



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Estudo de Técnicas de Filtragem Híbrida em Sistemas de Recomendação de Produtos

Carlos Eduardo Martins Barbosa

cemb@cin.ufpe.br

RECIFE, MARÇO DE 2014

Carlos Eduardo Martins Barbosa

Estudo de Técnicas de Filtragem Híbrida em Sistemas de Recomendação de Produtos

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Ricardo Bastos Cavalcante Prudêncio
(rbcp@cin.ufpe.br)

RECIFE, MARÇO DE 2014

Dedico este trabalho a todos aqueles que, de alguma forma, me deram forças para concluí-lo.

AGRADECIMENTOS

Gostaria de agradecer primeiramente a DEUS pela minha vida e saúde, com as quais tive a possibilidade de concluir este curso.

Gostaria de agradecer também aos meus pais pela minha criação e educação, por tudo o que tenho e por ser quem eu sou hoje.

Agradeço também aos meus irmãos e demais familiares, por estarem sempre ao meu lado, sempre dispostos a me ajudar.

Agradeço também ao professor Dr. Ricardo Bastos Cavalcante Prudêncio, por sua paciência, dedicação e orientação durante este projeto.

Por fim, agradeço à Universidade Federal de Pernambuco, ao Centro de Informática e seu corpo docente e em especial a todos os meus amigos que me acompanharam neste difícil caminho durante os últimos quatro anos e meio.

“Agradeço todas as dificuldades que enfrentei; não fosse por elas, eu não teria saído do lugar. As facilidades nos impedem de caminhar. Mesmo as críticas nos auxiliam muito”

- Chico Xavier

RESUMO

Em uma definição simples, Sistemas de Recomendação é uma área de pesquisa bastante rica que se utiliza de várias técnicas e ferramentas para prover sugestões de itens que sejam úteis para um usuário. No contexto destes sistemas, um item pode ser qualquer coisa que possa ser recomendado a um usuário, tal como um livro, um filme ou um pacote de viagem. Apesar de atualmente existirem diversas aplicações práticas nesta área, ela ainda necessita de melhorias que tornem seus métodos mais efetivos, de tal forma que represente melhor a informação sobre os itens a serem recomendados. Este trabalho de graduação apresenta uma visão geral da área de Sistemas de Recomendação, descrevendo as técnicas de recomendação mais utilizadas atualmente. São elas: filtragem baseada em conteúdo, filtragem colaborativa e filtragem híbrida. Estas técnicas serão comparadas em suas vantagens e desvantagens e um sistema foi desenvolvido para que essa análise fosse possível. Diferentes formas de se calcular a recomendação dos itens aos usuários serão avaliadas, utilizando-se métricas de avaliação conhecidas nesta área.

Palavras-chave: Sistemas de Recomendação; filtragem baseada em conteúdo; filtragem colaborativa; filtragem híbrida.

ABSTRACT

In a simple definition, Recommendation Systems is a very rich research area that utilizes various techniques and tools to provide suggestions for items that are useful for a user. In the context of these systems, an item can be anything that can be recommended for a user, such as a book, a movie or a travel package. Although currently there are several practical applications in this area, it still needs improvement to make their methods more effective, such that better represents the information about the items to be recommended. This graduate work presents an overview of Recommendation Systems area, describing the recommendation techniques more used nowadays. They are content-based filtering, collaborative filtering and hybrid filtering. These techniques are compared for their advantages and disadvantages and a system was developed for this analysis to be possible. Different ways to calculate the recommendation of items for users will be evaluated using evaluation metrics known in this area.

Keywords: Recommendation Systems; content-based filtering; collaborative filtering; hybrid filtering.

SUMÁRIO

1. Introdução.....	1
1.1 Motivação.....	1
1.2 Objetivo.....	2
1.3 Metodologia.....	2
1.4 Organização do Documento.....	2
2. Sistemas de Recomendação.....	4
2.1 Definição.....	4
2.2 Estratégias.....	6
2.3 Técnicas.....	6
2.3.1 Filtragem Baseada em Conteúdo.....	7
TF-IDF.....	8
Vantagens e Desvantagens.....	9
2.3.2 Filtragem Colaborativa.....	10
Baseada em Memória x Baseada em Modelo.....	11
Baseada em Usuário x Baseada em Item.....	12
Algoritmos.....	12
Algoritmos de Similaridade.....	13
Vector Cosine-Based Similarity.....	13
Adjusted Vector Cosine-Based Similarity.....	13
Pearson Correlation.....	14
Algoritmos de Previsão.....	14
Simple Weighted Average.....	14
Weighted Sum of Others' Ratings.....	15
Slope One.....	15
Algoritmos de Recomendação.....	16
Vantagens e Desvantagens.....	16
2.3.3 Filtragem Híbrida.....	16
2.4 Medidas de Avaliação de Desempenho.....	17
2.4.1 Medidas de Exatidão Preditiva.....	18
MAE.....	18

MSE	18
NMAE	18
RMSE	19
2.4.2 Medidas de Exatidão de Classificação.....	19
Precisão.....	20
Recall	20
F-measure	20
Fallout.....	21
Curva ROC	21
2.4.3 Medidas de Exatidão de Ranking	22
3. Análise da Literatura	23
3.1 Trabalhos Similares.....	23
3.1.1 PHOAKS	23
3.1.2 Sistema de Recomendação de Bibliotecas Digitais	23
3.1.3 SisRecCol.....	24
3.1.4 Fab	24
3.1.5 FEERS	24
3.1.6 e-Recommender	25
3.1.7 P-Tango.....	25
3.2 Análise e Considerações.....	25
4. Especificação e Implementação do Protótipo.....	26
4.1 Tecnologias Utilizadas	26
4.2 Arquitetura	27
4.3 Modelagem do Banco de Dados.....	28
4.4 Algoritmos Implementados	29
4.5 Lucene	29
4.6 Perfil do Usuário	30
4.7 Recomendações Híbridas	32
4.7.1 Recomendação Híbrida Ponderada.....	32
4.7.2 Recomendação Híbrida pela Soma dos Inversos das Posições.....	33
4.8 Descrição do Protótipo	34
5. Experimentos e Resultados.....	64

5.1 Experimento 1 – Algoritmo Colaborativo.....	64
5.2 Experimento 2 – Algoritmo de Similaridade	66
5.3 Experimento 3 – Tamanho da Vizinhança	67
5.4 Experimento 4 – Parâmetro K.....	69
5.5 Experimento 5 – Parâmetro α	70
5.6 Experimento 6 – Algoritmo Híbrido	72
5.7 Experimento 7 – Técnica de Filtragem	73
5.8 Experimento 8 – Problema da Superespecialização.....	76
6. Considerações Finais	79
Referências Bibliográficas.....	80
Apêndice.....	83
Prova por Indução	83

LISTA DE ABREVIATURAS

API	Application Programming Interface
AUC	Area under Curve
FBC	Filtragem Baseada em Conteúdo
FC	Filtragem Colaborativa
FH	Filtragem Híbrida
KNN	K-Nearest Neighbors
MAE	Mean Absolute Error
MSE	Mean Squared Error
NMAE	Normalized Mean Absolute Error
RMSE	Root Mean Squared Error
ROC	Receiver Operating Characteristic
TF-IDF	Term Frequency - Inverse Document Frequency

LISTA DE FIGURAS

Figura 1 - Classificação das técnicas de filtragem	6
Figura 2 – Filtragem baseada em conteúdo	7
Figura 3 – Efetividade na busca	9
Figura 4 - Filtragem colaborativa	11
Figura 5 - Curva ROC	21
Figura 6 - Esboço da arquitetura do sistema desenvolvido	27
Figura 7 - Diagrama entidade relacionamento do sistema desenvolvido	28
Figura 8 - Tela de login	34
Figura 9 - Tela do administrador	34
Figura 10 - Tela de configuração de usuários	35
Figura 11 - Tela de configuração de categorias.....	35
Figura 12 - Tela de configuração de autores	36
Figura 13 - Tela de configuração de livros.....	36
Figura 14 - Tela de configuração de avaliações de usuários	37
Figura 15 - Tela de configuração de avaliações previstas	37
Figura 16 - Distribuição do número de avaliações por nota.....	38
Figura 17 - Distribuição do número de itens por número de avaliações	39
Figura 18 - Distribuição do número de usuários por número de avaliações	39
Figura 19 - Gráfico de esparsidade do sistema.....	40
Figura 20 - Tela dos itens do sistema	41
Figura 21 - Filtro de itens avaliados	42
Figura 22 - Ordenação dos itens por atributo	43
Figura 23 - Filtro de itens por atributo textual	44
Figura 24 - Filtro de itens por atributo numérico	45
Figura 25 - Tela de similaridade de usuários.....	46
Figura 26 - Tela de similaridade de itens	47
Figura 27 - Tela de centroides do perfil de usuário.....	48
Figura 28 - Tela de itens dos clusters do perfil do usuário.....	49
Figura 29 - Tela de recomendações.....	50
Figura 30 - Tela de comparação entre notas prevista e real dos itens já avaliados	51
Figura 31 - Comparação das medidas de erro de predição.....	52
Figura 32 - Comparação das medidas de erro de classificação	53
Figura 33 - Comparação de erro MAE dos algoritmos de similaridade colaborativos	54
Figura 34 – Comparação de erro MAE das técnicas de filtragem.....	55
Figura 35 - Comparação de erro MAE dos algoritmos colaborativos.....	56
Figura 36 - Comparação de erro MAE dos algoritmos híbridos	57
Figura 37 - Comparação de curva ROC entre algoritmos híbridos	58
Figura 38 - Variação de erro MAE com o tamanho da vizinhança	59

Figura 39 - Variação de erro MAE com o parâmetro α	60
Figura 40 - Variação de erro MAE com o parâmetro K.....	61
Figura 41 - Variação de erro MAE com a quantidade de iterações.....	62
Figura 42 - Matriz de confusão	63
Figura 43 - Experimento 1: MSE dos algoritmos de predição colaborativa	65
Figura 44 - Experimento 1: Curva ROC dos algoritmos de predição colaborativa	65
Figura 45 - Experimento 2: MSE dos algoritmos de similaridade colaborativa	66
Figura 46 - Experimento 2: Curva ROC dos algoritmos de similaridade colaborativa.....	67
Figura 47 - Experimento 3: MSE da variação da quantidade de vizinhos	68
Figura 48 - Experimento 3: Curva ROC da variação da quantidade de vizinhos.....	68
Figura 49 - Experimento 4: MSE da variação do parâmetro K.....	69
Figura 50 - Experimento 4: Curva ROC da variação do parâmetro K.....	70
Figura 51 - Experimento 5: MSE da variação do parâmetro α	71
Figura 52 - Experimento 5: Curva ROC da variação do parâmetro α	71
Figura 53 - Experimento 6: MSE dos algoritmos híbridos	72
Figura 54 - Experimento 6: Curva ROC dos algoritmos híbridos.....	73
Figura 55 - Experimento 7: MSE das técnicas de filtragem.....	74
Figura 56 - Experimento 7: Curva ROC das técnicas de filtragem.....	74
Figura 57 - Experimento 7: MAE das técnicas de filtragem	75
Figura 58 - Experimento 7: Precisão das técnicas de filtragem	75
Figura 59 - Experimento 8: Avaliações	76
Figura 60 - Experimento 8: Recomendação baseada em conteúdo	76
Figura 61 - Experimento 8: Recomendação colaborativa.....	77
Figura 62 - Experimento 8: Recomendação híbrida.....	77

LISTA DE TABELAS

Tabela 1 - Matriz de confusão	20
--	----

LISTA DE EQUAÇÕES

Equação 1 – Função de utilidade da recomendação	5
Equação 2 - Frequência do termo (TF)	8
Equação 3 – Frequência inversa (IDF)	8
Equação 4 - Similaridade do cosseno	8
Equação 5 – Similaridade do vetor cosseno	13
Equação 6 - Similaridade do vetor cosseno ajustado	13
Equação 7 - Correlação de Pearson entre usuários	14
Equação 8 - Correlação de Pearson entre itens	14
Equação 9 - Predição do algoritmo Simple Weighted Average	14
Equação 10 - Predição do algoritmo Weighted Sum of Others' Ratings	15
Equação 11 - Predição do algoritmo Slope One	15
Equação 12 - MAE	18
Equação 13 – MSE	18
Equação 14 – NMAE	19
Equação 15 - RMSE	19
Equação 16 - Precisão	20
Equação 17 - Recall	20
Equação 18 - F-measure	20
Equação 19 – Fallout	21
Equação 20 - Combinação linear da filtragem híbrida ponderada	32
Equação 21 - Filtragem híbrida com heurística da quantidade de avaliações	33
Equação 22 – Fórmula do nível de esparsidade	40
Equação 23 - Nível de esparsidade do sistema	41

1. Introdução

O surgimento da Internet possibilitou o acesso à informação para muitas pessoas. Muitos mecanismos de busca, como Google e Yahoo! passaram a ser bastante utilizados para localização de informações sobre diversos assuntos. Além disso, diversas mensagens de e-mail passaram a ser usadas para troca de informação entre as pessoas. Tudo isso fez com que a quantidade de dados armazenados na Internet crescesse muito rapidamente, passando a ser cada vez mais difícil um usuário localizar as informações de seu interesse. Desta forma tornou-se necessário o surgimento e a utilização de ferramentas de filtragem automáticas [25]. Surgiram assim os sistemas de recomendação.

Hoje em dia, os sistemas de recomendação possuem aplicações em muitas áreas. Eles estão presentes nos mecanismos de busca, em sites de leilão (eBay), em sites de comércio eletrônico (Amazon, Last.fm), na área acadêmica (SisRecCol, RecS-DL), e em muitas outras áreas [28].

1.1 Motivação

Sistemas de Recomendação é uma importante área de pesquisa, na qual muito trabalho tem sido feito, tanto academicamente quanto comercialmente. O interesse nessa área se deve à grande quantidade de problemas e aplicações práticas existentes, tornando-a um dos maiores desafios virtuais existentes atualmente [28]. Além disso, o rápido crescimento da Internet demanda sistemas de recomendação cada vez mais eficazes, para que seja possível filtrar apenas as informações relevantes na enorme quantidade de informações disponíveis.

Outro fator a se destacar é que os sistemas de recomendação vêm sendo adotados por grandes sistemas comerciais, tais como Amazon, Netflix e Google. Estas grandes empresas perceberam que um bom sistema de recomendação para seus produtos e serviços lhes dá uma importante vantagem competitiva, implicando diretamente em seu lucro, devido ao aumento do consumo e da fidelização do cliente [1]. Estima-se, por exemplo, que 35% das vendas da Amazon, 38% das notícias mais clicadas da Google e 2/3 dos aluguéis de filmes da Netflix são provenientes das suas recomendações [30].

1.2 Objetivo

Esse trabalho de graduação tem como objetivo realizar uma revisão da literatura relacionada a sistemas de recomendação, em especial os trabalhos que dizem respeito a técnicas de filtragem híbridas. Esta revisão mostrará a importância de tais sistemas no contexto atual, bem como irá comparar suas diversas abordagens. Objetiva-se assim mostrar como as técnicas de filtragem híbridas fortalecem as vantagens e minimizam as principais desvantagens de suas duas componentes: filtragem baseada em conteúdo e filtragem colaborativa.

Em seguida o problema será formalizado e um estudo de caso será feito escolhendo-se uma das técnicas de filtragem híbrida estudadas. Para viabilizar esse estudo, pretende-se implementar um sistema de recomendação. Diversos testes serão realizados nesse sistema, que contará com uma base de dados formada por informações extraídas da Amazon. Os resultados destes testes serão analisados, tanto a nível de relevância da recomendação quanto a nível de desempenho.

1.3 Metodologia

Durante o desenvolvimento deste trabalho foram realizadas as seguintes atividades:

- Estudo sobre sistemas de recomendação, como eles surgiram, sua definição, arquitetura, estratégias e técnicas;
- Estudo sobre trabalhos similares, a fim de conhecer experiências no desenvolvimento de sistemas de recomendação;
- Modelagem do sistema;
- Definição de parâmetros e estratégias a serem usadas no sistema proposto;
- Desenvolvimento e testes do sistema.

1.4 Organização do Documento

Este trabalho está organizado da seguinte forma: no capítulo 2 são abordados os fundamentos teóricos de Sistemas de Recomendação, detalhando os conceitos necessários para o entendimento do trabalho, bem como seu histórico, estratégias, técnicas e métricas de análise de desempenho mais utilizadas. No capítulo 3 é abordada a análise da literatura da área, com

exemplos de uso destes sistemas em meio acadêmico ou comercial. No capítulo 4 é explicado o protótipo de sistema de recomendação desenvolvido, bem como o corpus de dados utilizados no sistema, sua modelagem, arquitetura e principais funcionalidades. No capítulo 5 são descritos os experimentos realizados neste protótipo e a análise dos resultados destes experimentos. Por fim, no capítulo 6 são feitas as considerações finais sobre o trabalho, com suas contribuições e ideias de trabalhos futuros.

2. Sistemas de Recomendação

Os primeiros sistemas de recomendação surgiram em meados da década de 90, quando pesquisadores começaram a analisar alguns problemas de recomendação que dependiam de uma estrutura de avaliações. O primeiro sistema de recomendação propriamente dito foi o Tapestry, desenvolvido no Centro de Pesquisa da Xerox em Palo Alto, com a finalidade de filtrar a grande quantidade de e-mails que já estava incomodando os usuários de grupos de notícias. A ideia principal do Tapestry era exibir todos os artigos que um certo usuário considerasse relevante [14].

Os sistemas de recomendação surgiram para resolver o problema de sobrecarga de informações e para realizar indicações de itens aos usuários, sejam livros, artigos, filmes, restaurantes ou outras informações. Hoje em dia sua utilização é maior em lojas de comércio eletrônico, nas quais são fundamentais e cada vez mais relevantes, pois conseguem aprender sobre os consumidores e recomendar produtos de seu interesse [21].

2.1 Definição

Um sistema de recomendação é aquele que produz recomendações ao usuário, de itens que sejam de seu interesse ou que sejam úteis entre várias opções. Eles podem ainda identificar similaridade entre usuários e recomendar itens que já foram recomendados para usuários similares [9]. Além de recomendar os itens aos usuários, um sistema de recomendação também objetiva manter o usuário "on-line", navegando no sistema, aumentando assim o seu interesse de compra e fidelidade [20].

Um sistema de recomendação tem como componentes principais as informações sobre os itens e as informações sobre os usuários, sejam estas detalhadas em diversos atributos ou apenas descritas em termos de uma avaliação dada por um usuário a um item. Estas informações são armazenadas antes de o processo de recomendação ser iniciado. Também fazem parte de um sistema de recomendação os dados de entrada, que devem ser informados pelo usuário, e um algoritmo para combinar todas as informações e produzir as recomendações [9].

Na grande maioria dos casos, o problema da recomendação consiste em estimar notas para itens ainda não avaliados pelo usuário. Após estas estimativas, faz-se a recomendação dos itens cujas notas estimadas forem maiores. Uma definição mais formal deste problema pode ser vista a seguir:

Imagine que um determinado sistema tem uma grande quantidade de itens em seu catálogo de vendas e um usuário pretende realizar compras nesse sistema. Então, seja U o conjunto de todos os usuários desse sistema, e seja I o conjunto de todos os possíveis itens que podem ser recomendados a estes usuários (livros, filmes, restaurantes, etc.). Seja f a função utilidade que mede o quão útil é um determinado item i para um determinado usuário u e R um conjunto totalmente ordenado. Então, para cada usuário u que pertence a U , procura-se um item i' que pertence a I que maximiza a utilidade do usuário. Isto pode ser expressado pela equação 1 abaixo:

$$f: U \times I \rightarrow R$$

$$\forall c \in C, s' c = \operatorname{argmax}_{s \in S} = u(c, s)$$

Equação 1 – Função de utilidade da recomendação

Cada elemento do espaço de usuários U pode ser definido através de um perfil com várias características, tais como sexo, idade, profissão, estado civil. Da mesma forma, supondo que o espaço de itens I corresponde a uma coleção de livros, ele pode ser definido pelos atributos autor, categoria, título, sinopse, editora. Mas dependendo da técnica de recomendação utilizada, pode-se armazenar apenas um identificador ou código para cada usuário e item, que é o caso da filtragem puramente colaborativa, que só precisa dos valores das avaliações dos usuários aos itens [1].

A coleta das informações do usuário pelos sistemas de recomendação podem ser realizadas de forma explícita, na qual o usuário indica quais são as suas preferências espontaneamente, a partir da atribuição de uma nota de valor numérico ou marca positiva/negativa aos itens, ou implícita, na qual podem ser considerados vários fatores, tais como número de acessos ou tempo de acesso ao item, busca e compra de produtos [23]. Uma das desvantagens das avaliações explícitas é não se ter bons critérios de avaliação, afinal não dá para saber se um item muito avaliado com quatro estrelas é melhor que um item pouco avaliado com cinco estrelas.

A finalidade de um sistema de recomendação é determinar uma nota que reflita o grau de relevância, para um determinado usuário, de um item que ele ainda não avaliou. A partir destas notas, verifica-se quais são os itens que serão recomendados aos usuários. Mas também há sistemas de recomendação onde o importante não é o valor das avaliações estimadas, e sim a ordem em que os itens são recomendados ao usuário [25].

2.2 Estratégias

Um sistema de recomendação pode utilizar várias estratégias para personalizar a recomendação dos seus produtos. Uma estratégia de recomendação consiste na forma como as recomendações dos itens serão agrupadas ou visualizadas dentro do sistema. Dentre as principais estratégias de recomendação, destacam-se as listas de recomendação, as recomendações personalizadas, as associações por conteúdo e as avaliações de usuário. As listas de recomendação mostram as recomendações baseando-se no número de avaliações dos itens, na quantidade de vendas, etc. É uma estratégia de fácil implementação, mas perde o foco no usuário, pois a lista vai ser a mesma para todos os usuários. As recomendações personalizadas oferecem diversos tipos de filtros para as recomendações, tais como preço, nota média, etc. As associações por conteúdo mostram as recomendações agrupadas por algum atributo, como autor ou categoria. A estratégia de avaliações de usuário disponibiliza ao usuário a opção de avaliar o item e armazenar sua avaliação. Tem como desvantagem o fato de depender da veracidade das notas atribuídas pelos usuários [23]. No protótipo desenvolvido neste trabalho foi utilizada uma combinação destas estratégias, permitindo recomendações personalizadas, com opção de filtros e avaliações de usuários, em uma lista ordenada de forma decrescente pela nota prevista pelo sistema.

2.3 Técnicas

Diferentemente das estratégias de recomendação, que definem como uma sugestão será feita, as técnicas de recomendação são usadas para a efetiva realização da recomendação, a partir de uma predição sobre as informações dos itens e usuários. As principais técnicas de recomendação podem ser visualizadas na figura 1 a seguir:

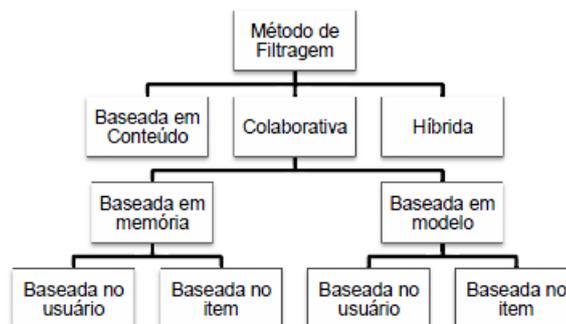


Figura 1 - Classificação das técnicas de filtragem

Apesar de as filtragens baseada em conteúdo, colaborativa e híbrida serem as principais, muitas outras técnicas estão sendo usadas na tentativa de identificar o comportamento dos usuários e de como usar estas informações para personalizar o atendimento aos usuários [23].

