**Índice**

**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.

**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

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

**Resultados:**

O que o site faz usando a teoria explicada acima, print de 1 ou 2 páginas. (README)

**Escalabilidade**

**Conclusão**

**Referências?**