

**Departamento de Engenharia de Electrónica e Telecomunicações e de
Computadores**

Chelas Movie Data Base

48642 : Daniel Martins Cabrita de Sousa (A48642@alunos.isel.pt)

49472 : Guilherme Filipe Cardoso Ferreira (A49472@alunos.isel.pt)

49149 : Diogo Vaz Guerra (A49149@alunos.isel.pt)

Relatório para a Unidade Curricular de Introdução à Programação na Web
da Licenciatura em Engenharia Informática e de Computadores

Professor : Filipe Bastos de Freitas

Resumo

Este projeto foi desenvolvido recorrendo à linguagem de programação *JavaScript* com *Node.js*, *Elasticsearch* e *Handlebars* para criar um site similar ao IMDb (CMDb). O objetivo do projeto foi criar um site de pesquisa e exibição de informações sobre filmes usando a API do IMDb, incluindo detalhes como o nome do filme, descrição, duração, ano, capa, elenco e produção. O *Elasticsearch* foi utilizado como base de dados e o *Handlebars* como motor de template para exibir os dados de forma dinâmica. O projeto foi desenvolvido com desempenho em mente, para garantir uma experiência rápida e fluida para os utilizadores.

Abstract

This project was developed using the *JavaScript* programming language with *Node.js*, *Elasticsearch* and *Handlebars* to create a website similar to IMDb (CMDb). The aim of the project was to create a website for searching and displaying information about movies using the IMDb API, including details such as movie name, description, duration, year, cover, cast and production. *Elasticsearch* was used as the database and *Handlebars* as the template engine to dynamically display the data. The project was developed with performance in mind, to ensure a fast and smooth experience for users.

Índice

Lista de Figuras	ix
Lista de Tabelas	xi
1 Estrutura do Programa	1
1.1 Server	2
1.2 Web-site	2
1.3 Web-api	2
1.4 Services	3
1.5 Imdb data	3
1.6 Elastic data	3
2 ElasticSearch	5
2.1 Users	5
2.2 Groups	6
2.3 Movies	7
3 Documentação	9
3.1 Popular (api/popular)	10
3.2 Search (api/search)	10
3.3 Get movie by Id (api/movie/:movieId)	11

3.4	Signup (api/signup)	11
3.5	List groups (api/groups)	12
3.6	Create group (api/groups)	12
3.7	Get group (api/groups/:groupId)	13
3.8	Update group (api/groups/:groupId)	13
3.9	Remove group (/api/groups/:groupId)	14
3.10	Add movie (api/groups/:groupId/:movieId)	14
3.11	Delete movie (api/groups/:groupId/:movieId)	15
3.12	Get CSS (site.css)	15
3.13	Get Script (scripts:ScriptName)	15
3.14	Get Home page (/)	16
3.15	Get Popular (/popular)	16
3.16	Search movie (/search)	16
3.17	Gets movie info (/movie/:movieId)	17
3.18	Get groups (/groups)	17
3.19	Create group (/groups)	17
3.20	Get group (/groups/:groupId)	18
3.21	Add movie (/groups/:groupId)	18
4	Conclusões	19

Lista de Figuras

1.1	Estrutura do projeto.	1
-----	-------------------------------	---

Lista de Tabelas

3.1	Caminhos do API	9
3.2	Caminhos do WEB-SITE	10

Estrutura do Programa

O projeto foi desenvolvido segundo a estrutura representada pela figura abaixo.