2.3.1 Filtragem Baseada em Conteúdo

Esta categoria tem sua raiz na área de recuperação e filtragem de informação, focando assim na recomendação de itens que contém informações textuais, tais como documentos e sites da Web [5]. O conteúdo nesse tipo de filtragem é geralmente descrito por palavras-chave. A importância de uma palavra em um documento pode ser definida de diferentes formas, sendo a mais comum a TF-IDF, um modelo estatístico definido por Salton [26], em 1993. Como o nome já lembra, TF-IDF atribui um peso maior para palavras que aparecem muito em um documento, mas que aparecem em poucos documentos, já que palavras que aparecem em muitos documentos não são úteis para dizer se um documento é relevante ou não. Nesta técnica, para realizar a associação de conteúdo entre os itens, é necessário identificar os atributos em comum entre eles. No conjunto de livros utilizados no sistema desenvolvido neste projeto foram identificados os atributos título, autor, categoria e sinopse. Desta forma o perfil de um usuário é definido a partir dos valores destes atributos dos itens já avaliados por ele. Assim, um novo item pode ser recomendado ao usuário de acordo com a sua similaridade (normalmente através da medida do cosseno) com os interesses do usuário extraídos do seu perfil, como pode ser visualizado na figura 2.

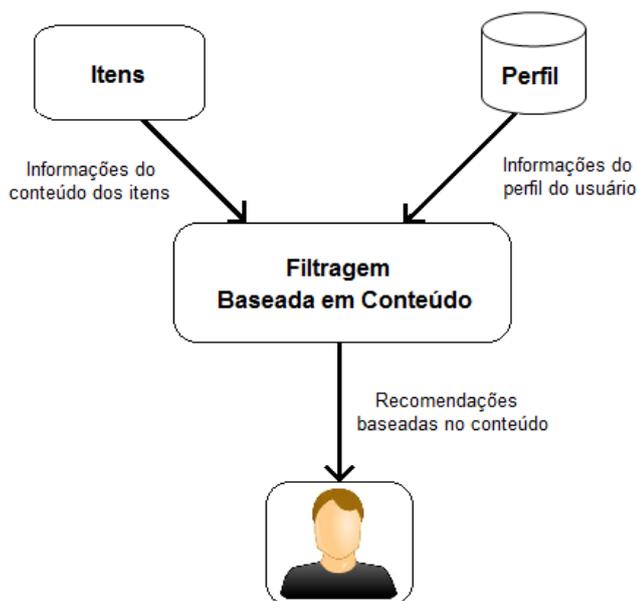


Figura 2 – Filtragem baseada em conteúdo

TF-IDF

No algoritmo TF-IDF utilizado na técnica de filtragem baseada em conteúdo, um conjunto de N documentos contém um conjunto de n termos, onde cada documento pode ser representado por um vetor do espaço n-dimensional de pesos dos termos, onde cada peso denota a importância de um termo para um usuário. A frequência do termo é calculada a partir da divisão da quantidade de ocorrências do termo no documento pela quantidade de ocorrências de todos os termos no documento, conforme pode ser visto na equação 2:

$$tf = \frac{n_i}{\sum_k n_k}$$

Equação 2- Frequência do termo (TF)

A frequência inversa transmite a relevância do termo, sendo calculada a partir do logaritmo da divisão da quantidade total de documentos sobre a quantidade total de documentos que contém aquele termo, conforme pode ser visualizado na equação 3:

$$idf = \log \frac{|D|}{|(d_i \supset t_i)|}$$

Equação 3- Frequência inversa (IDF)

Desta forma, é fácil perceber que os termos que apresentarão maior peso, e portanto diferenciarão melhor um documento dos outros do sistema, são aqueles que possuem alto valor de TF e baixo valor de IDF. Estes valores de frequência podem ser normalizados, dividindo-os pelos seus valores máximos. O valor final da medida TF-IDF é obtido multiplicando-se estes dois valores de frequência [26].

Por fim, a similaridade entre o perfil do usuário e o item que se quer recomendar pode ser definida utilizando-se o cosseno do ângulo entre os vetores TF-IDF de pesos das palavras-chave dos mesmos, como pode ser visualizado na equação 4:

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

Equação 4- Similaridade do cosseno

Sendo assim dois itens são mais similares quanto mais próximos se localizarem espacialmente os seus vetores [26].

A efetividade na busca por palavras-chave é um ponto que vem sendo bastante estudado pelos pesquisadores da área. Muitas vezes um documento relevante não é retornado na busca por não possuir o termo procurado como palavra-chave da consulta, e sim um sinônimo. Outro problema é identificar qual é o contexto em que um termo está sendo procurado, a fim de eliminar a ambiguidade da consulta [3]. Isso ocorre, por exemplo, ao pesquisar o termo JAGUAR em um mecanismo de busca. Serão retornadas algumas páginas referentes ao automóvel, outras referentes ao animal e outras referentes ao esquadrão da Força Aérea Brasileira, como pode ser visto na figura 3.

[Jaguar International - Market selector page](#)
www.jaguar.com/ ▾
Official worldwide web site of Jaguar Cars. Directs users to pages tailored to country-specific markets and model-specific websites.
[Jaguar International](#) - [Jaguar Canada](#) - [Jaguar India](#) - [Hong Kong](#)

[News for jaguar](#)

 [How reliable is a Jaguar X-type?](#)
Telegraph.co.uk - 1 day ago
Honest John advises a reader on how to maintain their Jaguar X-type.

[Jaguar Land Rover Posts Record Sales for 2013](#)
Wall Street Journal - 6 days ago

[Jaguar F-type R Coupé on film](#)
Telegraph.co.uk - 6 days ago

[Jaguar - Wikipedia, the free encyclopedia](#)
en.wikipedia.org/wiki/Jaguar ▾
The jaguar Panthera onca, is a big cat, a feline in the Panthera genus, and is the only Panthera species found in the Americas. The jaguar is the third-largest ...
[Jaguar Cars](#) - [Jaguar \(disambiguation\)](#) - [Jacksonville Jaguars](#) - [White panther](#)

[Jaguar Cars - Wikipedia, the free encyclopedia](#)
en.wikipedia.org/wiki/Jaguar_Cars ▾
Jaguar Cars is a brand of Jaguar Land Rover, a British multinational car manufacturer headquartered in Whitley, Coventry, England, owned by Indian Tata ...

[Images for jaguar](#) - Report images



Figura 3 – Efetividade na busca

Vantagens e Desvantagens

Dentre as principais vantagens da técnica de filtragem baseada em conteúdo, destaca-se o fato de ela não precisar que um item já tenha sido avaliado para gerar a recomendação, além de todos os itens terem chance de serem recomendados, já que dependem apenas do perfil do usuário.

Uma das principais desvantagens da filtragem baseada em conteúdo é que para a criação do perfil do usuário, ele tem que ter avaliado um número suficiente de itens, a fim de que as recomendações sejam mais precisas. Assim um usuário novo, que tem poucas avaliações, não terá recomendações muito precisas. Além disso, o usuário fica limitado a itens similares aos de seu perfil. Este problema é conhecido como superespecialização. Por exemplo, se um determinado usuário possui em seu perfil filmes de terror, mas também goste de filmes de ação, talvez ele nunca receba uma recomendação de filmes de ação. Outro problema é que a filtragem baseada em conteúdo não funciona bem em domínios que não sejam textuais, como imagens, vídeos e áudios, por ser difícil extrair os atributos relevantes dos mesmos. Além disso, os itens são filtrados sem considerar a qualidade dos itens a serem recomendados, assim dois itens distintos que apresentem os mesmos valores de atributos serão tratados como iguais, mesmo que um seja de qualidade e o outro não [19].

2.3.2 Filtragem Colaborativa

A técnica de filtragem colaborativa é a técnica de recomendação mais utilizada e tem sua essência na troca de experiências entre usuários que possuem interesses em comum [27]. Sendo assim, a recomendação dos itens ao usuário é feita levando-se em consideração as preferências desse usuário e as preferências dos usuários que são semelhantes a ele, como pode ser visualizado na figura 4. Isto é feito porque usuários semelhantes tendem a gostar das mesmas coisas. Antigamente os sistemas de filtragem colaborativa requeriam que o usuário informasse quais eram os seus interesses. Hoje em dia, com os avanços feitos na área, este processo foi automatizado: o sistema identifica os interesses do usuário a partir das avaliações dadas por ele aos itens [23]. Nesta abordagem é realizado um cálculo de similaridade entre os usuários ou itens, a partir de suas avaliações. Por fim o valor da pontuação de um usuário para um item que ele ainda não avaliou é previsto a partir desses valores de similaridade.

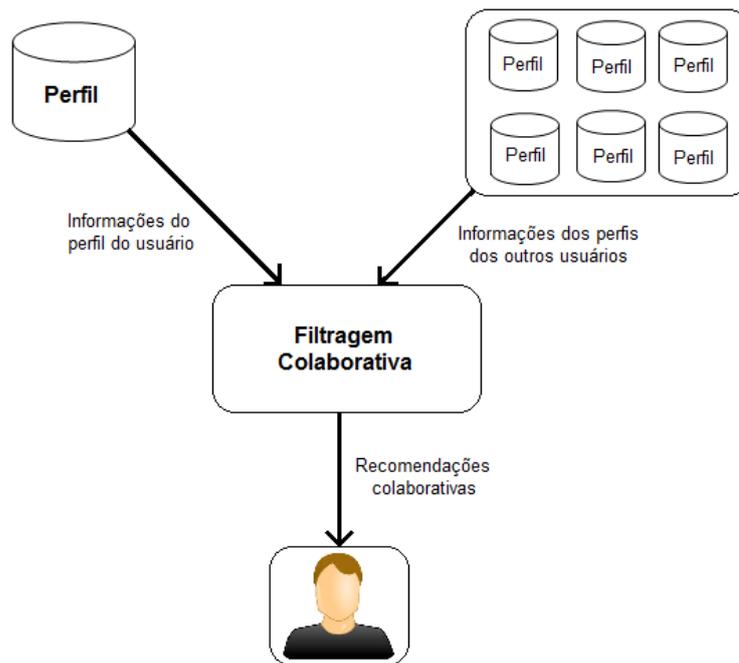


Figura 4 - Filtragem colaborativa

Como foi visto na figura 1, existem dois tipos diferentes de filtragem colaborativa, a baseada em memória e a baseada em modelo. As duas podem ainda se basear nos itens ou nos usuários para a recomendação. As que se baseiam nos usuários usam a similaridade entre um usuário e os seus vizinhos mais próximos para prever a nota que ele daria a itens que ainda não avaliou. As que se baseiam em itens, ao invés de calcularem a similaridade entre os usuários, realizam as previsões a partir da similaridade entre os itens.

Baseada em Memória x Baseada em Modelo

A filtragem colaborativa baseada em memória é a pioneira e calcula as avaliações previstas baseando-se em uma heurística de vizinhança dos itens ou dos usuários [27]. Sua principal desvantagem é que uma matriz usuário x item que tenha grandes dimensões irá fazer com que o cálculo das similaridades e previsões se torne muito caro computacionalmente, já que esta técnica utiliza toda a base de dados de usuários e itens.

A filtragem colaborativa baseada em modelo surgiu para reduzir o tempo necessário para o cálculo das previsões nas recomendações e ficou mais conhecida após o concurso da Netflix [6] com o algoritmo baseado em fatores latentes, vencedor do concurso. Nela, a partir das informações de itens, usuários e avaliações disponíveis, um modelo é elaborado e utilizado para o cálculo das previsões, normalmente através de dados estatísticos ou de técnicas de

aprendizagem de máquina. A recomendação baseada em modelo costuma ter melhor acuidade para vários casos, além de resolver o problema da escalabilidade que a filtragem colaborativa baseada em memória apresenta, pois quando uma recomendação é solicitada, as relações de similaridade entre usuários ou itens já estarão presentes no modelo inicialmente construído. Porém o processo de aprendizado do modelo pelo sistema pode ser bem complexo e demorado. Com isso, apesar de mais demorada, a técnica baseada em memória tende a apresentar resultados melhores do que a técnica baseada em modelo, além de levar vantagem também no ponto de justificativa da recomendação, por ser intuitivo mostrar quais os itens e usuários similares na recomendação [19].

Baseada em Usuário x Baseada em Item

Os sistemas de recomendação que utilizam a filtragem colaborativa baseada em itens recomendam itens ao usuário baseando-se nas avaliações (notas) que este usuário já atribuiu a outros itens. Já os sistemas de filtragem colaborativa baseada em usuário recomendam itens baseando-se nas notas que outros usuários deram a este item [27].

Quanto mais avaliações existirem dos mesmos usuários sobre os itens do sistema, maior será o foco nos itens, por eles criarem um relacionamento entre si. Sendo assim, para sistemas em que o número de usuários é muito maior que o número de itens, é preferível uma recomendação colaborativa baseada em itens. Já no caso em que o número de itens é muito maior que o número de usuários, é preferível uma recomendação colaborativa baseada em usuários. Além disso, se o número de usuários é muito maior que o número de itens, a recomendação colaborativa baseada em itens requer menos memória e menos cálculos de similaridade [12].

O fator de novidade, ou a chance de se recomendar itens não esperados, é maior nas recomendações baseadas em usuários, pois na baseada em itens haverá poucos itens que o usuário não espera receber como recomendação, até porque o número de itens tende a ser menor que o número de usuários nessa recomendação [12].

Algoritmos

Geralmente os sistemas de recomendação colaborativos dividem-se em três etapas: a formação da vizinhança de usuários ou itens, a partir de um subconjunto de usuários ou itens

com maiores valores de similaridade, o cálculo da previsão de notas entre um usuário e os itens que ele ainda não avaliou, e a recomendação dos itens melhor avaliados [18].

Algoritmos de Similaridade

O cálculo da similaridade em um algoritmo de filtragem colaborativa é um passo muito importante. Em filtragens baseadas em itens, a similaridade entre dois itens é calculada usando-se os usuários que avaliaram a ambos. Em filtragens baseadas em usuário, a similaridade entre dois usuários é calculada considerando os itens avaliados por ambos [31].

Os algoritmos de similaridade local que foram abordados e desenvolvidos neste trabalho são do tipo KNN, ou seja, baseados na vizinhança dos usuários ou itens mais próximos. Neste algoritmo foram usadas três diferentes técnicas: o cálculo do cosseno do ângulo entre dois vetores, a medida do cosseno ajustado e o cálculo do coeficiente de correlação linear de Pearson. Os resultados do cálculo de similaridade da técnica de filtragem colaborativa variam entre 1, caso em que há total similaridade, e -1, caso em que há total dissimilaridade [15].

Vector Cosine-Based Similarity

A similaridade entre dois usuários ou itens também pode ser calculada como o cosseno do ângulo entre eles (equação 5):

$$sim(i, j) = \frac{\sum_{u \in U_{ij}} A_{u,i} A_{u,j}}{\sqrt{\sum_{u \in U_{ij}} A_{u,i}^2} \sqrt{\sum_{u \in U_{ij}} A_{u,j}^2}}$$

Equação 5 – Similaridade do vetor cosseno

Adjusted Vector Cosine-Based Similarity

Um problema da abordagem anterior é que ela não considera a diferença entre as notas dos usuários. Para isso existe o cálculo do Vetor Cosseno Ajustado, que subtrai a média de todas as avaliações do usuário, como pode ser visto a seguir na equação 6:

$$sim(i, j) = \frac{\sum_{u \in U_{ij}} (A_{u,i} - \bar{A}_u)(A_{u,j} - \bar{A}_u)}{\sqrt{\sum_{u \in U_{ij}} (A_{u,i} - \bar{A}_u)^2} \sqrt{\sum_{u \in U_{ij}} (A_{u,j} - \bar{A}_u)^2}}$$

Equação 6 - Similaridade do vetor cosseno ajustado

Pearson Correlation

Esta técnica mede o quanto duas variáveis se relacionam. Pode ser baseada em usuário ou item. Ela remove os efeitos da variância das avaliações dos usuários, medindo a linearidade entre duas variáveis [24]. A similaridade entre dois usuários u e v é dada pela equação 7:

$$w_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}},$$

Equação 7 - Correlação de Pearson entre usuários

Onde I representa o conjunto de itens que foram avaliados por u e v , enquanto \bar{r}_u representa a média das avaliações dos itens correlacionados do usuário u e $r_{u,i}$ representa a avaliação do usuário u para o item i . A similaridade entre dois itens i e j é dada pela equação 8:

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}},$$

Equação 8 - Correlação de Pearson entre itens

Onde U representa o conjunto de usuários que avaliaram os itens i e j , enquanto \bar{r}_i representa a média das avaliações do item i por todos os usuários e $r_{u,i}$ representa a avaliação do usuário u para o item i .

Algoritmos de Previsão

O cálculo da previsão é uma das partes mais importantes de um sistema de recomendação por filtragem colaborativa [31]. Neste projeto foram implementados os seguintes algoritmos de previsão:

Simple Weighted Average

É um algoritmo para predição baseada em item em que é usada uma simples média ponderada, como pode ser visualizado na equação 9:

$$P_{u,i} = \frac{\sum_{n \in N} r_{u,n} w_{i,n}}{\sum_{n \in N} |w_{i,n}|},$$

Equação 9 - Predição do algoritmo Simple Weighted Average

Onde N representa todos os itens avaliados pelo usuário u , $w_{i,n}$ representa a similaridade entre os itens i e n , e $r_{u,n}$ representa a avaliação do usuário u para o item n [31].

Weighted Sum of Others' Ratings

É um algoritmo para predição baseada em usuário. Para fazer uma predição da avaliação de um item i a um usuário u , é calculada a soma ponderada das avaliações dos outros usuários (todos do sistema ou um grupo dos mais próximos ou similares) a esse item i , conforme a equação 10:

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u) \cdot w_{a,u}}{\sum_{u \in U} |w_{a,u}|},$$

Equação 10 - Predição do algoritmo Weighted Sum of Others' Ratings

Onde $w_{a,u}$ é a similaridade entre os usuários a e u , \bar{r}_a e \bar{r}_u são, respectivamente, as médias das avaliações correlacionadas para os usuários a e u , U é o conjunto de usuários que avaliaram o item i e $r_{u,i}$ representa a avaliação do usuário u para o item i [31].

Slope One

É um algoritmo simples de filtragem colaborativa para previsão baseado em item. Ele funciona supondo que um usuário avaliou diversos itens com notas não binárias. Essas notas são colocadas em uma matriz de usuários por itens, de tal maneira que cada célula corresponda à nota dada a um item j por um usuário i . Se um usuário i não tiver dado notas a um item j , o elemento $A_{i,j}$ da matriz fica igual a 0. A partir dessa matriz, podem ser obtidas relações entre os dados, sendo possível, matematicamente, prever qual seria a nota dada por um usuário i a um item j ainda não avaliado por ele. Foi calculado com base na equação 11:

$$P(A,i) = \frac{(R(A,j) + Diff(i,j)) + (R(A,k) + Diff(i,k)) + \dots + (R(A,z) + Diff(i,z))}{N}$$

Equação 11 - Predição do algoritmo Slope One

Onde $R(A,j)$ representa a avaliação do usuário A ao item j , N representa a quantidade de itens da base de dados e $Diff(i, j)$ representa a diferença média entre as notas dadas aos itens i e j [27].

Algoritmos de Recomendação

Consiste em recomendar um conjunto de itens mais bem classificados (com maiores notas previstas) que serão de interesse para um determinado usuário. É um algoritmo que analisa a matriz usuário-item para descobrir as relações entre eles e recomendar os N itens melhor ranqueados (abordagem TOP-N) [31].

Vantagens e Desvantagens

As vantagens da filtragem colaborativa são, em geral, os pontos negativos na filtragem baseada em conteúdo. Por não analisar o conteúdo dos itens, é possível que sejam recomendados itens diversos e inesperados, o que costuma ser uma característica bem desejável nos sistemas de recomendação. Uma das principais desvantagens da filtragem colaborativa é que um usuário recém adicionado ao sistema ou que apresente um número muito pequeno de avaliações, faz com que o sistema não seja capaz de calcular precisamente a sua similaridade com os demais usuários. De maneira similar, um item recém adicionado ou que possua poucas avaliações não será recomendado até que seja avaliado por uma quantidade suficiente de usuários. Estes problemas anteriores são conhecidos como *cold start problem* [19, 22, 29]. Além disso pode ocorrer de usuários serem considerados similares para o sistema, mas não possuírem gostos similares para outros itens. Isso costuma ocorrer quando eles avaliaram alguns itens iguais, mas possuem poucas avaliações. Outro problema nesta técnica é que ela apresenta baixa precisão quando a matriz usuário x item, que armazena os valores das avaliações dos usuários aos itens, é esparsa, o que costuma ocorrer. Já o caso em que um usuário apresenta um gosto diferente do normal (ovelha negra), torna-se difícil o sistema de recomendação encontrar usuários similares, o que torna as recomendações pobres e não confiáveis. A escalabilidade e o custo computacional alto também são problemas desta abordagem, pois requerem a computação de milhares de itens e usuários, o que pode demandar um tempo de resposta inaceitável.

2.3.3 Filtragem Híbrida

As técnicas de filtragem de informação descritas anteriormente são fundamentais para os sistemas de recomendação. A partir delas é possível descobrir a relação existente entre diferentes itens e diferentes usuários e, quanto melhor forem estas técnicas, melhores serão as

predições e recomendações geradas ao usuário. Porém estas técnicas apresentam alguns problemas, já abordados anteriormente.

A filtragem híbrida é uma técnica que procura combinar as vantagens e também atenuar as desvantagens das abordagens anteriores [16]. Existem diferentes estratégias de filtragem híbrida [7, 10, 17, 34]. As principais, no que diz respeito à forma como as suas componentes serão combinadas para gerar as recomendações, são:

Ponderada: nesta abordagem a filtragem baseada em conteúdo e a filtragem colaborativa são implementadas separadamente e uma combinação linear é feita com os seus resultados. Pode ser necessária uma normalização nos resultados individuais, antes de aplicar a combinação linear, caso as técnicas gerem valores em escalas diferentes.

Mista: nesta abordagem as recomendações geradas pelas duas técnicas são combinadas no processo final de recomendação, de tal forma que as duas recomendações sejam apresentadas ao usuário na mesma lista.

Combinação sequencial: nesta abordagem a filtragem baseada em conteúdo cria os perfis dos usuários e, posteriormente, estes perfis são usados no cálculo da similaridade da filtragem colaborativa [4].

Comutação: nesta abordagem o sistema utiliza algum critério, como por exemplo a confiança no resultado, para comutar ou chavear entre a filtragem baseada em conteúdo e a filtragem colaborativa. Pode-se também realizara a comutação de uma técnica nos pontos de desvantagem da outra técnica.

2.4 Medidas de Avaliação de Desempenho

Já foi possível perceber o quão importante e útil é um sistema de recomendação nos dias de hoje. Mas quando se está desenvolvendo um, é importante ter a consciência de que eles devem recomendar itens que sejam relevantes ao usuário [11]. Para avaliar essa relevância das recomendações, diversas métricas têm sido utilizadas. Algumas delas serão abordadas a seguir.

