**Recomendador de Filmes**

**Trabalho de Programação em Lógica**

Realizado pelos alunos:

Maximiliano Vítor Phillips e Sá (up202305979),

Orlando Miguel Carvalho-Soares (up202303606),

e Rita Maria Pinho Moreira (up202303885)

**Índice**

1. **Introdução (objetivo)**

O seguinte projeto foi desenvolvido em contexto académico, de forma a consolidar competências de programação lógica. Para tal efeito, foi criado um sistema de recomendação de filmes, com interface web e integração com a API TMDb (The Movie Database).

O servidor principal, server.pl, junta todos os módulos auxiliares: gestão de utilizadores avaliações, recomendações, integração TMDb e interface HTML.

1. **Requisitos/Dependências do Sistema**

* OS: Windows/ Linux/ MacOS.
* SWI-Prolog instalado no PATH do sistema (para funcionar no terminal).

1. **Como Configurar**

* Descarregue a pasta.
* Aceda ao diretório "data" no terminal.
* Execute o comando: *swipl server.pl*
* Consulta: *server(PORT).* (Substitua o PORT pela porta pretendida, por exemplo, 8080).
* O servidor será executado em: *localhost:PORT*.

1. **Implementação**

Para a criação deste trabalho, foram criados diversos ficheiros, ambos contendo factos e regras respetivas a cada tópico abordado no mesmo. Esses ficheiros foram usados para o ficheiro server.pl, que culmina todo o código para o sistema de recomendação funcionar devidamente.

**Explicação do código no geral:**

**Cabeçalho:**

Para este sistema foram usadas variadas librarias built-in, tais como:

* library(http/http\_session): gere sessões HTTP;
* library(http/thread\_httpd): inicia o servidor HTTP;
* library(http/http\_dispatch): define handlers para rotas;
* library(http/http\_parameters): lê parâmetros de pedidos GET/POST;  
  library(http(html\_write): gere HTML em DCG;
* library(http/http\_client): fornece as principais ações HTTP, GET, DELETE; POST e PUT;
* library(http/http\_open): abre os dados num servidor HTTP como uma reprodução do código Prolog;
* library(apply): define meta-predicados que aplicam um predicado em todos os membros de uma lista
* library(http/json): suporta a leitura e escrita em objetos JSON;
* library(date): processa datas, definindo predicados de tempo e datas, tal como estruturas de dados;
* library(uri): fornece primitivas baseadas em C de alta performance, usadas para manipular URIs;
* library(http/http\_files): serve ficheiros simples de uma hierarquia.

Devido à utilização de ficheiros de certa forma externos ao ficheiro principal, foram usados os predicados ensure\_loaded/1, cujo argumento é um ficheiro dado para importar os seus predicados públicos, e module/1, que torna um módulo default de trabalho num módulo iterativo, visto que estamos a usar diversos ficheiros num só. Assim, foram importados os predicados dos ficheiros users.pl e recommend.pl, e foram importados os módulos knn, tmdb\_integration e rating.

O predicado initialization/1, que carrega um ficheiro dado, foi usado para definir a chave de acesso à API do TMDB (The Movie Database), usando um predicado set\_tmdb\_api\_key/1 pertencente ao módulo carregado tmdb\_integration, e para inicializar a base de dados das avaliações, com o predicado init\_ratings\_db/1, pertencente ao módulo rating, que, caso existe o ficheiro ratingsdb.pl, carrega-o.

Foi utilizado o predicado http\_handler/3, com a função de processar pedidos de endpoints individuais, não só processaram páginas de site como ficheiros estáticos de CSS e imagens. Cada http\_handler(root(X), Predicado, Opções) associa a rota “/X”, ou, para root(.), a rota “/”, a um predicado que vai gerar resposta HTML ou executar alguma lógica.

Por último, foi definida a base do API, com a chave (tmdb\_api\_key/1) e url (tmdb\_base\_url/1).

**Layout:**

Para o layout do server foi usado o predicado html\_meta/1, que declara uma regra de renderização HTML, usando o seu conteúdo como argumento.

Foi definido um page\_wrapper/2, com argumentos o título da página e o corpo, com acesso ao CSS e ao estado atual do utilizador.

**Server:**

Para gerar a sessão de HTTP, foi definido um predicado server/1, de argumento Port, que utiliza a biblioteca http\_server/2, e http\_dispatch e Port como respetivos argumentos.

**Movies:** base de dados, ficheiro gendb.pl, tmdb usage (tmdb\_integration.pl)

Para se poder aceder a filmes e suas respetivas informações, foi gerada, a partir do dataset imdb movies.csv, uma base de dados em Prolog, armazenada no ficheiro movie.pl. É usado o predicado db/3 para guardar o ID do filme e os seus dados, tais como o seu nome, duração, ano de estreia, avaliação, país, realizador e género. A base de dados foi gerada usando o ficheiro gendb.pl.

**Users:** DCG, base de dados e ficheiro users.pl

Tendo os filmes listados, é necessário ter usuários para os adicionar e avaliar. Para gerar a base de dados userdb.pl foram usados predicados do ficheiro users.pl, como new\_user/2. Para cada utilizador registado, é gerada a sua password encriptada, para maior segurança, e os seus dados (nome de utilizador e password) serão armazenados na base de dados. Se o nome de utilizador já existir, a mensagem será “Username already exists”, se não será “New User added and logged in”.