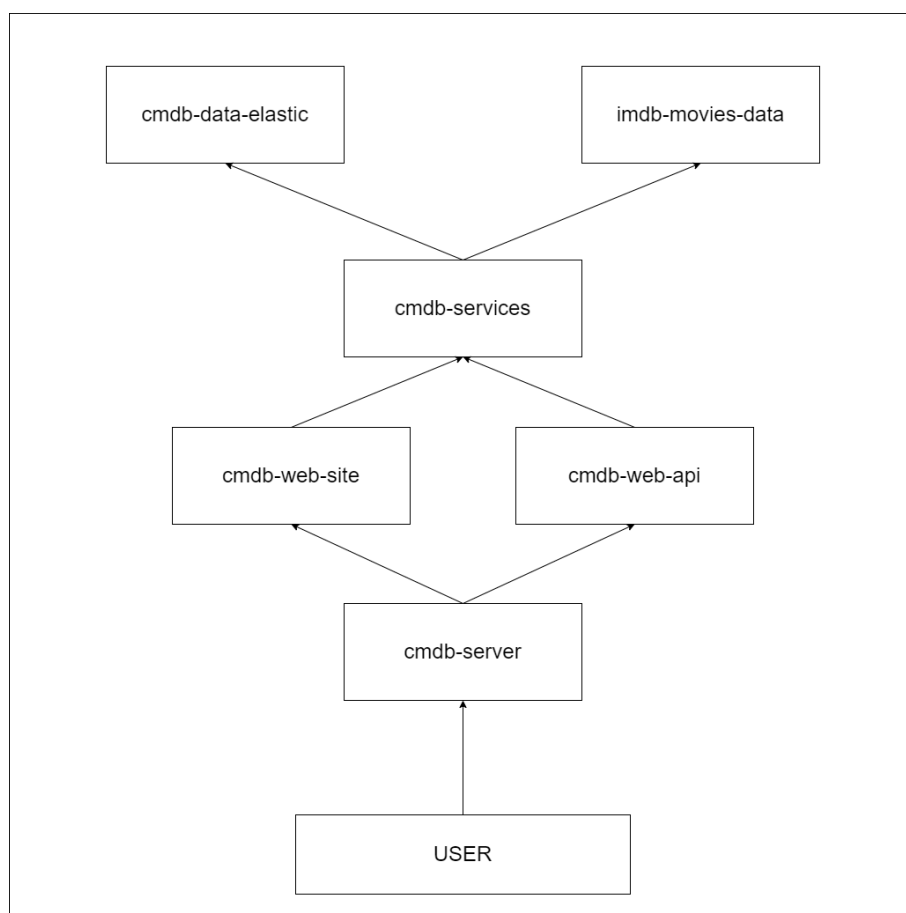


Figura 1.1: Estrutura do projeto.

O programa encontra-se subdividido em 6 partes:

- Server;
- Web-site;
- Web-api;
- Services;
- Imdb data;
- Elastic data.

1.1 Server

Este módulo é responsável por iniciar o servidor e estabelecer a ligação entre o cliente e os serviços internos. Estabelece também a ligação do módulo *services* a cada um dos possíveis caminhos, e os *middlewares* necessários para o normal funcionamento do programa e de outros módulos, tal como estabelecer o caminho usado pelo *HandleBars* para renderizar a interface do utilizador no browser.

É o *endpoint* do programa sendo o único módulo a que o utilizador tem acesso.

1.2 Web-site

Representa a interface gráfica do utilizador. Renderiza apenas a informação necessária e de forma dinâmica e fluida para que o acesso ao utilizador seja o mais simples e direto possível.

O dinamismo é assegurada pelo uso da linguagem de formatação *HandleBars* e a fluidez pelo uso de *CSS*, que permite a modelação das páginas *HTML* de forma a que a apresentação de informação ao utilizador seja a mais simples possível.

1.3 Web-api

O módulo permite que exista comunicação entre o cliente e o servidor através de respostas no formato *JSON*.

Por um lado, este módulo permite que todos os serviços disponíveis sejam utilizados fora de uma interface gráfica, por um cliente automatizado, assim como a própria API do IMDb está a ser utilizada no módulo *imdb-movie-data*.

Por outro lado, permite que a própria interface gráfica execute pedidos ao servidor sem que este retorne informação desnecessária, como na atualização de um grupo e na remoção/inserção de filmes ou grupos na base de dados.

1.4 Services

Este módulo representa a camada de segurança interna ao servidor e estabelece a comunicação entre os diferentes pedidos (api, web-site) e a própria base de dados. Devido à sua responsabilidade, é necessário que este módulo seja o mais redundante possível de forma a que qualquer tipo de pedido seja alvo de uma verificação meticulosa antes de transmitir os dados à memória interna, podendo, em caso de falta de uma verificação ampla, a possibilidade de corrupção da informação existente ou da introdução de informação maliciosa ou errada.

1.5 Imdb data

Permite a comunicação entre os serviços internos e a API do IMDb de forma a requisitar as informações necessárias ao funcionamento do programa, como as informações de filmes, a procura por um determinado título ou a apresentação dos top 250 filmes mais famosos.

1.6 Elastic data

Este é o módulo que serve como base para a manutenção da informação individual de cada utilizador, ou seja, permite a introdução e/ou remoção de grupos e filmes e a atualização das informação dos grupos. Mantém também as credenciais dos utilizadores para a validação e introdução de novos na base de dados.



ElasticSearch

A base de dados desenvolvida usando a plataforma *ElasticSearch* é composta por 3 índices que são:

- Users;
- Get groups (/groups);
- Movies.

2.1 Users

Neste índice são armazenados todos os dados referentes a cada utilizador tais como *username*, *password* e *token*.

Para a criação de um novo usuário é realizado um acesso, recorrendo ao método *_search*, à base de dados para verificar se já existe algum *username* igual ao do usuário que se está a registar, isto para garantir que não possam existir dois usuários com o mesmo *username*, visto que é através deste que o mesmo vais se identificar.