As medidas de avaliação de desempenho para sistemas de recomendação são geralmente divididas em três categorias: as medidas de exatidão preditiva, as medidas de exatidão de classificação e as medidas de exatidão de rankings [8]. A primeira delas avalia o quão próximo dos valores reais são os valores previstos pelos sistemas de recomendação. A segunda avalia a frequência com a qual os sistemas de recomendação fazem recomendações corretas ou

incorretas. Por fim, a terceira avalia a corretude da ordem de recomendação dos itens [15]. Neste trabalho serão avaliadas apenas as medidas do primeiro e do segundo grupos.

2.4.1 Medidas de Exatidão Preditiva

Dentro do grupo de medida de exatidão preditiva encontram-se:

MAE

O erro médio absoluto mede o desvio médio entre as avaliações previstas e os valores reais de avaliação. Esta medida é definida conforme equação 12:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| = \frac{1}{n} \sum_{i=1}^n |e_i|$$

Equação 12 - MAE

Onde o numerador corresponde ao somatório do módulo do erro (diferença) entre os valores previsto e real de cada avaliação de um usuário para um item e o denominador n corresponde à quantidade de pares usuário x item possíveis, ou seja, à quantidade de células na matriz usuário x item.

MSE

O erro médio quadrático difere do MAE por punir grandes erros de uma forma mais severa, já que acrescenta ao erro total o quadrado da diferença entre o valor previsto e o valor real da avaliação. É definida conforme equação 13:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (f_i - y_i)^2 = \frac{1}{n} \sum_{i=1}^n (e_i)^2$$

Equação 13 - MSE

NMAE

Esta medida é uma variação do erro médio absoluto, normalizando o valor final do erro através dos limites inferior e superior dos valores das avaliações do sistema. É definida conforme equação 14:

$$\text{NMAE} = \frac{\text{MAE}}{x_{\max} - x_{\min}} = \frac{\sum_{i=1}^n |f_i - y_i|}{n(x_{\max} - x_{\min})} = \frac{\sum_{i=1}^n |e_i|}{n(x_{\max} - x_{\min})}$$

Equação 14 – NMAE

RMSE

Esta medida é uma variação do erro médio quadrático, correspondendo à sua raiz quadrada. É definida conforme equação 15:

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{\sum_{i=1}^n (f_i - y_i)^2}{n}} = \sqrt{\frac{\sum_{i=1}^n (e_i)^2}{n}}$$

Equação 15 - RMSE

Onde x_{\min} é o mínimo valor possível e x_{\max} é o máximo valor possível de uma avaliação no sistema.

Dentre as principais vantagens das medidas de exatidão preditiva, destacam-se o fato de serem mais fáceis de computar, de utilizarem propriedades estatísticas conhecidas, e calcularem a exatidão das predições atuais. Como desvantagens elas são muito específicas e sensíveis, principalmente em sistemas com pequenas escalas de avaliação [15].

2.4.2 Medidas de Exatidão de Classificação

As medidas de exatidão de classificação avaliam a frequência das decisões corretas e incorretas do sistema. O problema de classificação resume-se a recomendar ou não recomendar um determinado item a um usuário. Para isso é necessário converter as avaliações para uma escala binária, como por exemplo 0 para não recomendar e 1 para recomendar o item ao usuário, por considerar que ele não vá ser ou vá ser relevante, respectivamente. Dentre as principais medidas de classificação estão precisão, recall, f-measure, fallout e curva ROC [15].

Os dados utilizados neste sistema apresentam avaliações em uma escala de 1 a 5. Assim serão considerados relevantes ao usuário os itens com notas maiores ou iguais a 4. Conseqüentemente serão considerados irrelevantes os itens com notas inferiores a 4. Para o cálculo das medidas de classificação é necessário ainda que os itens sejam separados em dois conjuntos, um contendo os que foram recomendados para o usuário e outro contendo os que não foram recomendados.

Na tabela 1 a seguir pode ser visualizada a matriz de confusão, construída a partir destes conjuntos, e que oferece uma visualização fácil do número de classificações corretas e incorretas do sistema:

	Recomendados	Não recomendados	Total
Relevantes	VP (Verdadeiros positivos)	FN (Falsos negativos)	REL = VP + FN
Não relevantes	FP (Falsos positivos)	VN (Verdadeiros negativos)	NREL = FP + VN
Total	REC = VP + FP	NREC = FN + VN	N = REC + NREC = REL + NREL

Tabela 1 - Matriz de confusão

A seguir serão descritas as principais medidas de exatidão de classificação.

Precisão

A precisão está associada à habilidade de ordenar os itens mais relevantes nos primeiros lugares, e dessa forma corresponde à fração de todos os itens recomendados que são relevantes, ou seja, a quantidade de itens recomendados que são do interesse do usuário em relação ao conjunto de todos os itens que lhe são recomendados (equação 16).

$$\text{Precisão} = \frac{VP}{VP + FP} = \frac{VP}{REC}$$

Equação 16 - Precisão

Recall

O recall está associado à habilidade de recuperar todos os itens relevantes do sistema, e dessa forma corresponde à fração de todos os itens relevantes que foram recomendados, ou seja, indica a quantidade de itens de interesse do usuário que são recomendados (equação 17). Também é chamada de taxa de verdadeiros positivos.

$$\text{Recall} = \frac{VP}{VP + FN} = \frac{VP}{REL}$$

Equação 17 - Recall

F-measure

F-measure é uma combinação das medidas precisão e recall (equação 18).

$$\text{F-measure} = \frac{2 \cdot \text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}}$$

Equação 18 - F-measure

Fallout

Fallout ou taxa de falsos positivos corresponde à fração de todos os itens não relevantes que foram recomendados (equação 19).

$$\text{Fallout} = \frac{\text{FP}}{\text{FP} + \text{VN}} = \frac{\text{FP}}{\text{NREL}}$$

Equação 19 – Fallout

Curva ROC

Outra medida bastante comum para avaliar as classificações feitas por um sistema de recomendação é a área abaixo da curva ROC. A curva ROC é utilizada para medir o quanto um valor produzido por um sistema é capaz de distinguir os elementos relevantes dos não relevantes [32]. Ela é um gráfico que mostra o relacionamento entre benefício (taxa de verdadeiros positivos) e custo (taxa de falsos positivos), sensibilidade e especificidade, ou ainda sinal e ruído. Ela mostra que, para um classificador, a taxa de verdadeiros positivos não pode aumentar sem também aumentar a taxa de falsos positivos. Esta curva consegue dar uma indicação visual se um classificador é melhor ou pior do que outro, apenas comparando-se seus pontos na curva. A área sob a curva (AUC) é utilizada para resumir a performance do classificador em uma simples métrica. Assim, se a AUC é de 0.5, então isso significa que em 50% das vezes o sistema classificaria corretamente os elementos e em 50% das vezes classificaria de forma errada. Se os valores forem menores do que 0.5 o sistema está trocando os elementos relevantes pelos não relevantes. E quanto mais próximos de 1.0 estiverem os valores melhor é o classificador (figura 5).

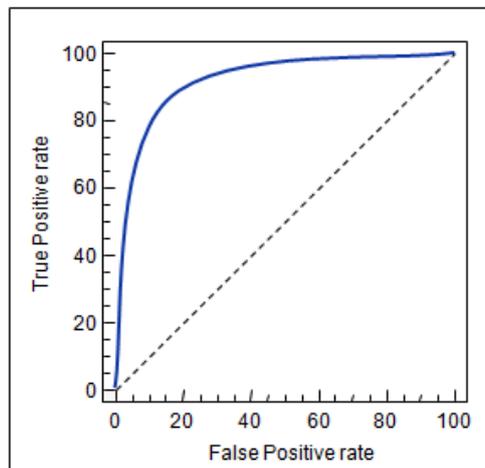


Figura 5 - Curva ROC

Dentre as principais vantagens das medidas de exatidão de classificação destacam-se: são apropriadas para avaliar a atual performance de um sistema de recomendação e são medidas bem estabelecidas. Dentre as desvantagens destaca-se o fato de ser difícil saber previamente se um item é ou não relevante para o usuário. Outra desvantagem é que estas medidas necessitam de um largo conjunto de dados para apresentar bons resultados.

2.4.3 Medidas de Exatidão de Ranking

As medidas de exatidão de ranking avaliam se a ordem da recomendação dos itens ao usuário está correta. Desde que a ordem esteja correta, os valores das predições não importam, e podem estar incorretos. Em outras palavras, ela mede a habilidade de o sistema de recomendação apresentar uma lista na qual os itens estão ordenados de acordo com as preferências do usuário [15]. Este grupo de medidas não foi utilizado na avaliação dos resultados do sistema implementado.

3. Análise da Literatura

O capítulo anterior descreveu aspectos relevantes sobre os sistemas de recomendação. Isso foi muito importante para o restante do trabalho, visto que é necessário ter esse conhecimento para propor soluções nesta área. O objetivo deste capítulo é mostrar trabalhos e soluções já propostos e desenvolvidos para esses sistemas, com as mais diversas abordagens. Isto é importante porque, para compreender a necessidade da técnica de filtragem híbrida, foco deste trabalho, é preciso antes entender os problemas apresentados por implementações de suas componentes.

3.1 Trabalhos Similares

A seguir são descritos alguns dos trabalhos similares que foram estudados antes da implementação do protótipo deste trabalho.

3.1.1 PHOAKS

O PHOAKS é um sistema de recomendação baseado em filtragem colaborativa que realiza a recomendação de recursos da Web a partir de informações contidas nas mensagens postadas pelos usuários da Usenet, que é um sistema Web que funciona como um quadro de avisos, em que qualquer pessoa pode comentar, enviar ou ler um artigo [23].

3.1.2 Sistema de Recomendação de Bibliotecas Digitais

O Sistema de Recomendação de Bibliotecas Digitais utiliza a filtragem baseada em conteúdo e foi desenvolvido para realizar a recomendação de artigos científicos na área da Ciência da Computação, armazenando-os em uma biblioteca digital. O conteúdo utilizado neste sistema é proveniente das informações contidas no Currículo Lattes dos usuários (formação acadêmica, produção bibliográfica) e na descrição dos artigos científicos.

3.1.3 SisRecCol

O SisRecCol é um sistema de recomendação que utiliza a técnica de filtragem colaborativa. Ele foi desenvolvido para apoiar o processo de aprendizagem colaborativa e para facilitar o acesso dos alunos aos materiais de aprendizado, para que possam aprofundar o seu conhecimento.

3.1.4 Fab

O Fab é um sistema de recomendação que utiliza as técnicas de filtragem colaborativa e baseada em conteúdo para recomendar páginas da Internet aos usuários. Ele faz parte do projeto da Biblioteca Digital da Universidade de Stanford. Sua arquitetura possui basicamente dois agentes: o agente de coleta e o agente de seleção. O agente de coleta é responsável por pesquisar páginas Web e indexá-las. O agente de seleção é responsável por selecionar, a partir dos interesses do usuário, quais páginas lhe devem ser recomendadas. Após a recomendação o usuário deve avaliar esta recomendação, a fim de que o agente de seleção atualize o seu perfil [4]. No Fab, o perfil contendo as páginas de interesse do usuário é construído a partir de filtragem baseada em conteúdo. Assim, os usuários mais similares a um usuário alvo são determinados a partir da semelhança entre os perfis dos mesmos. Com os usuários mais similares são feitas as recomendações a partir de técnicas de filtragem colaborativa puras. Por utilizar um modelo aprendido por uma técnica baseada em conteúdo como entrada para uma técnica colaborativa, esse sistema é considerado um sistema híbrido de combinação sequencial.

3.1.5 FEERS

O FEERS foi desenvolvido na Universidade Federal de Pernambuco, a fim de recomendar filmes para os usuários a partir de suas avaliações. Esse sistema também utiliza um modelo aprendido por uma técnica de filtragem baseada em conteúdo como entrada para uma técnica de filtragem colaborativa, constituindo assim uma técnica híbrida de combinação sequencial [13].

3.1.6 e-Recommender

O e-Recommender também foi desenvolvido na Universidade de Pernambuco, a partir de um trabalho de conclusão de curso e baseado-se no FEERS, utilizando inclusive os mesmos algoritmos. Ele recomenda produtos de uma loja de comércio eletrônico a partir dos produtos já comprados pelos usuários. A diferença entre o e-Recommender e o FEERS é que o primeiro não considera as avaliações feitas pelos usuários, e sim as compras que foram realizadas por eles [13].

3.1.7 P-Tango

O P-Tango foi desenvolvido no Instituto Politécnico de Worcester, e tem como objetivo a recomendação de notícias em um jornal on-line. Ele é um sistema de recomendação que utiliza a filtragem híbrida ponderada, a partir da média ponderada entre as recomendações de notícias feitas pela filtragem colaborativa e pela filtragem baseada em conteúdo. Nesse sistema o perfil do usuário é formado por palavras-chave, sendo estas ou fornecidas pelo usuário, ou geradas implicitamente a partir de artigos avaliados explicitamente e positivamente. Os pesos dados a cada componente são ajustados à medida em que as recomendações vão sendo realizadas.

3.2 Análise e Considerações

Além destes sistemas, existem diversos outros sistemas de recomendação desenvolvidos, sendo que a maioria deles utiliza a técnica de filtragem colaborativa, por ser um método simples de o usuário avaliar os itens, e por não depender de nenhuma informação além das avaliações.

A partir da análise destes e de outros sistemas, percebeu-se que só é viável criar um sistema que utilize a filtragem baseada em conteúdo se os itens a serem recomendados possuírem metadados que os descrevam. A partir desses metadados torna-se fácil identificar similaridade entre um item e os interesses do usuário. Como o sistema proposto neste trabalho continha metadados e também avaliações de usuários a itens, decidiu-se combinar as duas técnicas anteriores. Desta forma o sistema não será simplesmente de avaliação de itens, como também de análise do conteúdo dos itens.

4. Especificação e Implementação do Protótipo

Neste capítulo são apresentadas as tecnologias e a estrutura utilizadas no protótipo de sistema de recomendação que foi desenvolvido, com o objetivo de consolidar e validar o que foi proposto neste trabalho. O estudo de caso escolhido para este trabalho foi o de um domínio de recomendação de livros digitais. As informações destes livros, dos usuários, dos autores, das categorias e das avaliações do sistema foram extraídas do site da Amazon [2], através da implementação de um Crawler. Estas informações foram extraídas apenas para uma carga inicial do sistema, o qual permite a criação de novos itens, usuários e avaliações, atualizando os valores das recomendações previstas em tempo real. Esta carga inicial do sistema continha 109 usuários, 96 livros digitais e 387 avaliações de usuários a itens, sendo que todos os itens do sistema receberam pelo menos uma avaliação, e a escala de avaliação do sistema varia de 1 a 5. A única informação que o sistema tem dos usuários são as avaliações que eles deram aos itens. O usuário não provê mais nenhuma informação a seu respeito. Acredita-se que isso seja bom, por ser não intrusivo, ou seja, não requerer que o usuário preencha questionários, por exemplo.

A partir do conhecimento obtido com os trabalhos similares e as pesquisas sobre sistemas de recomendação, resolveu-se adotar, como estratégia de recomendação, a estratégia de listas ordenadas de itens, em ordem decrescente de valores de notas previstos. Além disso foram definidos diversos parâmetros e algoritmos a serem utilizados.

4.1 Tecnologias Utilizadas

No desenvolvimento do protótipo deste sistema foi utilizada a linguagem de programação C# através do Visual Studio 2012. Devido à grande quantidade de dados envolvidos no sistema, foi utilizado o sistema gerenciador de banco de dados SQL Server Management Studio 2012 para armazenar os dados, evitando a necessidade de eles ficarem todo o tempo na memória, e para poder acessar estes mesmos dados posteriormente. Foram analisados diversos sistemas de gerenciamento, mas o SQL Server foi o escolhido por ter um bom relacionamento com a ferramenta de desenvolvimento escolhida. O computador utilizado para o desenvolvimento deste protótipo e para os experimentos nele realizados contém 4GB de RAM e 2.66GHz de processador.

4.2 Arquitetura

Na figura 6 pode ser visualizado o esboço da arquitetura do sistema desenvolvido.

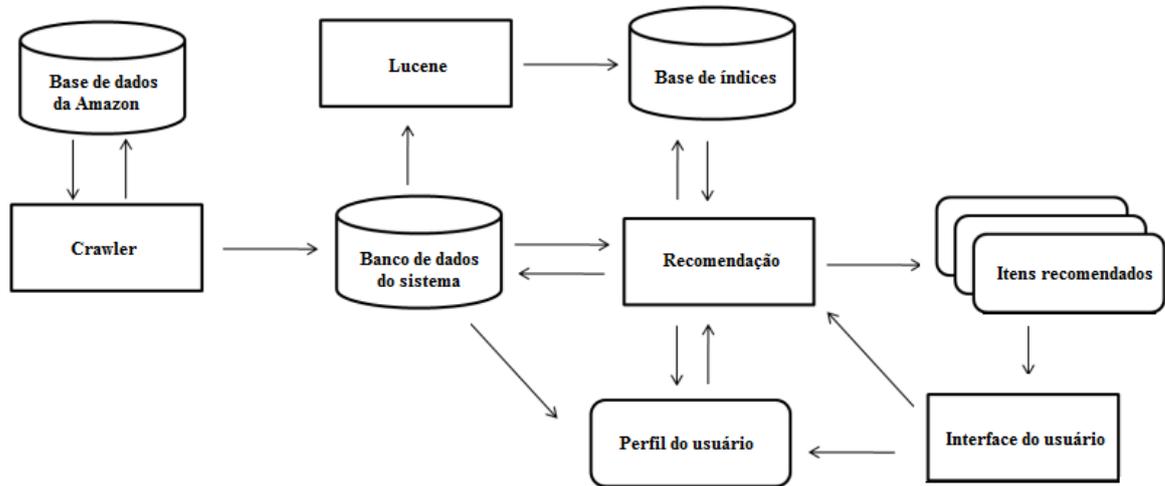


Figura 6 - Esboço da arquitetura do sistema desenvolvido

Inicialmente o Crawler extrai as informações da base de dados da Amazon e as insere no banco de dados do sistema. Quando o usuário solicita as recomendações de itens através da interface, o sistema realiza os algoritmos de predição e recomendação a partir do perfil deste usuário (conteúdo dos itens já avaliados por ele se a filtragem for baseada em conteúdo, valores das avaliações se a filtragem for colaborativa, e ambos se a filtragem for híbrida), da base de índices gerada pelo Lucene (se a filtragem for baseada em conteúdo ou híbrida), e do banco de dados do sistema, do qual recupera os itens ainda não avaliados pelo usuário para predição, bem como os perfis dos demais usuários no caso da filtragem ser colaborativa ou híbrida. Após as predições e recomendações serem realizadas, uma lista com os itens recomendados é apresentada ao usuário na interface do sistema.

4.3 Modelagem do Banco de Dados

Na figura 7 pode ser visualizado o diagrama entidade relacionamento do banco de dados utilizado para armazenar as informações relevantes do sistema desenvolvido.

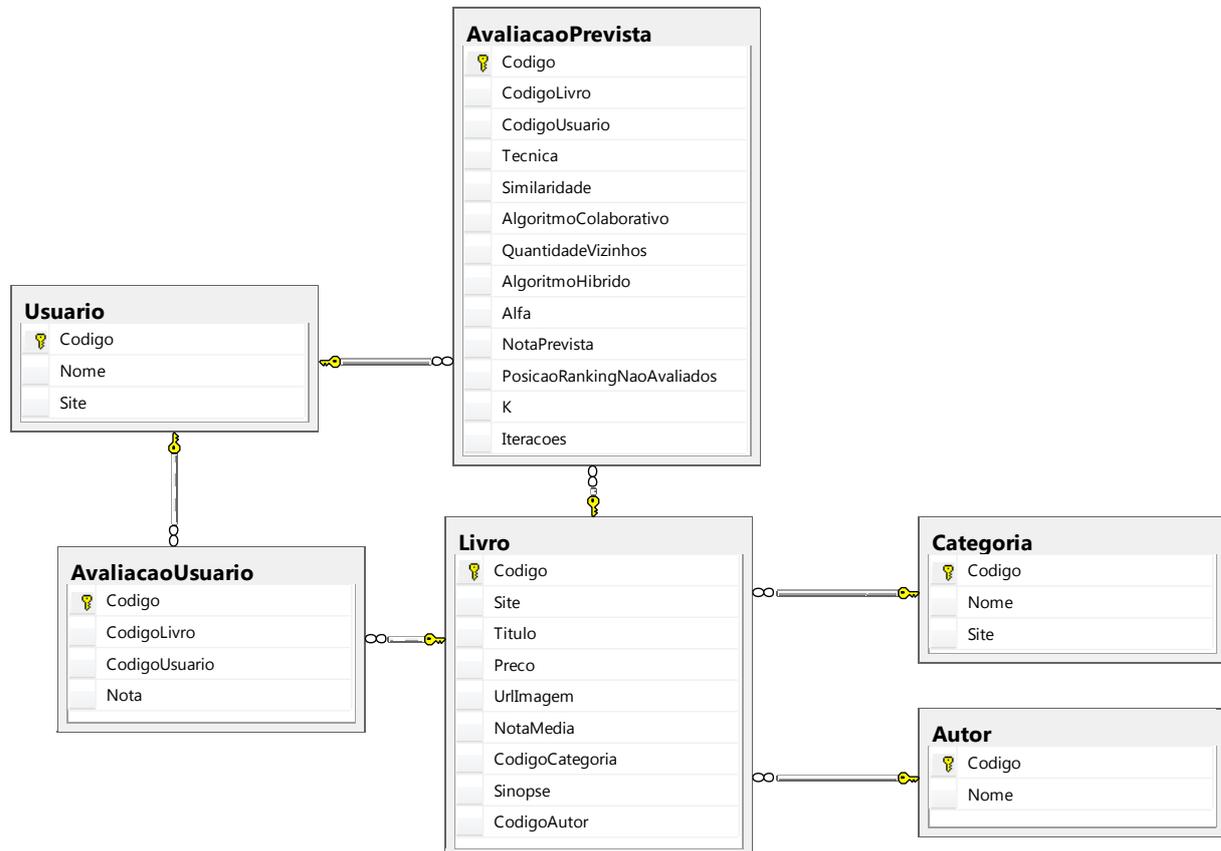


Figura 7 - Diagrama entidade relacionamento do sistema desenvolvido

Como pode ser visto são seis as tabelas desse diagrama, sendo armazenadas as informações sobre os usuários, livros, categorias de livros, autores dos livros, avaliações feitas pelos usuários e previsão das avaliações a serem feitas pelos usuários. A tabela AvaliacaoPrevista foi criada no banco para, toda vez que uma previsão de nota de um usuário para um item fosse realizada, a mesma fosse salva no banco de dados, a fim de que na próxima vez que essa informação fosse necessária ela pudesse ser retornada mais rapidamente, sem que fosse necessário mantê-la na memória. Ela contém os diversos parâmetros que podem ser variados no sistema, tais como algoritmo colaborativo, algoritmo híbrido, número de centroides K do algoritmo de clustering, quantidade de vizinhos dos algoritmos colaborativos KNN, etc.

4.4 Algoritmos Implementados

Os algoritmos de sistemas de recomendação, no geral, não são simples de serem implementados, e requerem uma quantidade significativa de dados para que se aumente a precisão na recomendação. É na precisão da recomendação que se encontra o maior desafio para estes sistemas: fazer a melhor combinação possível entre o que está sendo recomendado e aqueles que estão procurando por recomendações.

