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

15 – 01 – 2023

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

1

Introdução

Na fase anterior do trabalho, construi-se uma base de dados relacional capaz de guardar as informações de uma pequena empresa de transportes, as quais são produzidas através de códigos em *PostgreSQL*. Nesta fase do trabalho implementou-se a aplicação em *Java* que servirá de ligação entre o utilizador e a base de dados, de forma a permitir uma interação mais simples da parte do mesmo.

2

Estrutura do Programa

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

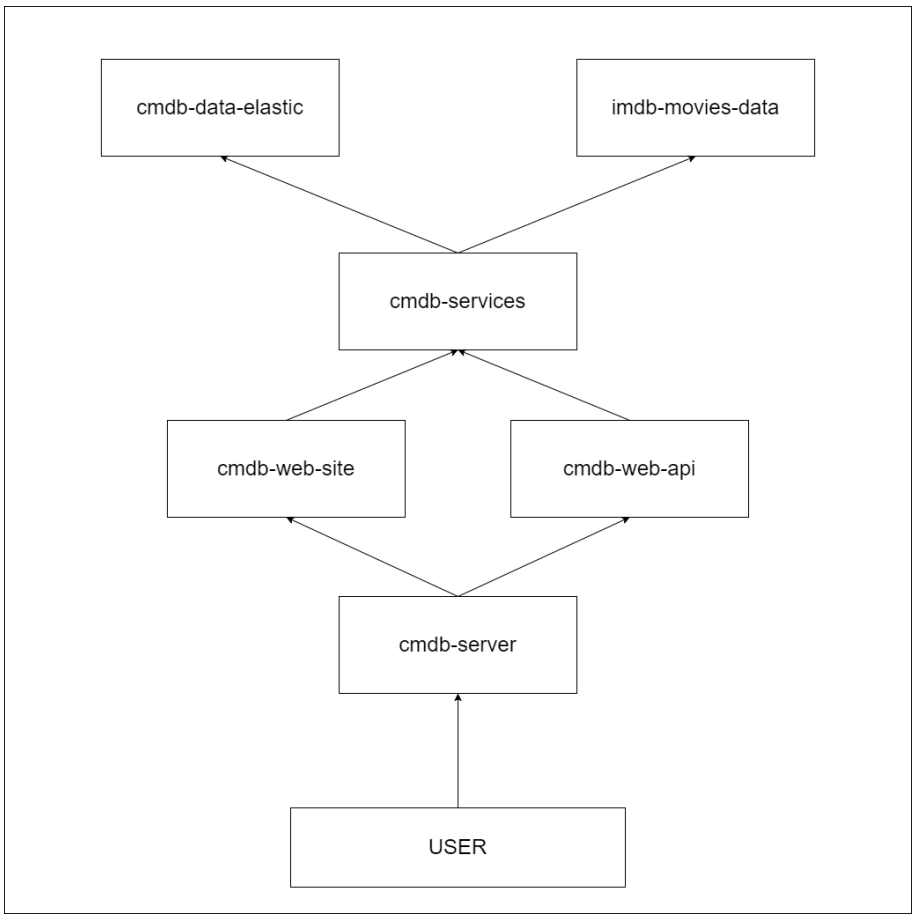

Figura 2.1: Estrutura do projeto.

O programa encontra-se subdividido em 6 partes:

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

2.1 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.

2.2 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.

2.3 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.

2.4 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.

2.5 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 titulo ou a apresentação dos top 250 filmes mais famosos.

2.6 Elastic data

Este é o módulo que serve como base para o

3

ElasticSearch

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

- Users;
- Groups;
- Movies.

3.1 Users

Neste indice 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*.

3.2 Groups

Os dados de cada grupo armazenados neste indice são:

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

Neste indice 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 indice 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.

3.3 Movies

Este indice 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: titulo 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 indice 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.

4

Conclusões

Com este trabalho aprofundámos o conhecimento sobre o desenvolvimento da aplicação CMDB. Assim