**Recommend:** KNN, recommend.pl

O recomendador que tem como base o KNN cria um vetor (User-Rating) para cada filme, calcula a similaridade do cosseno entre todos os vetores, criando um novo vetor (Similarity–Film), remove os filmes já vistos pelo utilizador dessa nova lista, ordena por ordem decrescente a similaridade, e retorna no máximo N (10 no nosso caso) filmes. Se N = 10 fica por aí, se N < 10, o recomendador usa os N filmes que encontrou com o KNN e depois recorre a fallbacks como por exemplo um recomendador que recomenda filmes do género favorito do utilizador (10-N). Logo o recomendador acaba sempre por recomendar 10 filmes. Na página do recomendador temos uma estatística que mostra quantos filmes foi o KNN e/ou os fallbacks.

O outro recomendador mais básico, recebe input do utilizador em forma de perguntas pré escolhidas, e com base nas respostas, devolve os filmes que atendem aos requisitos.

**Ratings:** rating.pl, ratingsdb.pl

Os utilizadores podem dar ratings (0-5 estrelas) e deixar um comentário sobre os filmes que viram. Esta funcionalidade também é uma peça fundamental para o nosso recomendador que usa KNN. Pois é dando match de ratings entre utilizadores que o recomendador decide se filmes devem ser recomendados a certo utilizador ou não. Os ratings são guardados como factos no ficheiro “ratingsdb.pl”.

1. **Resultados:**

Cada página do site tem o utilizador logged-in no canto superior esquerdo e um logo no canto superior direito. O logo no canto superior direito redireciona o utilizador de volta para a página inicial quando clicado.

**Home Page**

A página inicial apresenta o título " Welcome to Prolog, the Movie Recommender" e o subtítulo " All of your movie needs, in one place! :)" e tem os botões seguintes:

* Register
* Login
* Get Recommendations
* Show Your Films
* Show Your Ratings
* Show All Films
* A screenshot of a computer

  AI-generated content may be incorrect.Logout

**Register**

A página permite ao utilizador criar uma conta fornecendo um nome de utilizador e uma palavra-passe. O nome de utilizador deve ser um que ainda não exista na base de dados.

**Login**

Permite ao utilizador efetuar o login na sua conta com as suas credenciais de nome de utilizador e palavra-passe.

**Get Recommendations**

Abre uma página de menu com as seguintes opções de recomendação:

* *Based on Your Films*: O utilizador precisa de estar logged-in para funcionar. Gera uma lista de recomendações com base nas avaliações dos utilizadores e noutros filmes semelhantes com uma classificação elevada.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

* *By Specific Questions*: Abre uma página com as seguintes questões e gera recomendações com base nas respostas (se fornecidas):
* *Do you prefer older movies (pre-2000)?*
* *Yes (pre-2000)/ No, modern movies*
* *What types of movies are you in the mood for?*
* *Emotional/ Historical/ Cerebral/ Adventurous/ Funny*
* *Do you have time for longer movies?*
* *Yes (>119min)/ No (<120min)*
* *Which country's movie do you prefer?*
* *US/ UK/ Canada/ Japan/ Korea/ China*
* *Which country's movie do you prefer?*
* *Yes (>7)*

**Show Your Films**

É necessário estar logged-in para ter acesso. Apresenta uma lista dos filmes que o utilizador adicionou à sua conta, com as opções de remover ou fazer um rating ao lado de cada "título (ano)" da lista. O utilizador pode clicar no nome do filme para abrir a página respectiva.

**Show Your Ratings**

É necessário estar logged-in para ter acesso. Apresenta uma lista de todas as ratings feitas pelo utilizador atual e uma opção para ver os seus filmes ou regressar à página incial.

**Show All Films**

A screenshot of a computer

AI-generated content may be incorrect.Apresenta uma lista de todos os filmes na base de dados com opções de filtro (ano, país, género), um campo para pesquisar pelo nome do filme e botões para aplicar os filtros fornecidos pelo utilizador e repor os filtros para os valores predefinidos. Na parte inferior da página existe uma opção para voltar à página inicial.

**Film Pages**

A screenshot of a computer

AI-generated content may be incorrect.Página que exibe um poster e detalhes de qualquer filme, bem como botões para adicionar aos seus filmes, fazer um rating ou aceder à página oficial do IMDB. É necessário estar logged-in para ter acesso aos botões para adicionar aos seus filmes ou fazer um rating. Os ratings feitos pelos utilizadores para o filme também são listados na página do filme, juntamente com a opção de voltar à página inicial.

**Logout**

Encerra a sessão do utilizador atual e permanece na página inicial.

1. **Escalabilidade**
2. **Conclusão**
3. **Referências**

kaggle.com. (n.d.). *IMDb movies extensive dataset*. [online] Available at: https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset.

The Movie Database (2018). *The Movie Database*. [online] Themoviedb.org. Available at: https://www.themoviedb.org/.

Wilk, J. (2023). *How to Build a Movie Recommendation System Based on Collaborative Filtering*. [online] freeCodeCamp.org. Available at: https://www.freecodecamp.org/news/how-to-build-a-movie-recommendation-system-based-on-collaborative-filtering/.