Neste trabalho foram escolhidas as técnicas baseadas em conteúdo, filtragem colaborativa e filtragem híbrida, por serem as mais comuns e as mais estudadas. Quanto à técnica baseada em conteúdo foi implementado o TF-IDF, com a ajuda do framework do Lucene. Quanto à técnica da filtragem colaborativa foram implementados algoritmos baseados em memória e algoritmos de similaridade. Já as técnicas de filtragem híbrida escolhidas foram a ponderada e uma combinada a uma recomendação não personalizada, de acordo com as posições dos itens no ranking de recomendação.

Diversos testes foram realizados para validar as implementações dos algoritmos, e foi possível perceber que eles funcionaram satisfatoriamente. Por exemplo, para os algoritmos de similaridade, verificou-se que a similaridade entre um usuário e ele mesmo é sempre 1. Já a similaridade entre um usuário e outro que não avaliou nenhum item em comum com ele é sempre 0.

4.5 Lucene

Lucene.Net é uma biblioteca framework de alto desempenho voltada para Recuperação de Informação, que contém poderosas APIs para indexação de qualquer coisa que possa ser representada como texto e para implementação de tecnologias avançadas de busca. Ela é um API do projeto Apache Lucene, desenvolvido originalmente em Java. O Lucene.Net foi utilizado para o cálculo das recomendações da filtragem baseada em conteúdo, já que ela já utiliza o TF-IDF como cálculo de similaridade entre os vetores representativos dos textos e da consulta, para ordenação da base de índices. Esta consulta normalmente é formada a partir da concatenação dos valores dos atributos dos itens que o usuário já avaliou.

Esta biblioteca inclui diversas etapas. Durante a etapa de preparação dos documentos, o Lucene seleciona quais os termos do documento que melhor o descrevem, reduzindo a complexidade de representação do mesmo, já que os termos que não são muito significativos

para representar a semântica do documento são removidos. Na análise léxica, o Lucene remove as pontuações, símbolos e caracteres especiais. Na etapa de remoção de stopwords e stemming (redução da palavra ao seu radical) foi utilizada a classe SnowballAnalyzer, escolhendo uma lista de stopwords em português a serem removidas dos documentos. A etapa de identificação de grupos nominais não foi utilizada no escopo deste projeto.

4.6 Perfil do Usuário

Quanto maior o número de itens avaliados por um usuário maior será a memória necessária para armazenar o seu perfil. A partir dos estudos realizados foi proposta uma mudança na representação interna do perfil dos usuários, reduzindo a quantidade de itens já avaliados nele presentes, de forma a mantê-lo o mais informativo possível (Wilson e Martinez 2000). O sistema desenvolvido dá a possibilidade de o usuário escolher se seu perfil será formado por todos os itens que ele já avaliou ou apenas por alguns itens que melhor o representem. Para o segundo caso foi necessário descobrir automaticamente quais as regiões de interesse dos usuários, classificando os itens avaliados nestas regiões. Para isso foi implementado um algoritmo de clustering, uma das técnicas mais utilizadas na área de Aprendizagem de Máquina. Esta técnica consiste em um aprendizado não supervisionado já que, durante o seu treinamento, não possui nenhuma referência para a classificação dos dados [33].

O algoritmo desenvolvido neste sistema é bastante similar ao algoritmo K-means, que é um dos mais tradicionais algoritmos de clustering, além de também ser de fácil entendimento e implementação. Desta forma, seu funcionamento se dá pela melhor definição dos K centroides que melhor representem os dados do sistema. Enquanto no K-means tradicional são inclusos todos os dados, no algoritmo implementado neste projeto são utilizados apenas os itens já avaliados por um usuário. A definição dos centroides costuma ser feita através do cálculo de uma função de distância (normalmente a euclidiana) entre os pontos [32]. Neste projeto foi utilizado o TF-IDF como distância ou similaridade entre os itens. Basicamente os passos do algoritmo K-means tradicional são os seguintes:

- Selecionar uma partição inicial com K conjuntos aglomerados;
- Gerar uma nova partição associando cada item ao conjunto de centroide mais próximo, de acordo com a função de distância;

- Calcular os novos valores de centroide;
- Repetir os dois últimos passos até que haja estabilização dos valores ou até que o número máximo de iterações seja alcançado;
- Retornar a última configuração de centroides.

Na variação do K-means que foi implementada, os passos são os seguintes:

- Selecionar uma partição inicial com K conjuntos aglomerados;
- Gerar uma nova partição associando cada item ao conjunto de centroide mais próximo, de acordo com a similaridade calculada pelo TF-IDF;
- Calcular os novos valores de centroide;
- Repetir os dois últimos passos até que o número máximo de iterações seja alcançado;
- Retornar a configuração de centroides que melhor distribuiu os itens entre os clusters. Esta configuração foi determinada como sendo a configuração que maximizou o produto entre os pesos dos clusters, formados pela soma das similaridades entre um item e o centróide de seu cluster. A melhor distribuição dos itens nos clusters é o objetivo porque assim os interesses do usuário ficam bem definidos. Não seria interessante que todos os itens estivessem em apenas um cluster. O ideal é representar mais do que um interesse do usuário. Esta melhor distribuição ocorre quando o produto entre os pesos dos clusters é máximo, porque o produto entre números reais positivos é máximo quando estes números são iguais (ver Apêndice).

Como principal desvantagem estes algoritmos devem definir previamente o número K de clusters, pois não se sabe quantos conjuntos serão precisos para definir as regiões de interesse dos usuários. A escolha de um valor muito alto ou muito baixo para o parâmetro K poderia levar o sistema a tomar decisões equivocadas. Outra desvantagem, a depender da quantidade máxima de iterações, é o problema da escalabilidade, já que é necessário calcular diversas vezes a função de distância ou similaridade até que haja estabilização dos valores dos centroides. Além disso é muito sensível à escolha da configuração inicial dos centroides, já que é um algoritmo determinístico, apresentando sempre o mesmo resultado a partir de uma mesma configuração inicial, se o número de iterações for o suficiente.

4.7 Recomendações Híbridas

O algoritmo se inicia calculando a similaridade entre os itens ou entre os usuários, prosseguindo com o cálculo das recomendações colaborativas para cada usuário, a partir das similaridades. Em seguida, e onde o algoritmo começa a diferir dos tradicionais, há a geração dos clusters que representam as regiões de interesse dos usuários. A partir destes clusters é calculada a filtragem baseada em conteúdo, considerando os itens não avaliados e todos os itens avaliados ou apenas os centroides dos clusters. Por fim é feita a recomendação híbrida, a partir das recomendações colaborativa e baseada em conteúdo, e a depender da abordagem utilizada. Estas abordagens são descritas a seguir.

4.7.1 Recomendação Híbrida Ponderada

Como já foi abordado, a recomendação híbrida ponderada se baseia em uma combinação linear dos valores da recomendação baseada em conteúdo com a recomendação colaborativa. Essa abordagem foi proposta inicialmente em 1999, no sistema de jornal on-line P-Tango. Escolheu-se abordar esta abordagem híbrida porque ela permite analisar individualmente cada uma das duas técnicas, a partir de um simples ajuste dos pesos [9]. Porém, como as técnicas estavam em escalas diferentes, seus resultados precisaram ser normalizados. Para isso eles foram divididos por seu valor máximo, deixando-os limitados entre 0 e 1. Em seguida, como as notas deveriam estar na mesma faixa das notas do sistema, com valor máximo de 5, os resultados da normalização foram multiplicados por 5.

O parâmetro α da combinação linear é ajustado para variar a importância de cada técnica no resultado final da recomendação híbrida. Quando ele é igual a 0 ou muito baixo, indica que a recomendação colaborativa está sendo desprezada ou tendo sua importância minimizada. Quando ele é igual a 1 indica que a recomendação baseada em conteúdo é que está sendo desprezada ou tendo sua importância minimizada. A seguir, na equação 20, pode ser visualizada a fórmula dessa abordagem híbrida.

$$(1 - \alpha).Valor_{FBC} + \alpha.Valor_{FC}$$

Equação 20 - Combinação linear da filtragem híbrida ponderada

4.7.2 Recomendação Híbrida pela Soma dos Inversos das Posições

Nesta técnica de filtragem híbrida forma-se um ranking em ordem decrescente de nota prevista para os algoritmos de filtragem baseada em conteúdo (FBC), filtragem colaborativa (FC) e recomendação não personalizada por quantidade de avaliações (QA). Para cada um dos algoritmos guarda-se a posição em que cada item apareceu no ranking. Desta forma a nota híbrida prevista e atribuída a cada um dos itens ainda não avaliados pelo usuário, é calculada como na equação 21:

$$\text{NotaHíbrida} = \frac{1}{\text{PosFBC}} + \frac{1}{\text{PosFC}} + \frac{1}{\text{PosQA}}$$

Equação 21 - Filtragem híbrida com heurística da quantidade de avaliações

Esta técnica de recomendação híbrida foi escolhida para possibilitar recomendações de itens que foram bem avaliados por usuários similares, que possuem características semelhantes com os itens já avaliados pelo usuário e que também sejam populares, ou seja, avaliados por muitos usuários. Se algum dos itens não for recomendado ou receber nota zero em algum dos algoritmos, a parcela correspondente é anulada. Assim quando um item novo for inserido no sistema, por ainda não possuir avaliações, as duas últimas parcelas desta soma são anuladas, considerando-se apenas a filtragem baseada em conteúdo. Já quando um usuário novo for inserido no sistema, por ainda não ter avaliado nenhum item e assim não ser possível saber quais são as suas preferências, considera-se apenas a última parcela da soma, recomendando para eles os itens mais avaliados do sistema, desconsiderando assim a filtragem baseada em conteúdo e a filtragem colaborativa.

4.8 Descrição do Protótipo

O sistema desenvolvido para teste e avaliação das técnicas de recomendação inicia-se com uma tela de Login. Nesta tela de Login existe a possibilidade de criar um novo usuário ou escolher um já existente (figura 8).

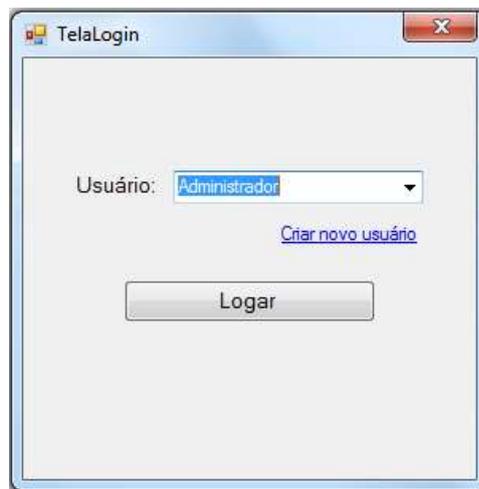


Figura 8 - Tela de login

Se nessa tela for escolhido o usuário administrador, haverá acesso a uma tela (figura 9) que permite acesso às telas de configuração das informações referentes a usuários (figura 10), categorias (figura 11), autores (figura 12), livros (figura 13), avaliações de usuários (figura 14) e avaliações já previstas pelo sistema (figura 15). Estas telas funcionam como um sistema de gerenciamento das informações que estão no banco de dados, podendo alterá-las, removê-las ou cadastrar novas informações.

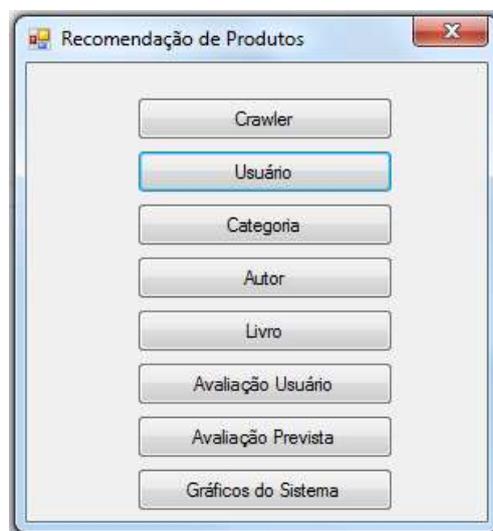


Figura 9 - Tela do administrador

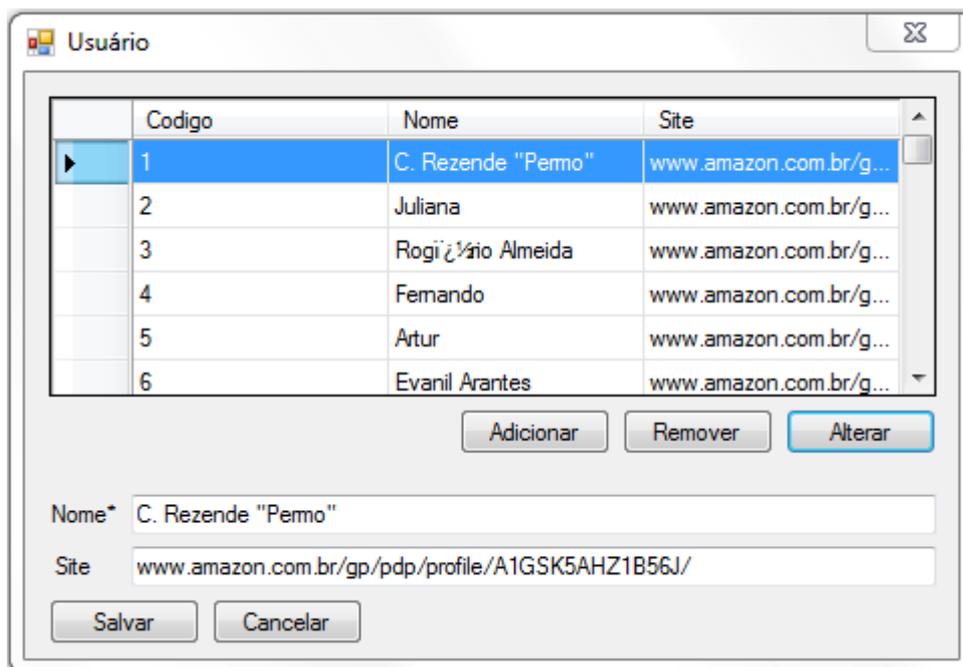


Figura 10 - Tela de configuração de usuários

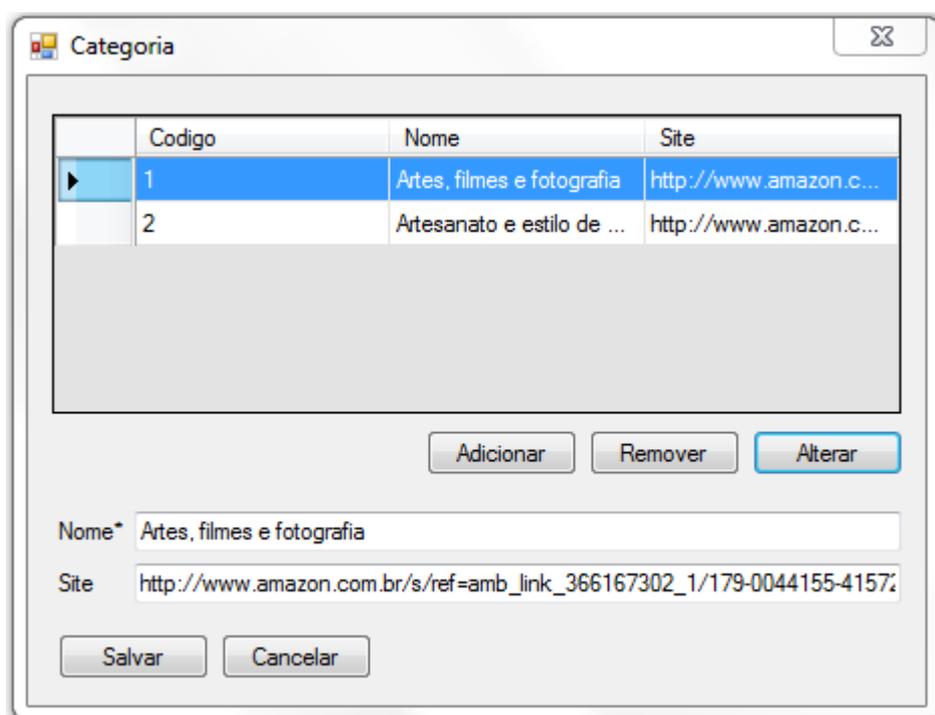


Figura 11 - Tela de configuração de categorias

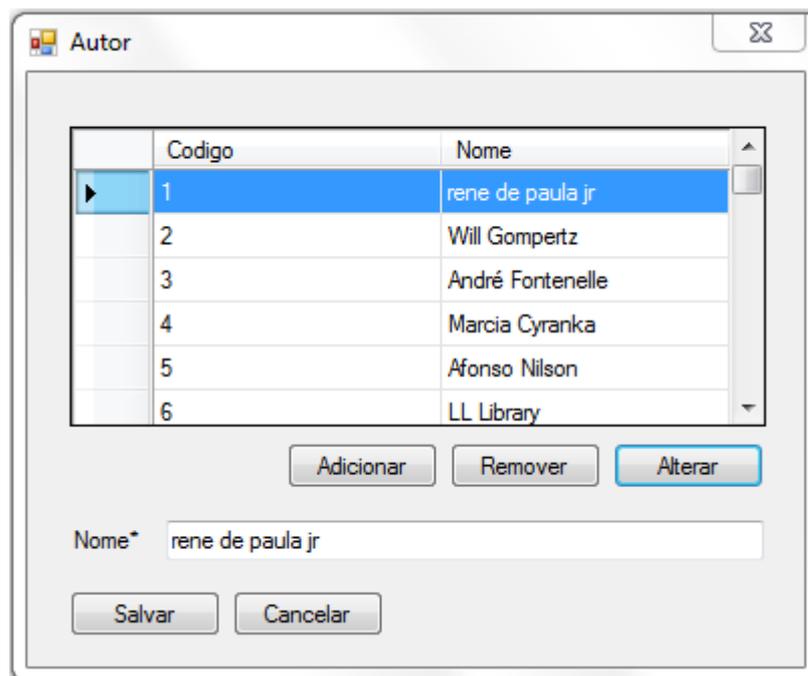


Figura 12 - Tela de configuração de autores

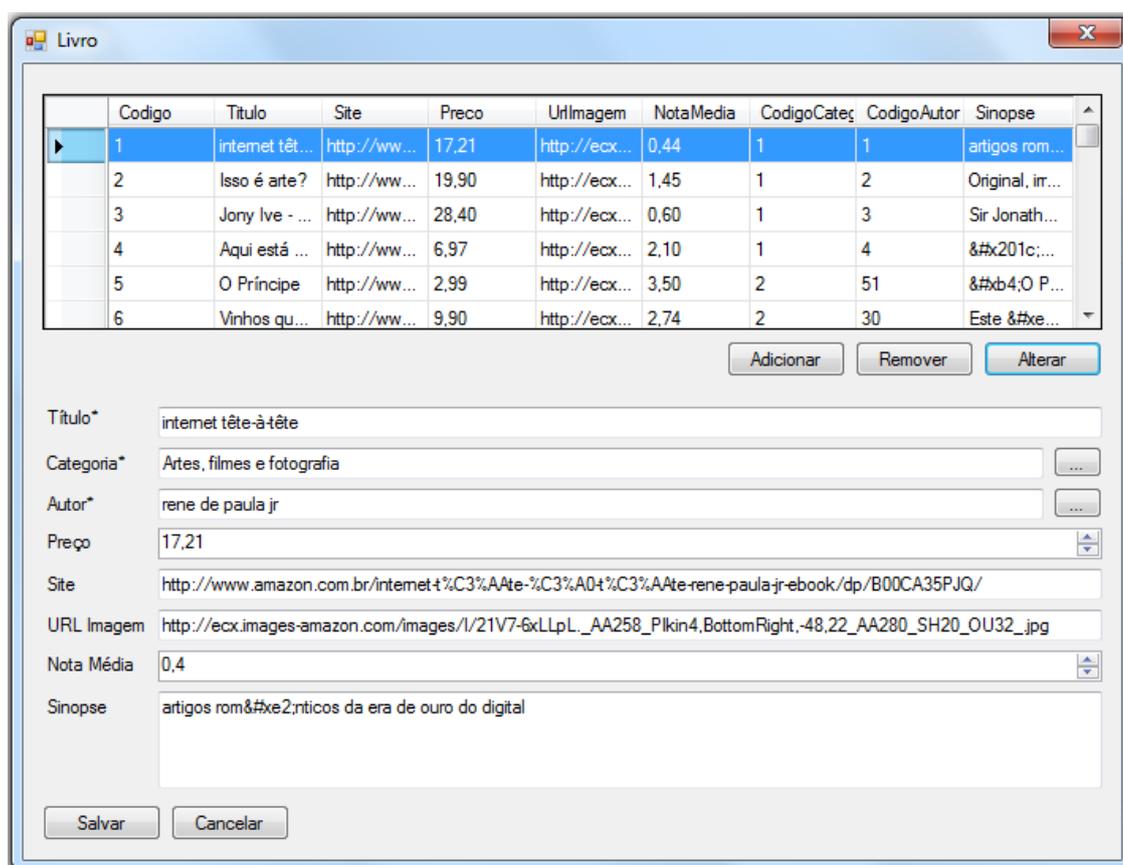


Figura 13 - Tela de configuração de livros

Avaliação Usuário

	Codigo	CodigoLivro	CodigoUsuario	Nota
▶	1	4	1	4,00
	2	5	1	5,00
	3	6	52	4,00
	4	7	52	4,00
	5	8	2	5,00
	6	9	2	5,00

Usuário* ...

Livro* ...

Nota

Figura 14 - Tela de configuração de avaliações de usuários

Avaliação Prevista

	Codigo	CodigoL	CodigoL	Tecnica	Algoritm	Similarid	Algoritm	Quantid	Alfa	NotaPre	Posicao	K	Iteracao
▶	1	64	24	0						5,0000	1	4	1
	2	64	14	0						2,5553	2	4	1
	3	64	22	0						2,5364	3	4	1
	4	64	33	0						2,1016	4	4	1
	5	64	53	0						2,0471	5	4	1
	6	64	31	0						1,9852	6	4	1

Usuário* ...

Livro* ...