No processo de *login* a abordagem adotada é de fazer uma pesquisa recorrendo ao método *_search* do *Elastic* que vai retornar uma *query* de objeto com o *username* que pretende fazer *login*. A partir da resposta proveniente da base de dados é verificado se a *password* e o *username* fornecidos pelo usuário pertencem ao mesmo objeto da *query*

de resposta da base de dados, e caso isso não se verifique esse usuário não poderá iniciar sessão com aquele *username* e *password*.

2.2 Groups

Os dados de cada grupo armazenados neste índice são:

- *_id*: identificador do documento gerado pelo *ElasticSearch*;
- *userId*: *token* do usuário;
- *name*: nome do grupo;
- *description*: descrição do grupo

Neste índice é guardado em cada entrada, além das informações do grupo, o *token* do usuário para que seja possível estabelecer uma relação entre os *users* e os seus respetivos grupos, desta forma quando o usuário quiser ver os seus grupos, são filtrados da base de dados os grupos que possuem no campo *userId* o *token* do usuário que tem a sessão iniciada. O parâmetro *_id* é utilizado pelas operações que envolvem a alteração do grupo, acesso aos detalhes do mesmo e eliminação.

Na operação de eliminação visto que neste índice só são armazenadas informações relativas ao grupo e nenhuma relativamente aos filmes que lhe pertencem, é realizado um pedido à base de dados para que o mesmo identifique os filme pertencentes ao grupo eliminado e os remova da mesma. Para tal foi feito um pedido do tipo *POST* ao *Elastic* com o método *_delete_by_query*, para que o mesmo elimine todos os filmes que possuem o *groupId* do grupo em questão.

2.3 Movies

Este índice vai armazenar as seguintes informações dos filmes provenientes da API:

- `_id`: identificador do documento gerado pelo *ElasticSearch*;
- `groupId`: `_id` do grupo;
- `id`: identificação proveniente da API;
- `title`: título do filme;
- `description`: descrição do filme;
- `runtime`: duração do filme em minutos;
- `year`: ano de publicação do filme;
- `image`: capa do filme;
- `directors`: diretores;
- `actors`: atores que participaram.

Este índice vai armazenar toda a informação do filme que é proveniente da API e o `_id` do grupo para que se possa estabelecer uma relação entre o grupo e os respetivos filmes. Desta forma quando se acede ao grupo todos os filmes com aquele `groupId` são apresentados ao utilizador, para que o mesmo possa ver quais os filmes estão presentes, possa eliminar ou aceder aos detalhes do mesmo.

Para a operação de consulta dos detalhes do filme é utilizado o `id`, que corresponde à identificação na API, e é feito um pedido para receber as informações do mesmo, já no processo de eliminação é recorre-se ao `_id`, identificação do documento armazenado na base de dados, para se fazer um pedido do tipo *DELETE* à base de dados para que a mesmo elimine o documento com aquele identificação na base de dados.

3

Documentação

Neste capítulo é apresentada a documentação para o site CMDb.

Method	Path	Query	Parameters	Auth
GET	api/popular	yes	no	no
GET	api/search	yes	no	no
GET	api/movie/:movieId	no	yes	no
POST	api/signup	no	no	no
GET	api/groups	no	no	yes
POST	api/groups	no	no	yes
GET	api/groups/:groupId	no	yes	yes
PUT	api/groups/:groupId	no	yes	yes
DELETE	api/groups/:groupId	no	yes	yes
POST	api/groups/:groupId/:movieId	no	yes	yes
DELETE	api/groups/:groupId/:movieId	no	yes	yes

Tabela 3.1: Caminhos do API

Method	Path	Query	Parameters	Auth
GET	/site.css	no	no	no
GET	/scripts/:scriptName	no	yes	no
GET	/	no	no	no
GET	/popular	yes	no	no
GET	/search	yes	no	no
GET	/movie/:movieId	no	yes	no
GET	/groups	no	no	yes
POST	/groups	no	no	yes
GET	/groups/:groupId	no	yes	yes
POST	/groups/:groupId	no	yes	yes

Tabela 3.2: Caminhos do WEB-SITE

3.1 Popular (api/popular)

Method:GET

Descrição:

Obtém, através do API do IMDb, os top **n filmes mais famosos** e apresenta os resultados, após filtração de informação, ao utilizador em formato *JSON*.

Query:

- max - Limita o número de resultados apresentados ao utilizador(250 por omissão).

3.2 Search (api/search)

Method:GET

Descrição:

Obtém, através do API do IMDb, os top **n resultados** para a procura por **movieName** e retorna a informação ao utilizador em formato *JSON*.

Query:

- max - Limita o número de resultados apresentados ao utilizador(250 por omissão);
- movieName - O nome do filme a ser procurado.

