tag. The second row shows "Admin Notice" as the title, "System File" as the category, and "Check Tags" as the tag. Each row has buttons for "Check Deliverables", "Edit", "Delete", and "Download".

Title	Category	Tags	Deliverables	Edit	Delete	Download
Introduction To Cryptography	Book	Check Tags	Check Deliverables	Edit	Delete	Download
Admin Notice	System File	Check Tags	Check Deliverables	Edit	Delete	Download

Figura 4: Dashboard (vista pelo administrador).

Se, por sua vez, um utilizador tentar aceder a uma página protegida (tal como a

dashboard) sem se ter autenticado, irá obter a seguinte mensagem:

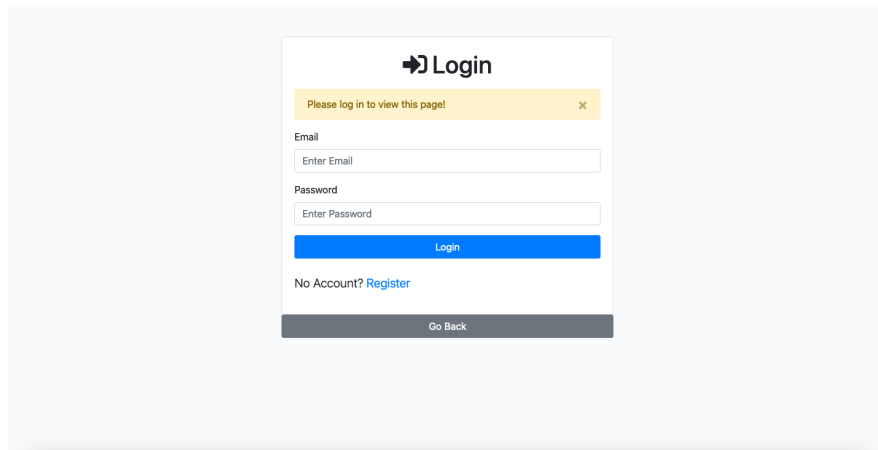
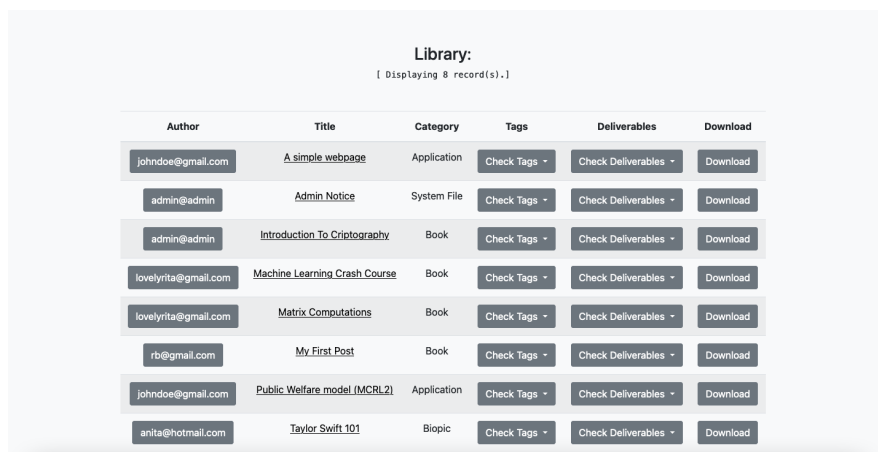


Figura 5: Mensagem de erro apresentada quando o utilizador tenta obter um recurso antes de se autenticar.

Adicionalmente, é possível ainda ver o *feed* da plataforma, isto é, todos os *posts* ao qual o utilizador autenticado tem acesso.

A screenshot of a "Library" page. It shows a table with 8 records. The table has columns for Author, Title, Category, Tags, Deliverables, and Download. Each row represents a resource with its respective details and interactive buttons for tags, deliverables, and download.

Author	Title	Category	Tags	Deliverables	Download
john.doe@gmail.com	A simple webpage	Application	Check Tags -	Check Deliverables -	Download
admin@admin	Admin Notice	System File	Check Tags -	Check Deliverables -	Download
admin@admin	Introduction To Cryptography	Book	Check Tags -	Check Deliverables -	Download
lovelyrita@gmail.com	Machine Learning Crash Course	Book	Check Tags -	Check Deliverables -	Download
lovelyrita@gmail.com	Matrix Computations	Book	Check Tags -	Check Deliverables -	Download
rb@gmail.com	My First Post	Book	Check Tags -	Check Deliverables -	Download
john.doe@gmail.com	Public Welfare model (MCRL2)	Application	Check Tags -	Check Deliverables -	Download
anita@hotmail.com	Taylor Swift 101	Biopic	Check Tags -	Check Deliverables -	Download

Figura 6: Feed da plataforma.

A visualização do perfil dos outros utilizadores está implementada, bem como os seus artigos. Nestas páginas, podem fazer-se comentários de modo a expressar a opinião sobre o recurso em questão.

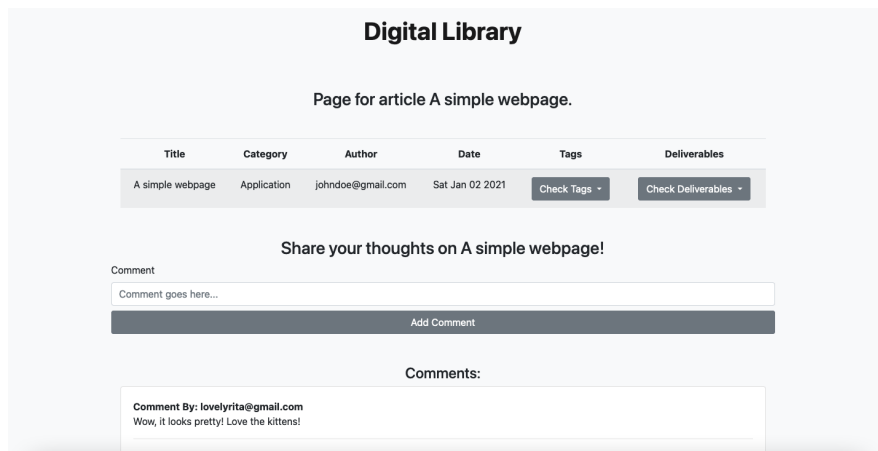


Figura 7: Página individual de um artigo.

4 Conclusão

Após serem testadas todas as funcionalidades da plataforma, é possível verificar que a mesma funciona corretamente, como descrito ao longo deste relatório.

Como trabalho futuro proporia alterar o *design* da plataforma para algo mais apelativo ao utilizador comum. Adicionalmente, poderiam ser adicionadas mais funcionalidades tais como mensagens entre utilizadores e fotos de perfil.

Referências

- [1] Matthew Eernisse. *EJS home page*. <https://ejs.co/>, 2012.
- [2] OpenJS Foundation. *ExpressJS home page*. <https://expressjs.com/>, 2017.
- [3] MIT License. *Mongoose home page*. <https://mongoosejs.com/>, 2013.
- [4] Inc. MongoDB. *MongoDB home page*. <https://www.mongodb.com/>, 2009.