Técnica*

Algoritmo Colaborativo

Similaridade

Algoritmo Híbrido

Vizinhos Alfa K Iterações

Nota Prevista*

Posição Ranking

Figura 15 - Tela de configuração de avaliações previstas

O administrador pode também iniciar o Crawler, que roda sobre o site da Amazon [2] coletando as informações necessárias e salvando-as no banco de dados do sistema. Também é possível visualizar gráficos referentes aos dados do sistema, tais como:

- Distribuição do número de avaliações por nota (figura 16)

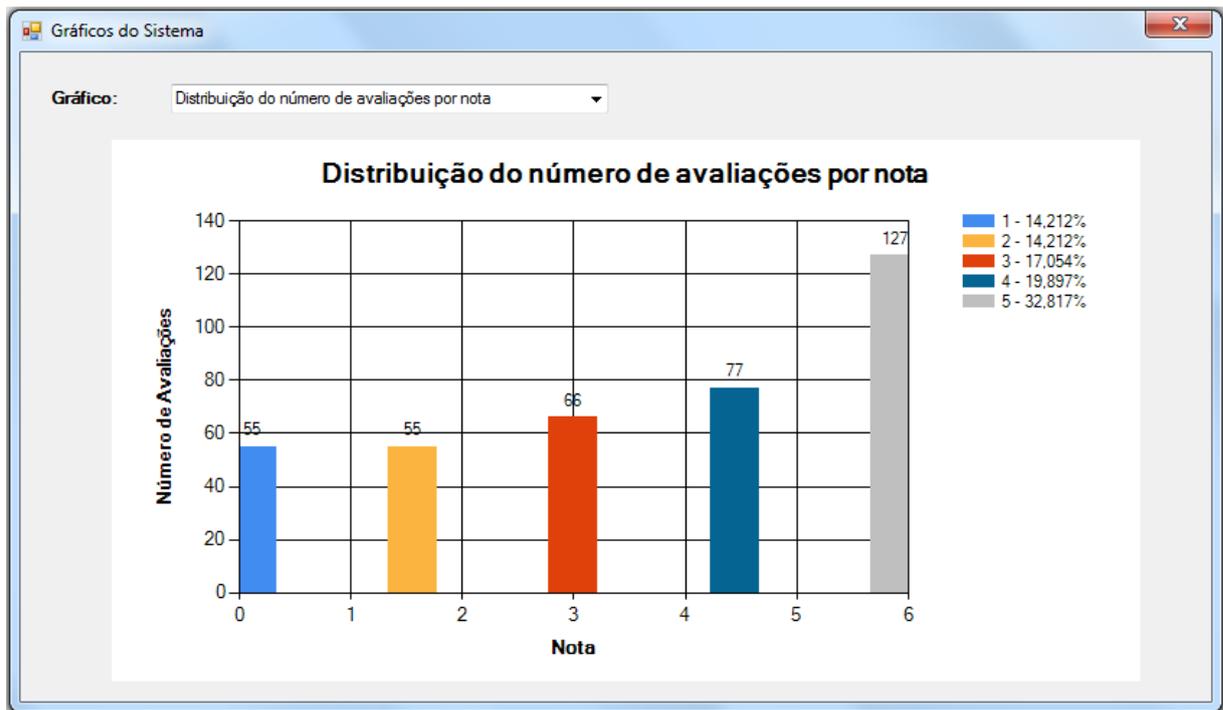


Figura 16 - Distribuição do número de avaliações por nota

É interessante observar também a distribuição de avaliações feitas pelos usuários. A escala de avaliação do sistema vai de 1 a 5 e foi possível perceber que quase 33% das avaliações foram iguais a 5, quase 20% iguais a 4, 17% iguais a 3, 14% iguais a 2 e 14% iguais a 1. A média das avaliações foi igual a 3.41, o que mostra que a maioria dos usuários avaliou os itens com notas altas.

- Distribuição do número de itens por número de avaliações (figura 17)

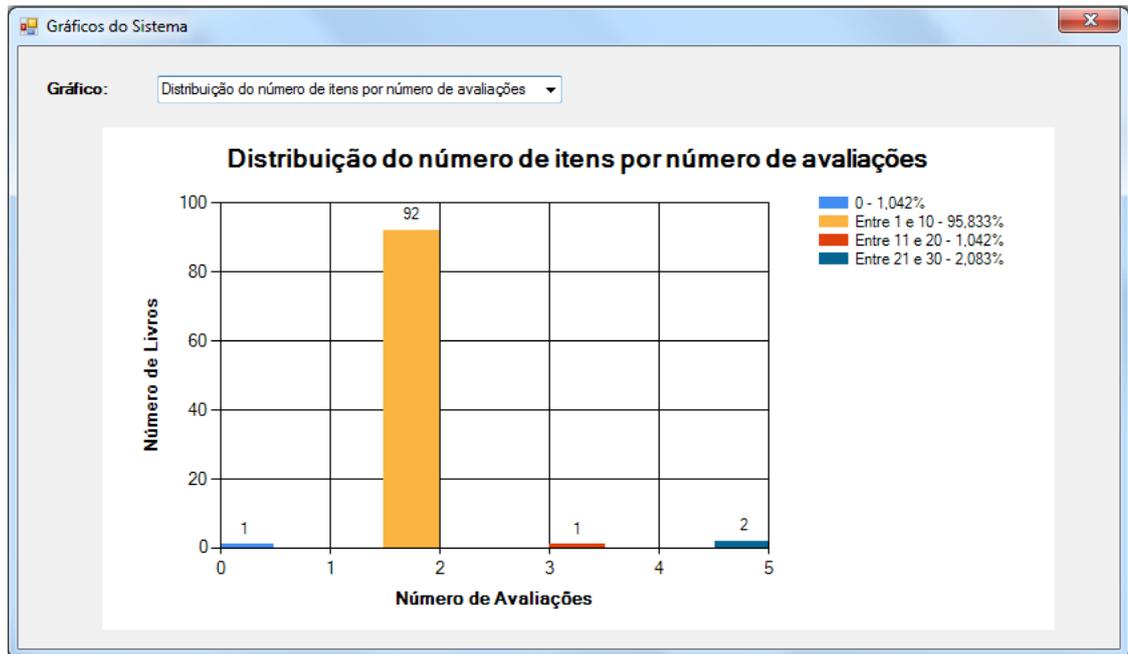


Figura 17 - Distribuição do número de itens por número de avaliações

Na figura anterior é possível perceber que apenas um dos itens do sistema não recebeu nenhuma avaliação e que o maior grupo dos itens recebeu de um a dez avaliações, representando cerca de 96% do total.

- Distribuição do número de usuários por número de avaliações (figura 18)



Figura 18 - Distribuição do número de usuários por número de avaliações

Na figura anterior é possível perceber que todos os usuário avaliaram pelo menos um item e que a maioria deles avaliou entre um e dez itens, representando pouco mais de 98% do total.

- Gráfico de esparsidade do sistema (figura 19)



Figura 19 - Gráfico de esparsidade do sistema

A esparsidade é um problema muito comum nas grandes lojas de comércio eletrônico e em sistemas de recomendação com conjuntos de dados muito grandes, tais como a Amazon, da qual os dados de entrada para este trabalho foram retirados. Em geral, menos de 1% dos itens disponíveis são avaliados pelos usuários, o que costuma diminuir a precisão das recomendações feitas pelos sistemas que utilizam estas informações. Vale ressaltar que a ocorrência de uma grande quantidade de usuários no sistema não representa um problema de esparsidade, já que quanto mais usuários para avaliar os itens do sistema melhores poderão ser as recomendações na filtragem colaborativa, uma vez que haverá mais usuários similares ao usuário alvo da recomendação. Para calcular o nível de esparsidade de um determinado conjunto de dados, pode ser utilizada a equação 22:

$$1 - \frac{\text{QuantidadeDeAvaliações Realizadas}}{\text{QuantidadeItens} * \text{QuantidadeUsuários}}$$

Equação 22 – Fórmula do nível de esparsidade

Na base de dados utilizada neste projeto, o nível de esparsidade é igual a 96,302% (equação 23):

$$1 - \frac{387}{(96 * 109)} = 0.96302$$

Equação 23 - Nível de esparsidade do sistema

Caso seja escolhido um usuário diferente do administrador, o usuário é redirecionado para a tela dos itens do sistema (figura 20). Esta ainda não é a tela de recomendação. É apenas a tela em que todos os itens do sistema são listados, com suas informações básicas (título, categoria, autor, sinopse, site do item na Amazon, preço, nota média e valor da avaliação dada pelo usuário logado). Se o item ainda não recebeu nenhuma avaliação, a nota média aparece como "Não definida". De forma similar, se o usuário logado ainda não avaliou um item, o valor da avaliação dada pelo usuário logado para este item também aparece como "Não definida".

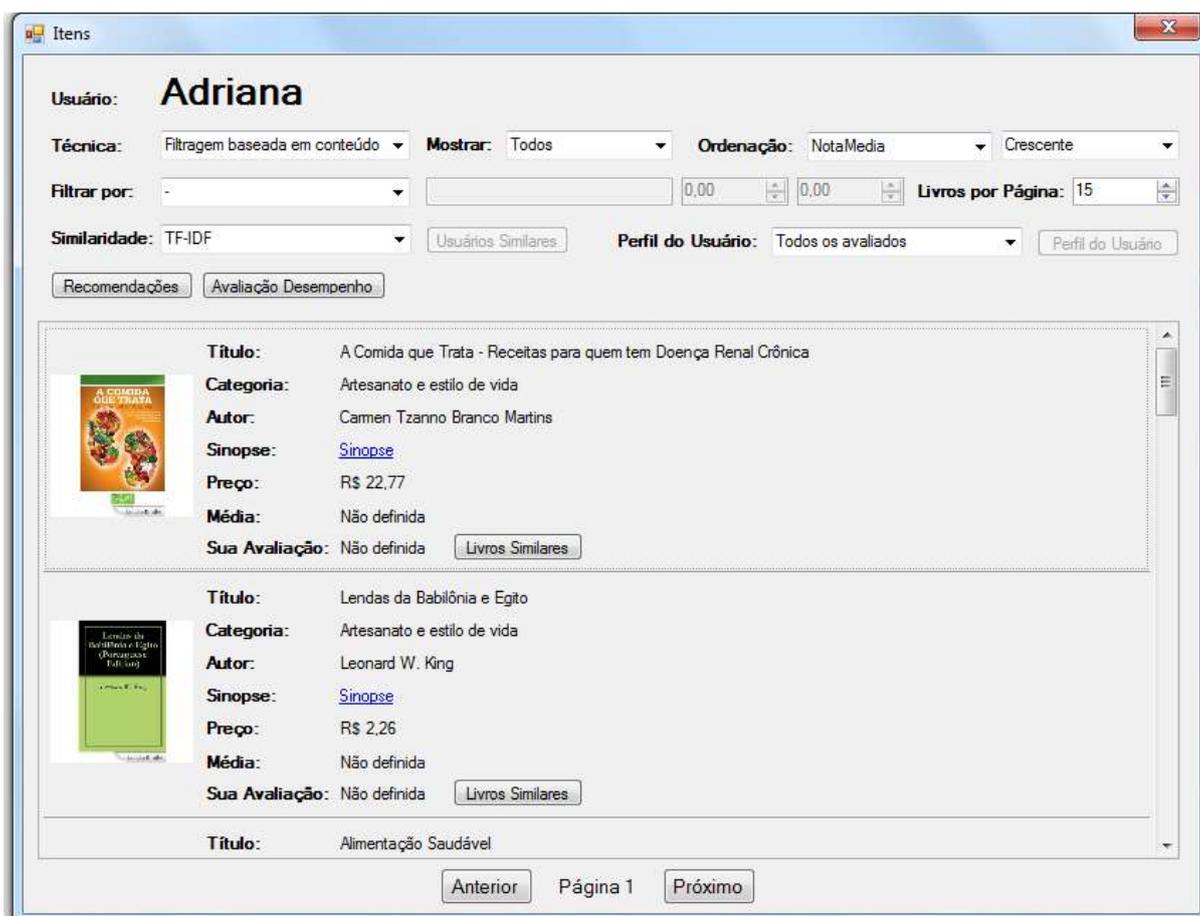


Figura 20 - Tela dos itens do sistema

Adicionalmente, na tela dos itens é possível ainda:

- Exibir apenas os itens já avaliados pelo usuário logado, apenas os itens ainda não avaliados pelo usuário logado, ou ambos (figura 21);

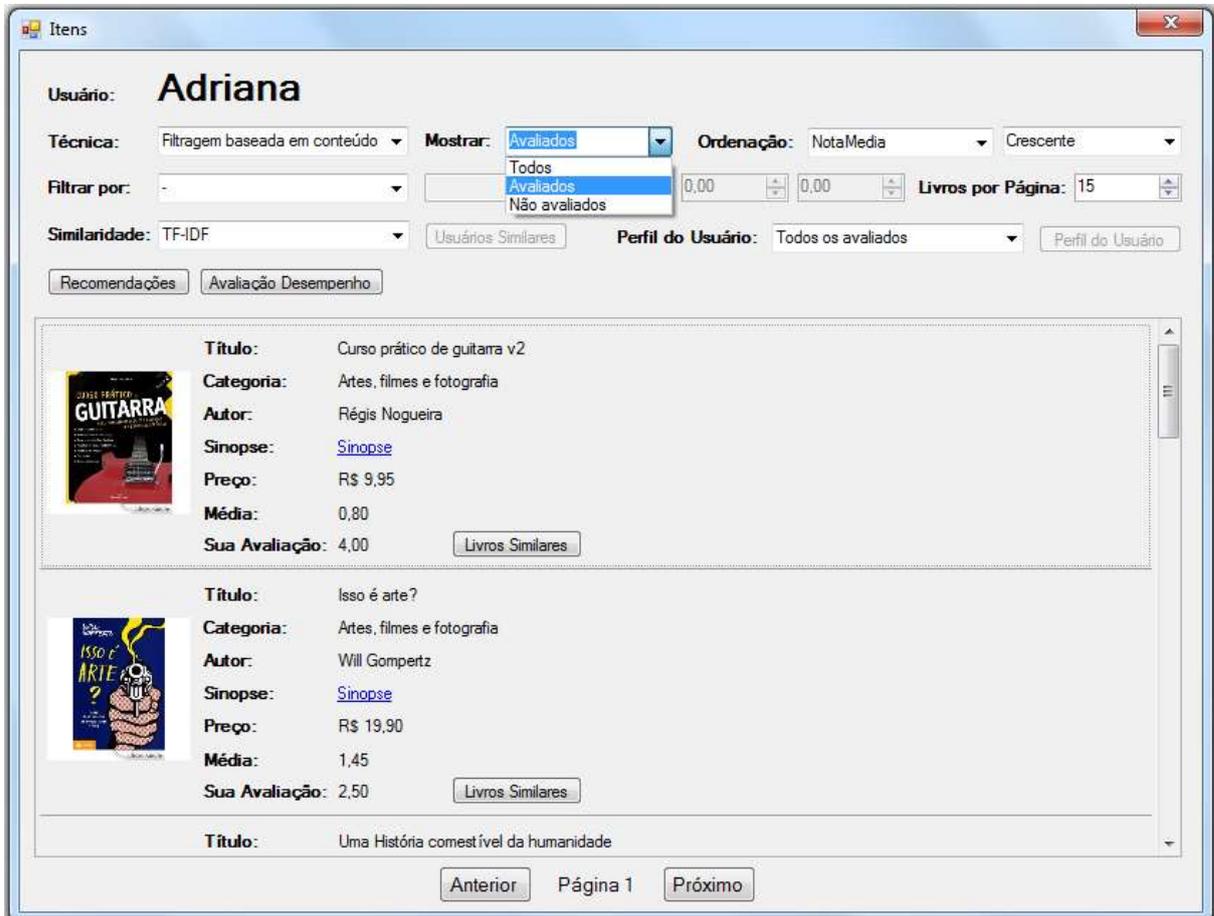


Figura 21 - Filtro de itens avaliados

- Ordenar os itens de forma crescente ou decrescente de acordo com algum de seus atributos, exceto sinopse, site e avaliação dada pelo usuário logado (figura 22);

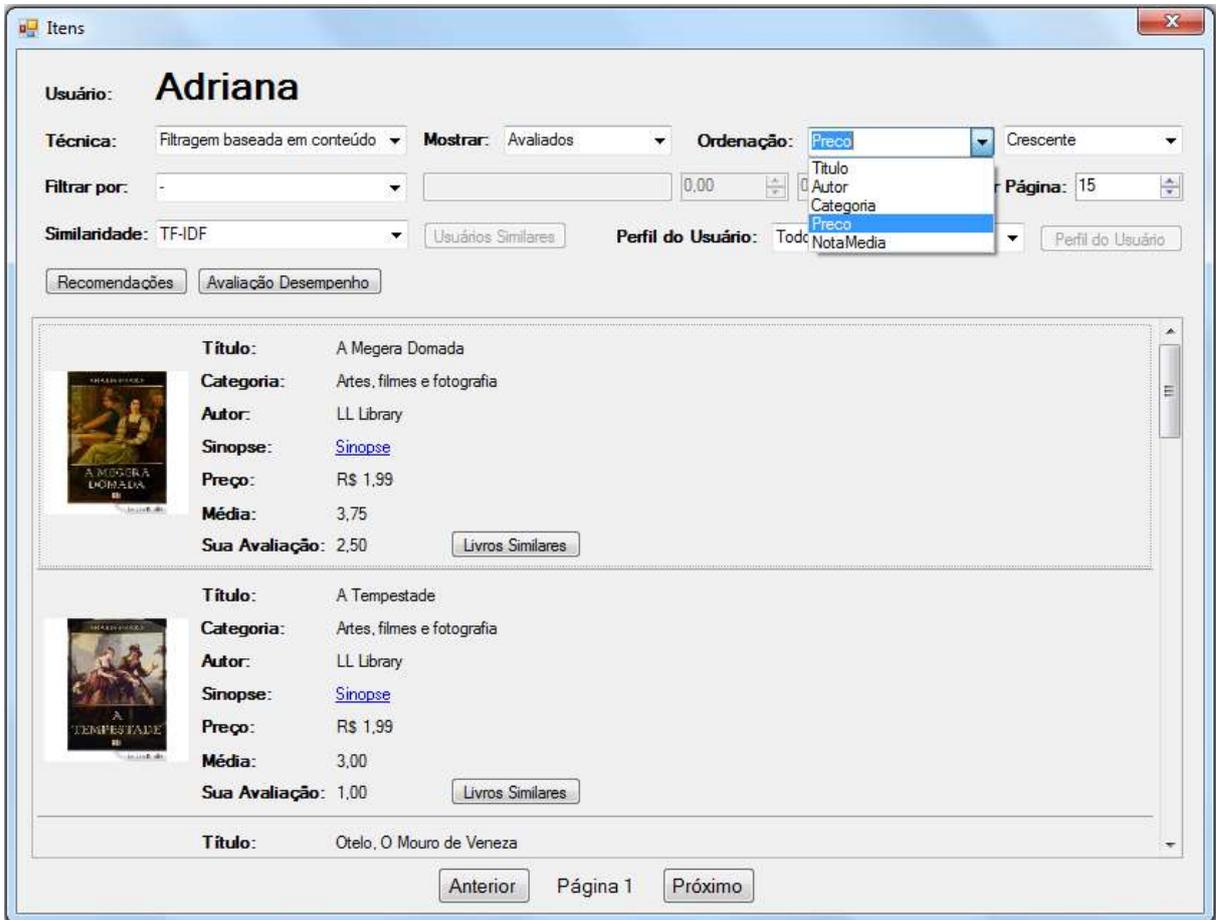


Figura 22 - Ordenação dos itens por atributo

- Filtrar os itens exibidos de acordo com algum de seus atributos, exceto sinopse, site e avaliação dada pelo usuário logado. Para o caso do título, categoria ou autor, é feita uma comparação com o texto digitado na busca (figura 23). Para o caso do preço e da nota média o usuário deve especificar um intervalo de valores nos quais os valores do item devem estar contidos (figura 24);

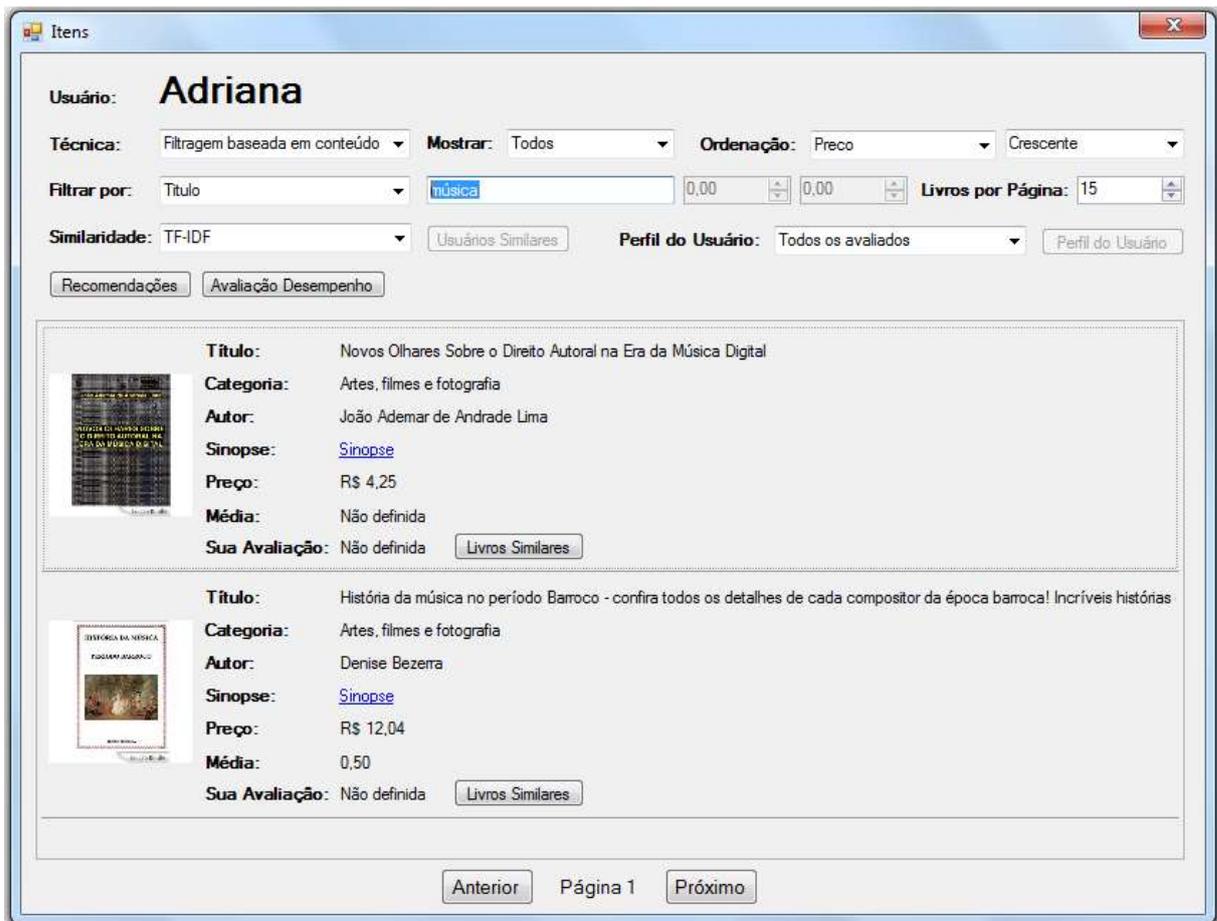


Figura 23 - Filtro de itens por atributo textual

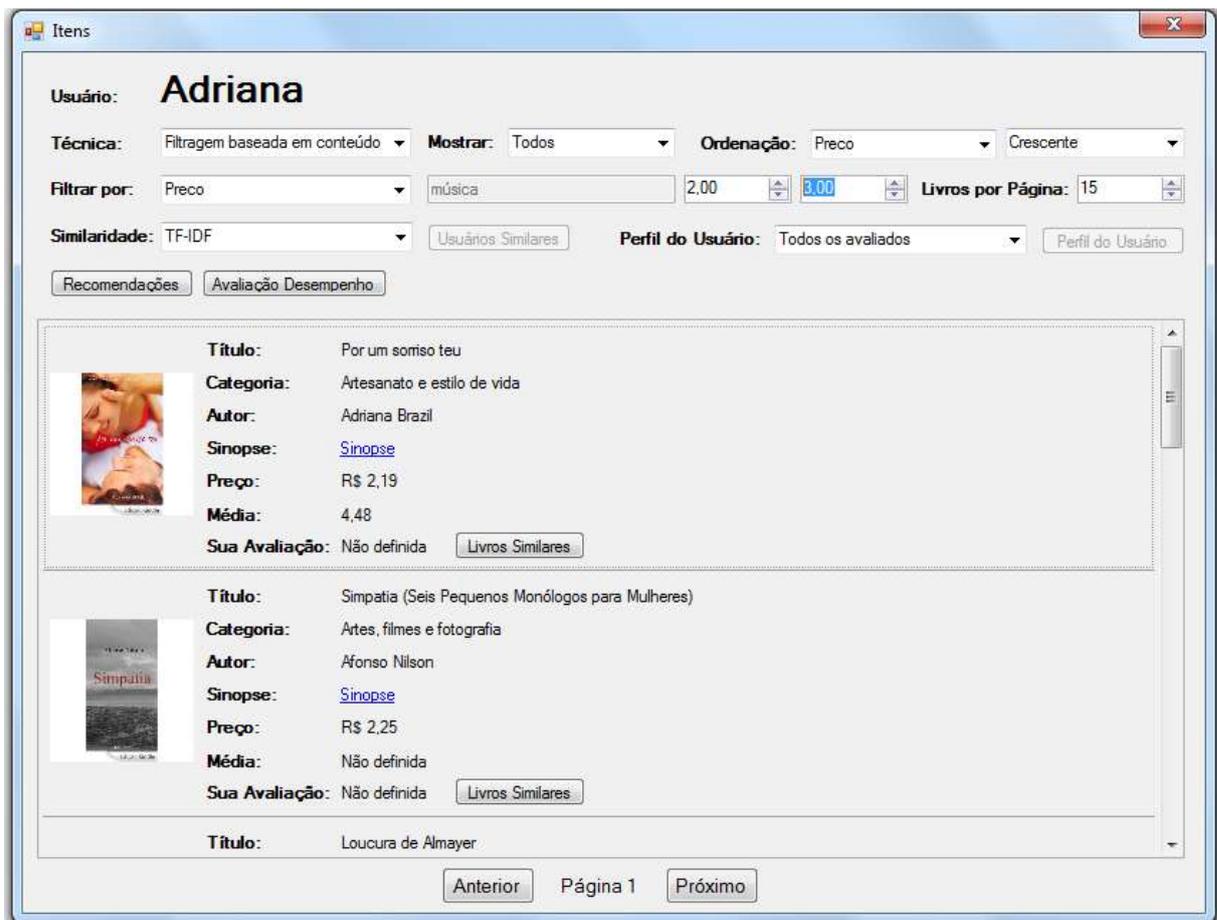


Figura 24 - Filtro de itens por atributo numérico

- Definir a quantidade de livros exibidos por página;
- Acessar a página anterior da lista de itens;
- Acessar a próxima página da lista de itens;

- Acessar a lista de usuários similares ao usuário logado. Esta opção só é possível caso seja selecionada a técnica de filtragem colaborativa. Para o cálculo da similaridade poderá ser escolhida uma dentre as seguintes técnicas: Cosine Similarity, Adjusted Cosine Similarity e Pearson Correlation. Nesta tela são exibidos os usuários em ordem decrescente de similaridade com o usuário logado (figura 25).