3.3 Get movie by Id (api/movie/:movieId)

Method:GET

Descrição:

Obtém, através do API do IMDb, as informações do filme com **movieId** e retorna as informações, após filtragem, ao utilizador em formato *JSON*.

Parameters:

- movieId - O id do filme.

3.4 Signup (api/signup)

Method:POST

Descrição:

Permite a criação de um **novo utilizador** e guarda as suas credenciais na **base de dados**.

Body Content:

- username - o nome do utilizador, **unico na base de dados**;
- password - a palavra-passe do utilizador.

3.5 List groups (api/groups)

Method: [GET](#)

Descrição:

Obtém da base de dados todos os grupos existentes para um **utilizador** e retorna a informação em formato *JSON*.

Request Content:

- user - um objecto com o **token** do utilizador

3.6 Create group (api/groups)

Method: [POST](#)

Descrição:

Cria um novo grupo para o utilizador com: **nome** e **descrição**.

Request Content:

- user - um objecto com o **token** do utilizador;
- name - o nome do novo grupo;
- description - a descrição do novo grupo.

3.7 Get group (api/groups/:groupId)

Method:GET

Descrição:

Obtém um dos grupos do utilizador e retorna a resposta em formato *JSON*.

Parameters:

- groupId - id do grupo a obter.

Request Content:

- user - um objecto com o **token** do utilizador

3.8 Update group (api/groups/:groupId)

Method:PUT

Descrição:

Atualiza o **nome** e a **descrição** do grupo na base de dados.

Parameters:

- groupId - id do grupo a atualizar

Request Content:

- user - um objecto com o **token** do utilizador;
- name - o novo nome do grupo;
- description - a nova descrição do grupo.

3.9 Remove group (/api/groups/:groupId)

Method:DELETE

Descrição:

Remove o grupo com o **groupId** do respetivo utilizador

Parameters:

- groupId - id do grupo a remover.

Request Content:

- user - um objecto com o **token** do utilizador.

3.10 Add movie (api/groups/:groupId/:movieId)

Method:POST

Descrição:

Adiciona um filme a um grupo escolhido pelo utilizador.

Parameters:

- groupId - id do grupo;
- movieId - id do filme a adicionar.

Request Content:

- user - um objecto com o **token** do utilizador.

3.11 Delete movie (api/groups/:groupId/:movieId)

Method:POST

Descrição:

Remove um filme do grupo escolhido pelo utilizador.

Parameters:

- groupId - id do grupo;
- movieId - id do filme a remover.

Request Content:

- user - um objecto com o **token** do utilizador.

3.12 Get CSS (site.css)

Method:GET

Descrição:

Obtém o ficheiro CSS que permite formatar os ficheiros HTML.

3.13 Get Script (scripts:ScriptName)

Method:GET

Descrição:

Obtém o ficheiro *JavaScript* a ser executado pelo ficheiro HTML.

Parameters:

- scriptName - o nome do ficheiro *JavaScript* a importar.

3.14 Get Home page (/)

Method:GET

Descrição:

Obtém o ficheiro HTML para a página principal.

3.15 Get Popular (/popular)

Method:GET

Descrição:

Versão em HTML da resposta de api-popular.

Query:

- max - limita o número de resultados a apresentar.

3.16 Search movie (/search)

Method:GET

Descrição:

Versão em HTML da resposta de api-search.

Query:

- max - limita o número de resultados a apresentar;
- movieName - o nome do filme a procurar.

3.17 Gets movie info (/movie/:movieId)

Method:GET

Descrição:

Versão em HTML da resposta de api-get-movie.

Parameters:

- movieId - Id do filme.

3.18 Get groups (/groups)

Method:GET

Descrição:

Versão em HTML da resposta de api-get-groups.

Request Content:

- user - um objecto com o **token** do utilizador

3.19 Create group (/groups)

Method:GET

Descrição:

Versão em HTML da resposta de api-create-groups.

Request Content:

- user - um objecto com o **token** do utilizador.

3.20 Get group (/groups/:groupId)

Method:GET

Descrição:

Versão em HTML da resposta de api-get-group.

Request Content:

- user - um objecto com o **token** do utilizador.

3.21 Add movie (/groups/:groupId)

Method:Post

Descrição:

Versão em HTML da resposta de api-add-movie.

Request Content:

- user - um objecto com o **token** do utilizador.

4

Conclusões

Com este trabalho pudemos aprofundar os conhecimentos sobre o desenvolvimento de uma aplicação web, entender a sua estrutura-base, as diferentes responsabilidades de cada módulo, a conceção de uma interface gráfica em *HTML/CSS* e a concessão de um sistema de credenciais para a validação do utilizador. Este trabalho foi, portanto, fundamental para a introdução ao desenvolvimento de aplicações na WEB e do trabalho envolvido na criação de uma mesma.