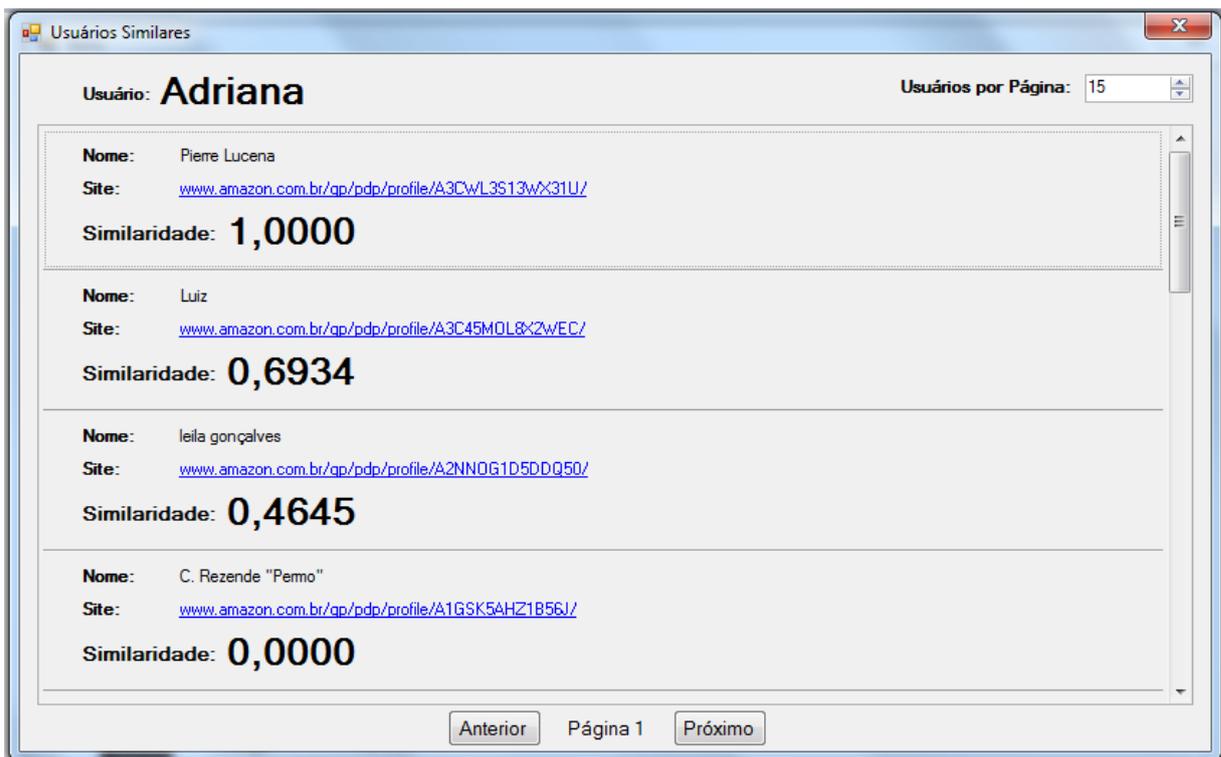


Figura 25 - Tela de similaridade de usuários

- Acessar a lista de itens similares a um determinado item da lista. Se for escolhida a técnica de filtragem baseada em conteúdo, a similaridade é calculada usando-se o TF-IDF. Se for selecionada a técnica de filtragem colaborativa, a similaridade poderá ser escolhida entre Cosine Similarity, Adjusted Cosine Similarity e Pearson Correlation. Se for selecionada a técnica de filtragem híbrida, a similaridade é calculada como uma combinação das anteriores. Nesta tela são exibidos os itens em ordem decrescente de similaridade com o item selecionado na lista, e pode-se definir o valor do peso dado a cada uma das técnicas componentes da filtragem híbrida (figura 26).



Figura 26 - Tela de similaridade de itens

- Definir se o perfil do usuário será o conjunto de todos os itens que ele já avaliou, ou apenas alguns deles que melhor representem os seus interesses. Se for selecionada esta segunda opção, o usuário poderá acessar a tela de seu perfil, com os K centroides do algoritmo de clustering utilizado para definir os itens que melhor representam seus interesses. Cada centróide apresenta um peso referente ao peso de seu cluster, calculado como sendo a soma de sua similaridade com os itens que o compõem. Nesta tela o usuário pode definir o valor deste parâmetro K e a quantidade de iterações do algoritmo de clustering, e assim recalculer estes centroides (figura 27).

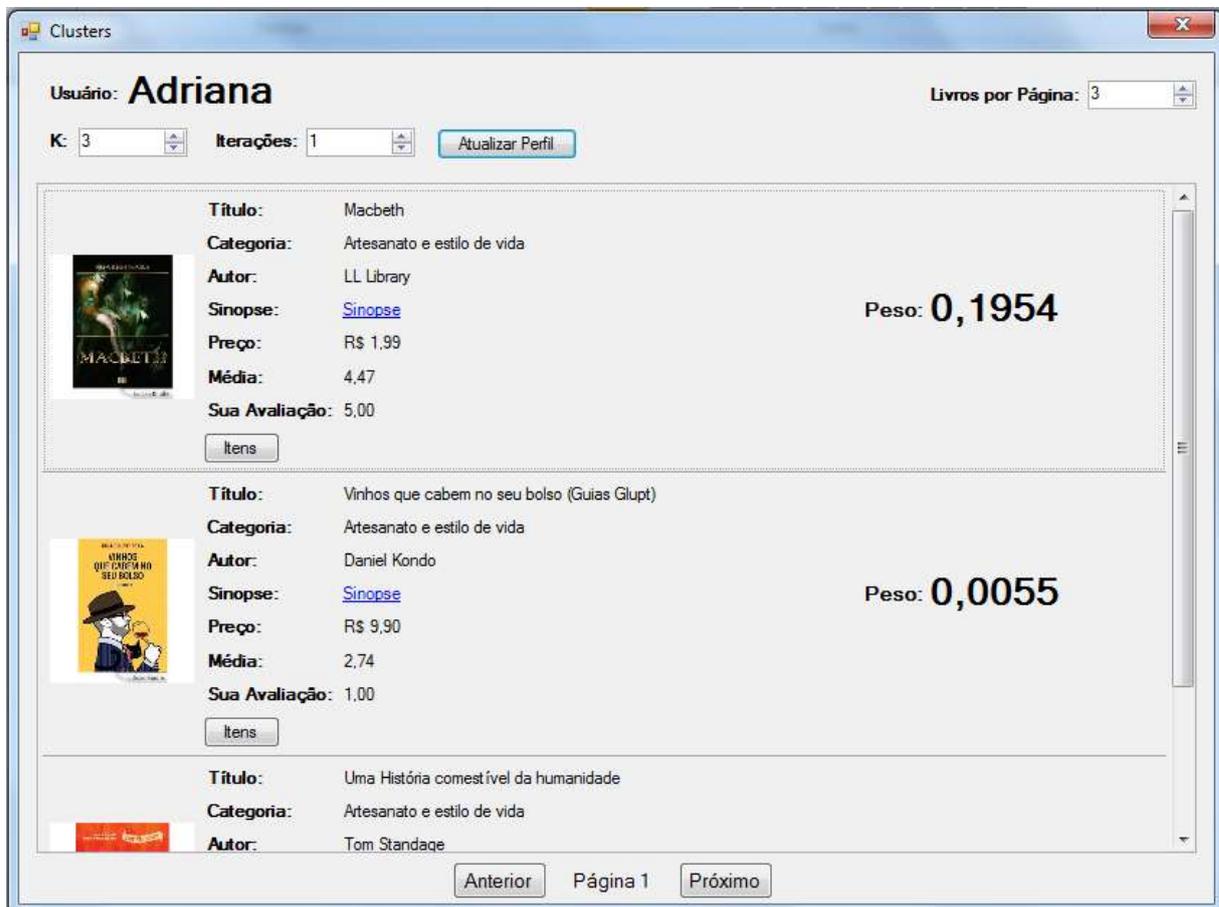


Figura 27 - Tela de centroides do perfil de usuário

Para cada centróide é possível ver quais são os itens que estão em seu cluster (figura 28), e o quão próximos estão dele (valor do peso).

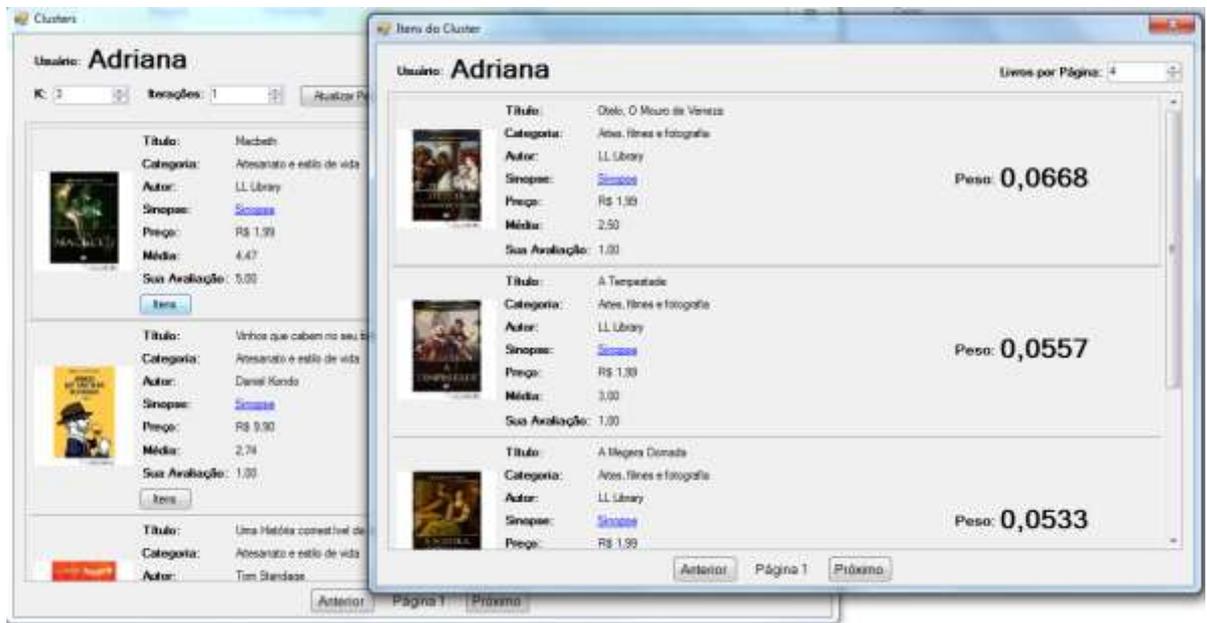


Figura 28 - Tela de itens dos clusters do perfil do usuário

- Acessar a tela das recomendações que o sistema oferece para o usuário logado. Esta é a tela mais importante do sistema de recomendação desenvolvido. Obviamente nela são listados apenas os itens que o usuário ainda não avaliou. Nela é possível qual algoritmo de predição de notas será utilizado. No caso da técnica baseada em conteúdo é usada a própria similaridade do TF-IDF. No caso da técnica colaborativa podem ser escolhidos o algoritmo Slope One, o Simple Weighted Average ou o Weighted Sum Others Ratings, sendo que para os dois últimos o usuário deverá definir o tamanho da vizinhança a ser utilizada no cálculo da predição. No caso da técnica híbrida podem ser escolhidos o algoritmo ponderado ou o da soma dos inversos das posições, sendo que apenas para o primeiro deles o usuário deverá definir um valor de α a ser utilizado no cálculo da predição. Dentre os itens não avaliados pelo usuário, serão listados apenas aqueles que foram recomendados, o que se baseia em uma porcentagem dos que apresentarem as melhores notas previstas ou todos aqueles com nota prevista superior a um determinado valor de limiar. Nesta tela o usuário pode ainda salvar uma nova avaliação, o que faz com que as notas previstas dos outros itens ainda não avaliados sejam recalculadas, bem como a nota média do item recém avaliado (figura 29).

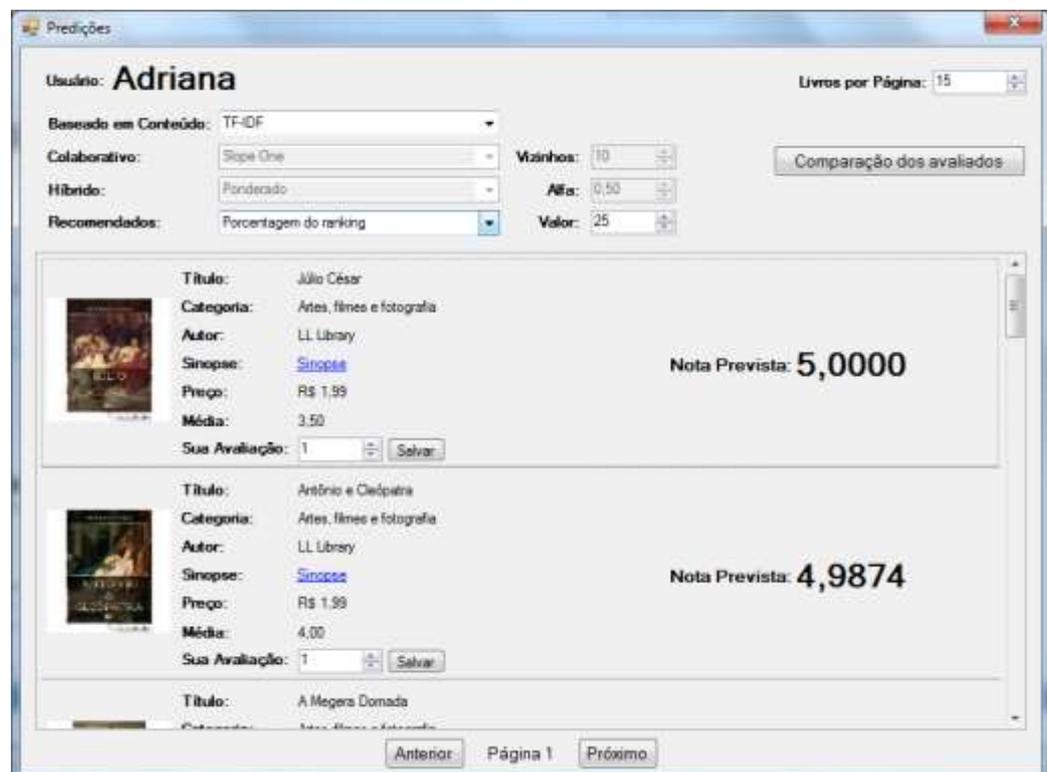


Figura 29 - Tela de recomendações

Nesta tela ainda é possível acessar a tela de comparação entre as notas reais e previstas dos itens já avaliados (figura 30). Neste ponto, todos os itens que forem avaliados no sistema já terão as notas previstas geradas, mas o mesmo não ocorre com os itens que já continham avaliações quando se deu a realização da busca das informações com o Crawler. Para estes últimos é então utilizada a técnica do leave one out, que consiste em assumir que o item não foi avaliado e então deixar o sistema prever uma nota para ele.

The screenshot shows a web application window titled "Comparação dos avaliados" for user "Adriana". The window displays a list of books with their predicted and real ratings. The books shown are:

Book Title	Category	Author	Price (R\$)	Average	Real Rating	Predicted Rating
Isso é arte?	Artes, filmes e fotografia	Will Gompertz	19,90	1,45	2,5000	0,3694
Vinhos que cabem no seu bolso (Guias Glupt)	Artesanato e estilo de vida	Daniel Kondo	9,90	2,74	1,0000	0,0169
Macbeth	Artesanato e estilo de vida	LL Library	1,99	4,47	5,0000	3,6304
A Tempestade						

At the bottom of the window, there are navigation buttons: "Anterior", "Página 1", and "Próximo". The window also shows "Livros por Página: 11" and the user's name "Usuário: Adriana".

Figura 30 - Tela de comparação entre notas prevista e real dos itens já avaliados

- Acessar a tela de avaliação de desempenho, na qual o usuário pode comparar e visualizar graficamente os valores de erros de predição e também os erros de classificação entre as técnicas variando diversos parâmetros. Os gráficos que ele pode visualizar são os seguintes:

- Comparação das Medidas de Erro de Predição (figura 31);

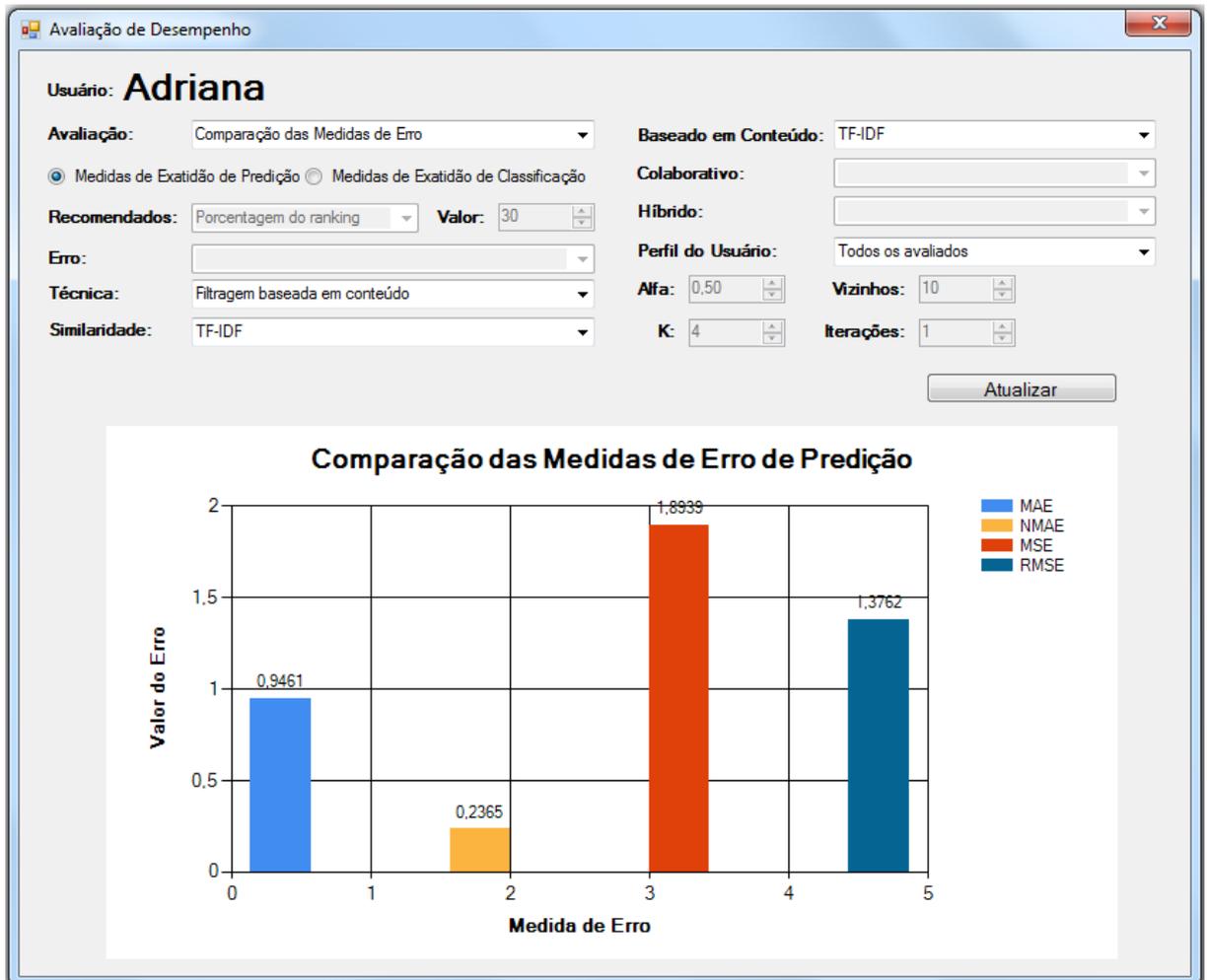


Figura 31 - Comparação das medidas de erro de predição

- Comparação das Medidas de Erro de Classificação (figura 32);

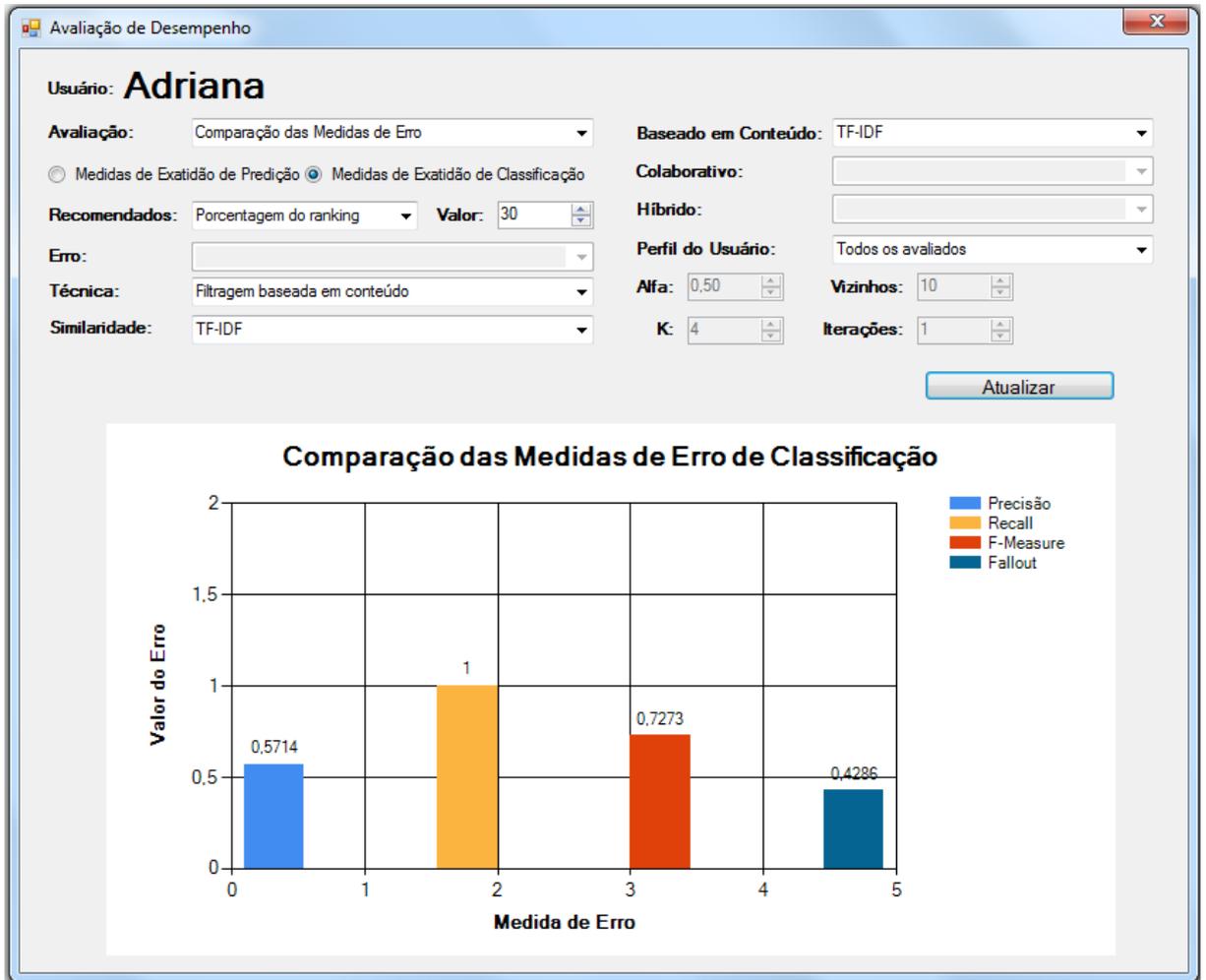


Figura 32 - Comparação das medidas de erro de classificação

- Comparação dos Algoritmos de Similaridade Colaborativos (figura 33);

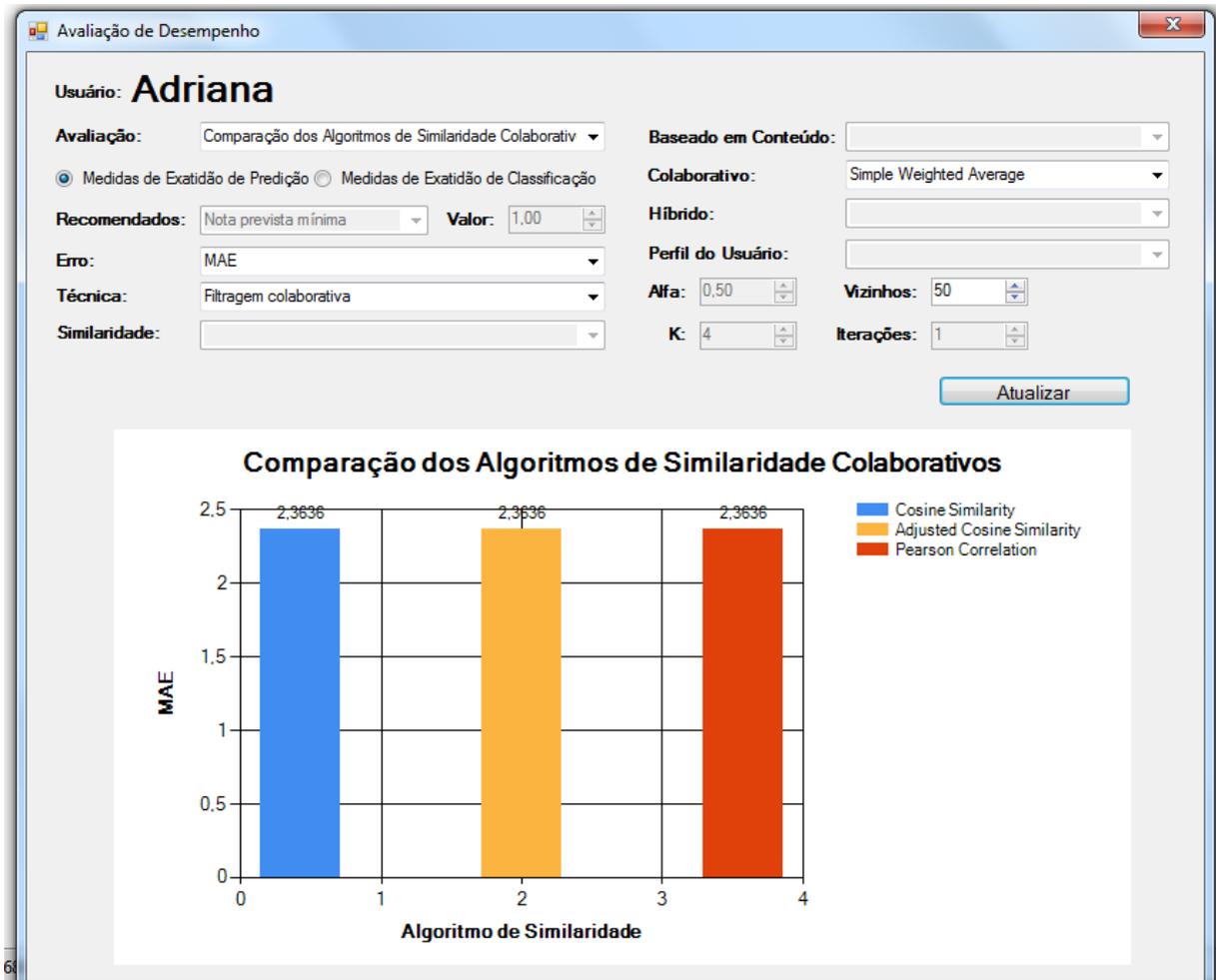


Figura 33 - Comparação de erro MAE dos algoritmos de similaridade colaborativos

- Comparação das Técnicas de Filtragem (figura 34);

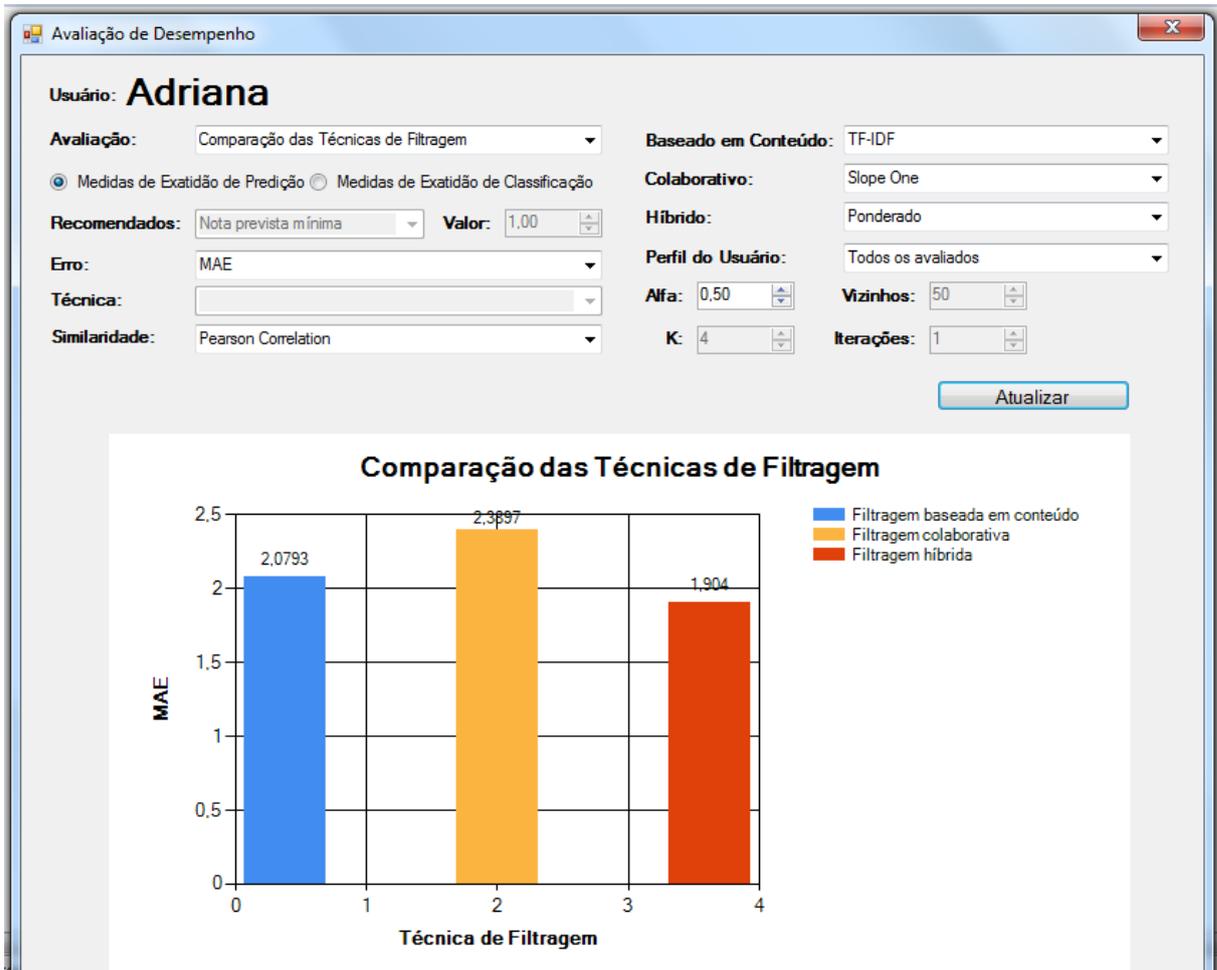


Figura 34 – Comparação de erro MAE das técnicas de filtragem

- Comparação dos Algoritmos Colaborativos (figura 35);

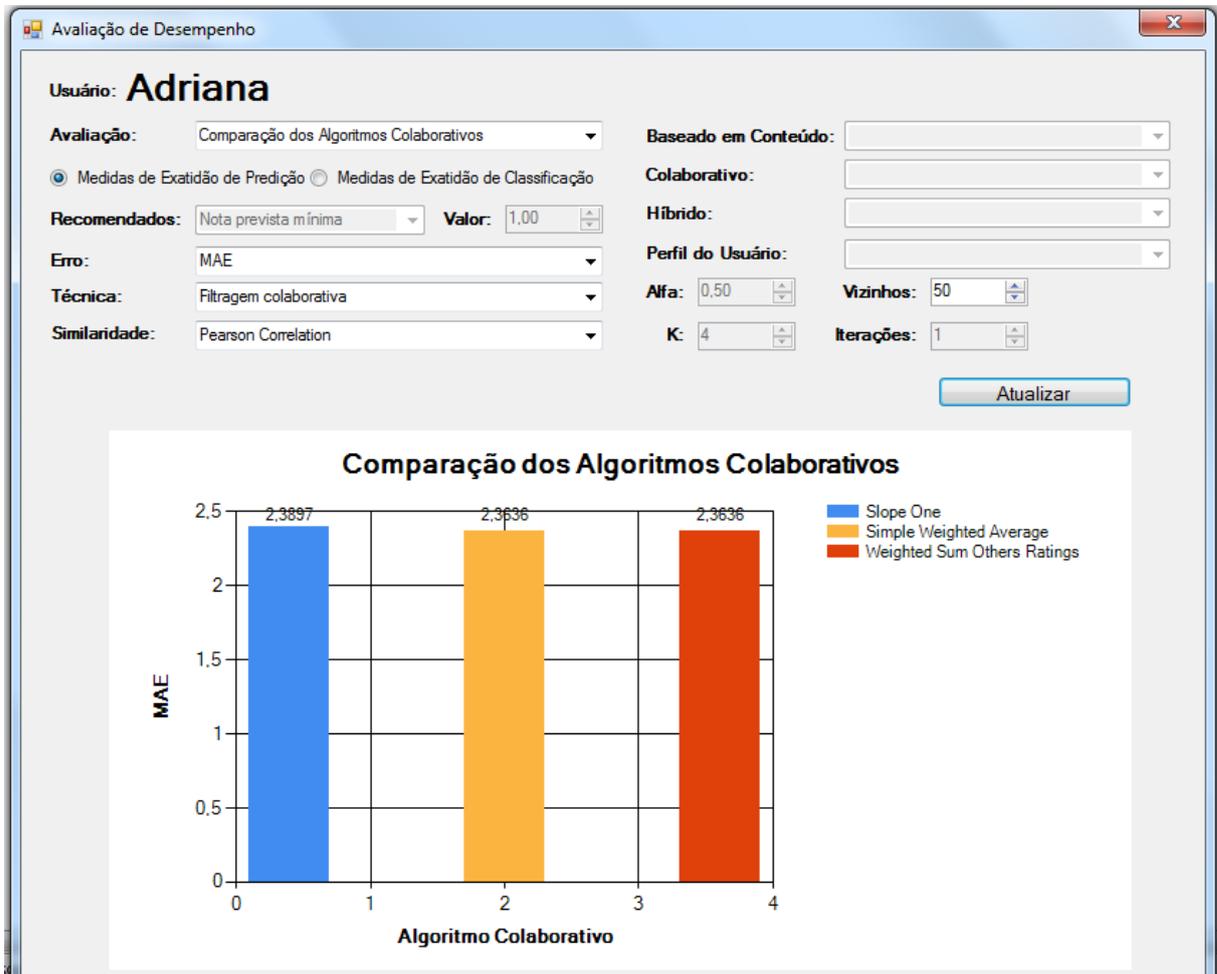


Figura 35 - Comparação de erro MAE dos algoritmos colaborativos

- Comparação dos Algoritmos Híbridos (figura 36);

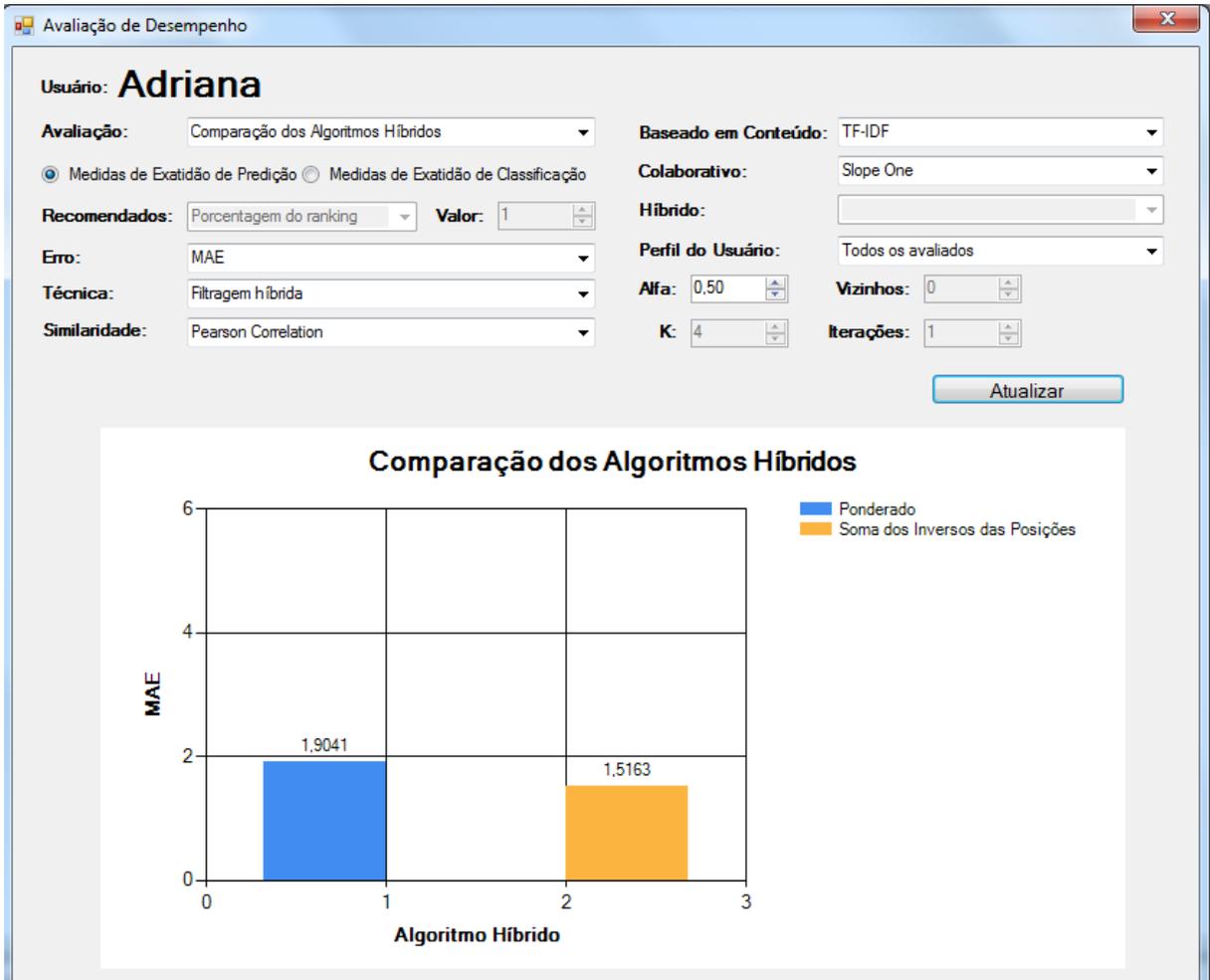


Figura 36 - Comparação de erro MAE dos algoritmos híbridos

Na figura 37 pode ser visualizada a comparação de desempenho entre os algoritmos híbridos, utilizando a curva ROC e o valor da AUC.

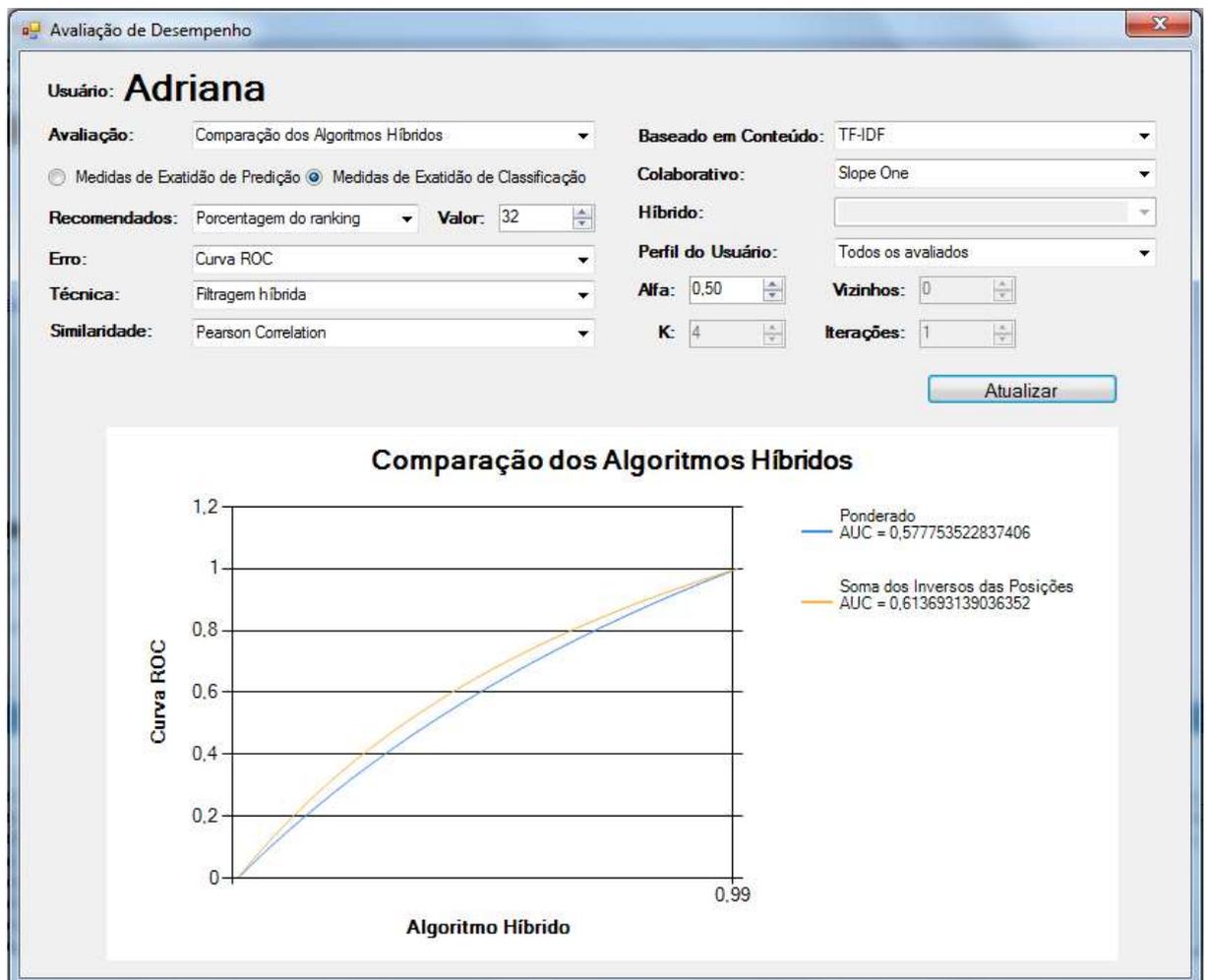


Figura 37 - Comparação de curva ROC entre algoritmos híbridos

- Variação do Tamanho da Vizinhança (figura 38);

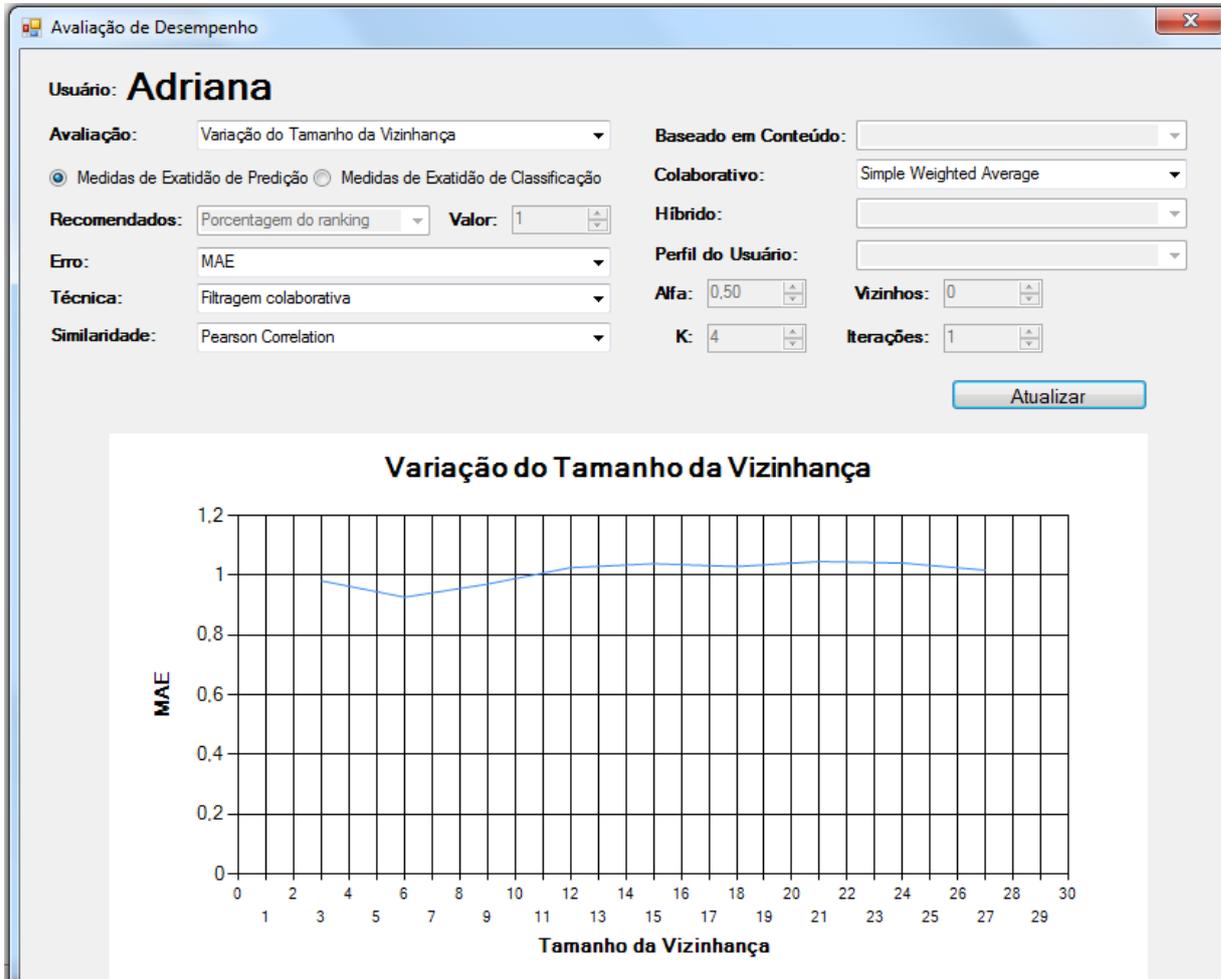


Figura 38 - Variação de erro MAE com o tamanho da vizinhança

- Variação do Parâmetro α (figura 39);

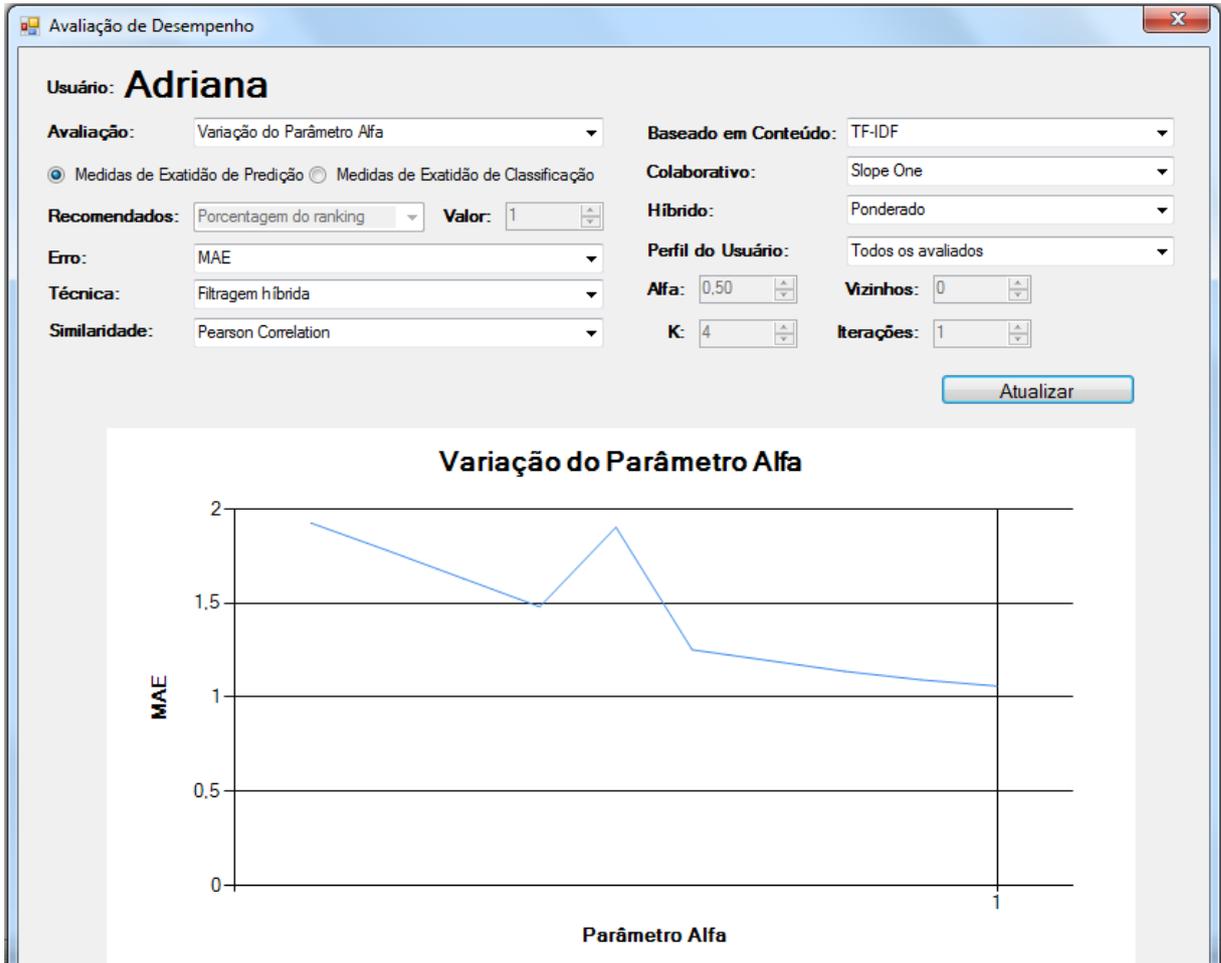


Figura 39 - Variação de erro MAE com o parâmetro α

- Variação do Parâmetro K (figura 40);

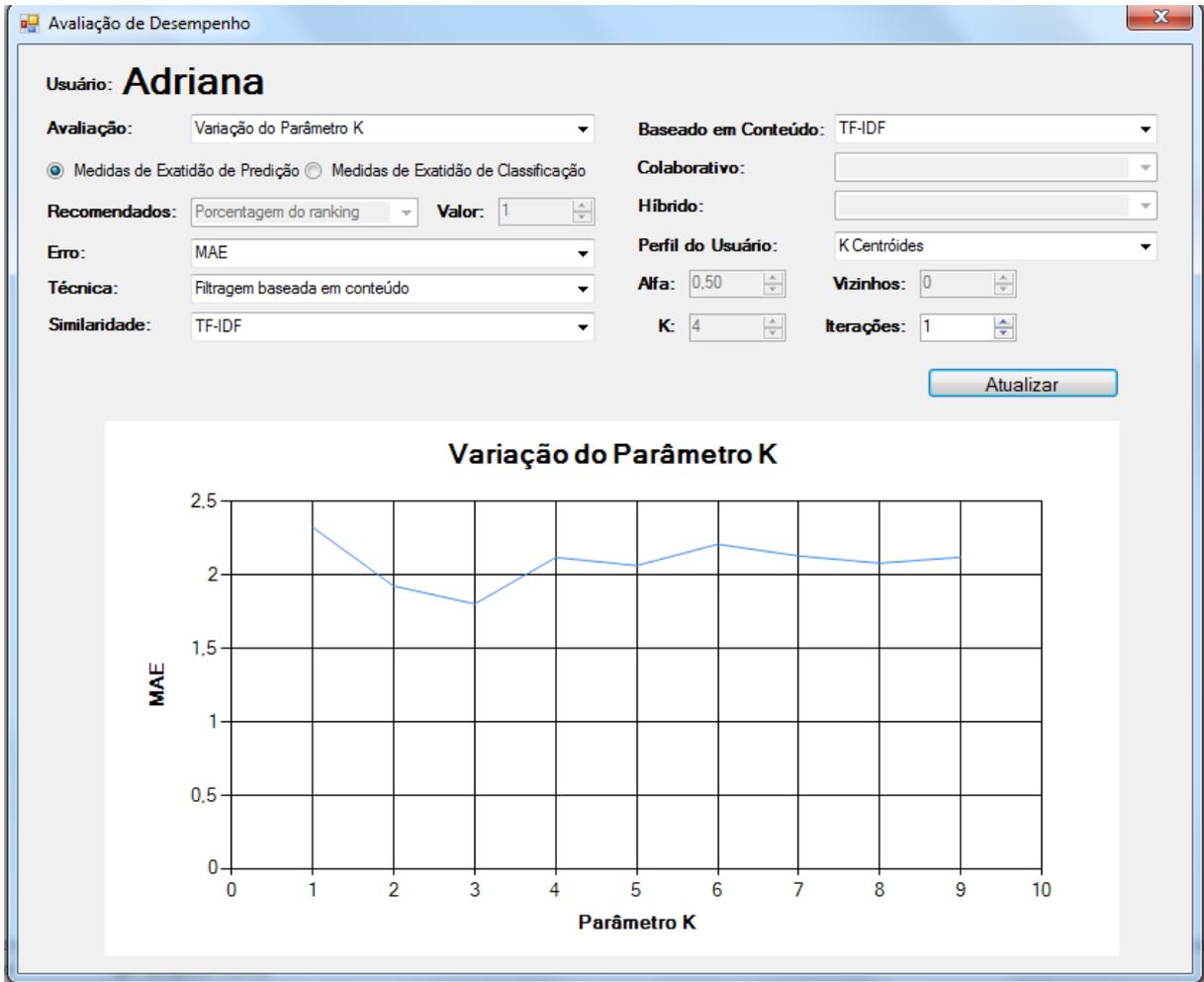


Figura 40 - Variação de erro MAE com o parâmetro K

- Variação da Quantidade de Iterações (figura 41);

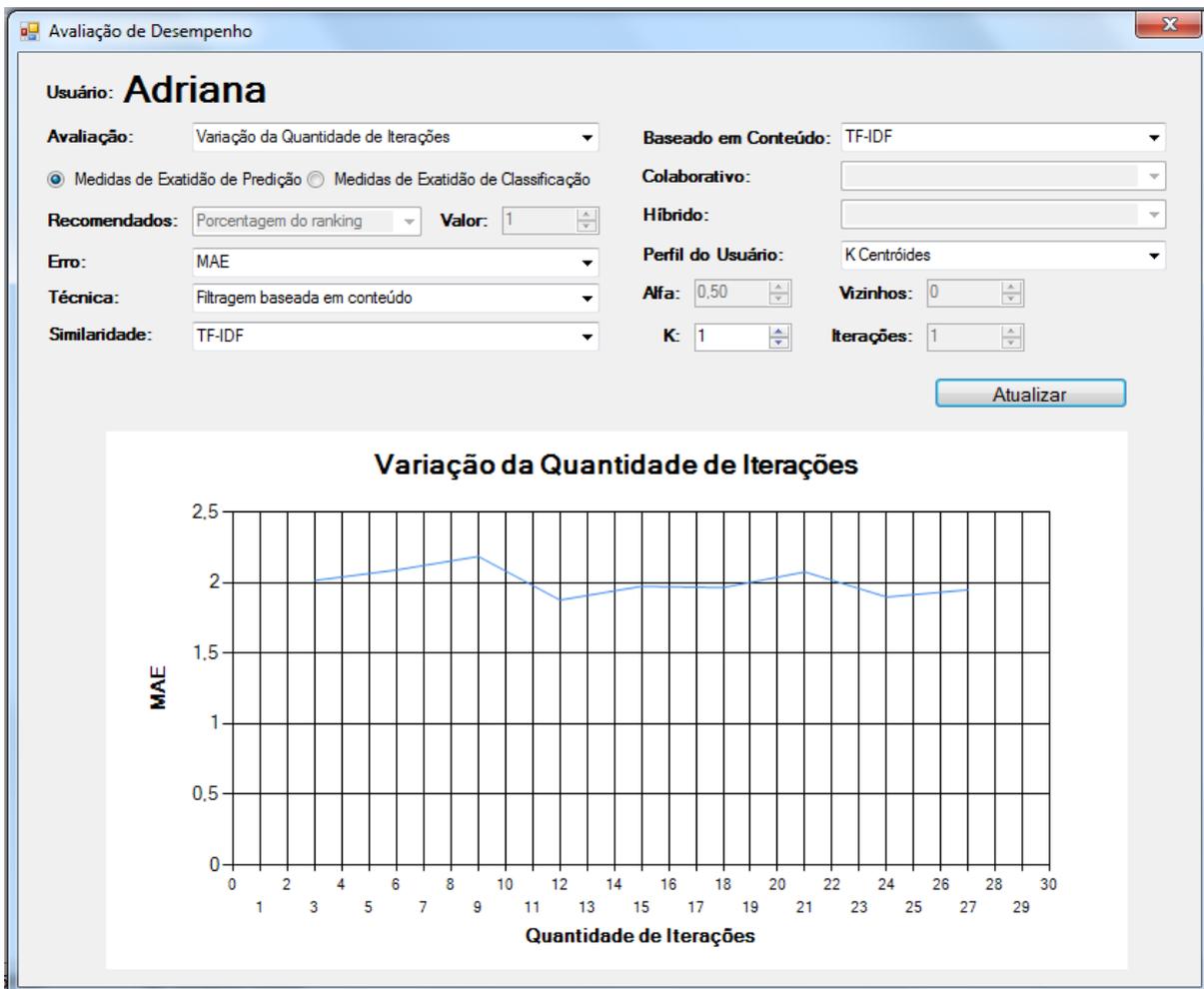


Figura 41 - Variação de erro MAE com a quantidade de iterações

o Matriz de Confusão (figura 42)

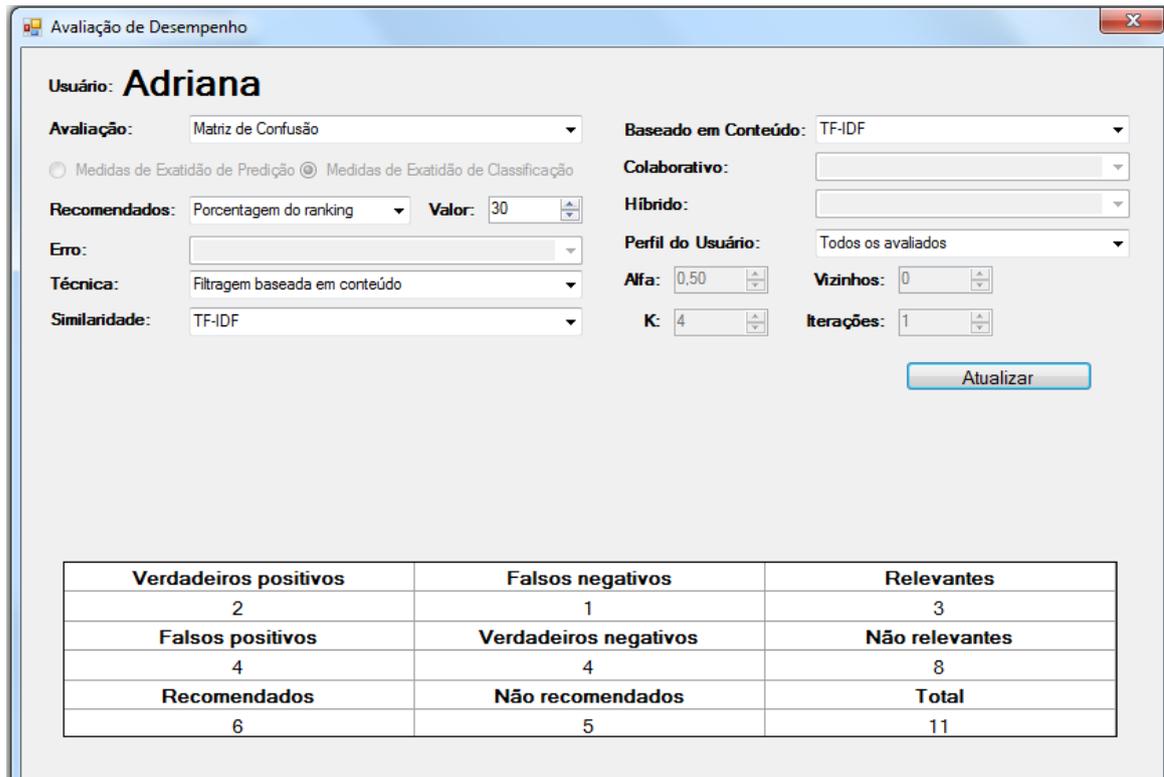


Figura 42 - Matriz de confusão

As medidas de erro preditivo utilizadas para avaliação do sistema foram MAE, NMAE, MSE e RMSE. As medidas de erro de classificação utilizadas foram precisão, recall, f-measure, fallout e curva ROC, todas estas últimas calculadas a partir da matriz de confusão. Esta matriz de confusão é gerada a partir da formação de quatro conjuntos: o conjunto dos itens relevantes, o conjunto dos itens não relevantes, o conjunto dos itens recomendados e o conjunto dos itens não recomendados. Foi decidido que um item é considerado relevante se foi avaliado pelo usuário com nota maior ou igual a três, sendo não relevante caso contrário. Já quanto a ser ou não recomendado, o sistema fornece duas opções ao usuário:

Porcentagem do ranking: neste caso um item é considerado recomendado ao usuário se apareceu entre os melhores na lista de predição de itens. Um valor de porcentagem deve ser fornecido pelo usuário. Como exemplo, se o usuário fornece o valor 25, então os 25% itens com maior nota prevista são considerados recomendados. De forma complementar, os 75% itens com menor nota prevista são considerados não recomendados.

Nota prevista mínima: neste caso um item é considerado recomendado ao usuário se sua nota prevista for maior ou igual ao valor fornecido pelo usuário, sendo não recomendado caso contrário.

5. Experimentos e Resultados

Neste trabalho foram realizados alguns experimentos com o objetivo de mostrar como as técnicas de filtragem híbridas conseguem superar as principais desvantagens de suas componentes colaborativa e baseada em conteúdo. Como os experimentos são feitos considerando o usuário que está logado no sistema, os mesmos serão realizados com três usuários distintos: um que avaliou poucos itens, um que avaliou muitos itens e um que avaliou uma quantidade média de itens. Os resultados serão avaliados, porém nem todos os gráficos serão exibidos neste trabalho, apenas os mais importantes.

O sistema apresenta uma grande variedade de parâmetros a serem variados, e assim haveria muitas combinações possíveis de experimentos. Resolveu-se estudar apenas algumas destas combinações, fixando algumas variáveis em valores considerados ideais para esta análise. Nos primeiros experimentos descritos a seguir, que visavam descobrir quais os melhores valores para fixar as variáveis para os últimos experimentos, optou-se por utilizar o erro MSE e a curva ROC. Com relação à curva ROC, considerou-se recomendado ao usuário um item cuja predição estivesse entre as 30% melhores.

5.1 Experimento 1 – Algoritmo Colaborativo

O primeiro experimento realizado objetivou determinar qual dos algoritmos colaborativos apresentava melhores resultados, tanto a nível classificativo quanto preditivo. Foram implementados dois algoritmos de previsão de avaliações do tipo baseado em item: Slope One e Simple Weighted Average. Este último utiliza uma matriz de similaridade entre os itens, que neste experimento foi computada usando a correlação de Pearson. Como algoritmo de previsão de avaliações baseado em usuário foi implementado o Weighted Sum Others Ratings, que utiliza uma matriz de similaridade entre os usuários, também computada usando a correlação de Pearson. Foi fixado o tamanho de vizinhança igual a 10. Após a execução do experimento, observou-se que o melhor algoritmo colaborativo foi o baseado em itens Simple Weighted Average, por possuir menor MSE (figura 43) e maior AUC (figura 44). Além de ter apresentado melhores resultados, foi possível perceber também que a abordagem baseada em item foi processada mais rapidamente do que a abordagem baseada em usuário. Isto pode ser explicado pelo fato de o sistema ser composto por mais usuários do que itens, apesar da diferença entre as quantidades não ser tão grande (109 usuários e 96 itens).

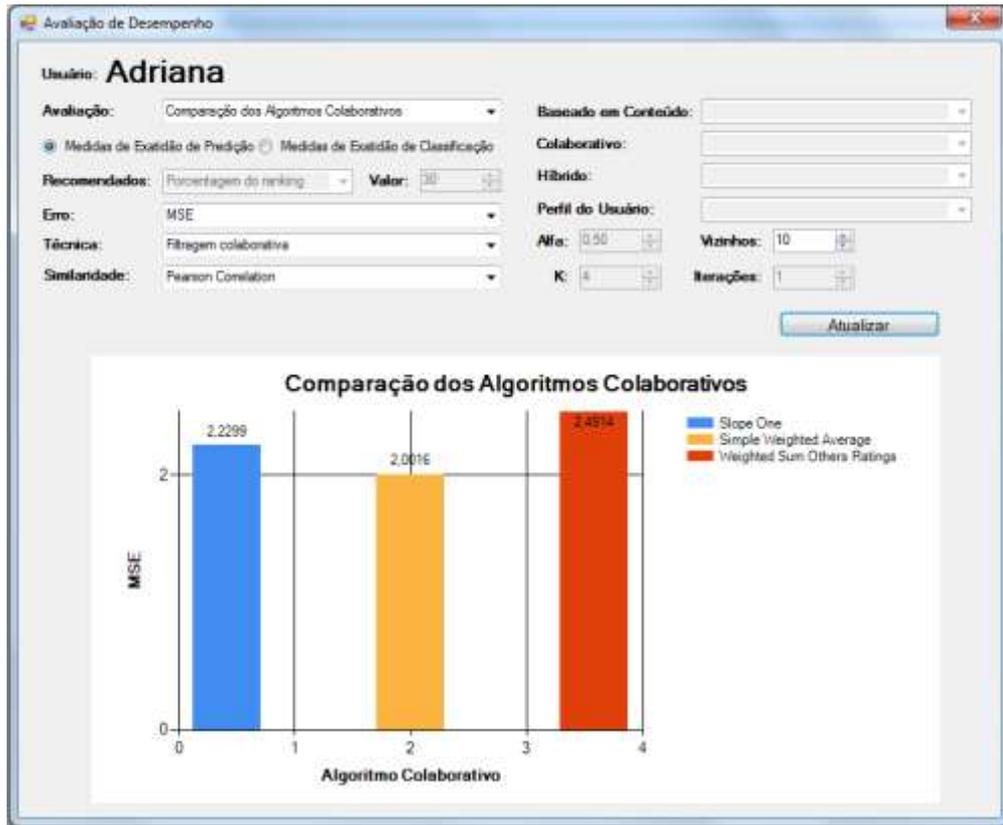


Figura 43 - Experimento 1: MSE dos algoritmos de predição colaborativa

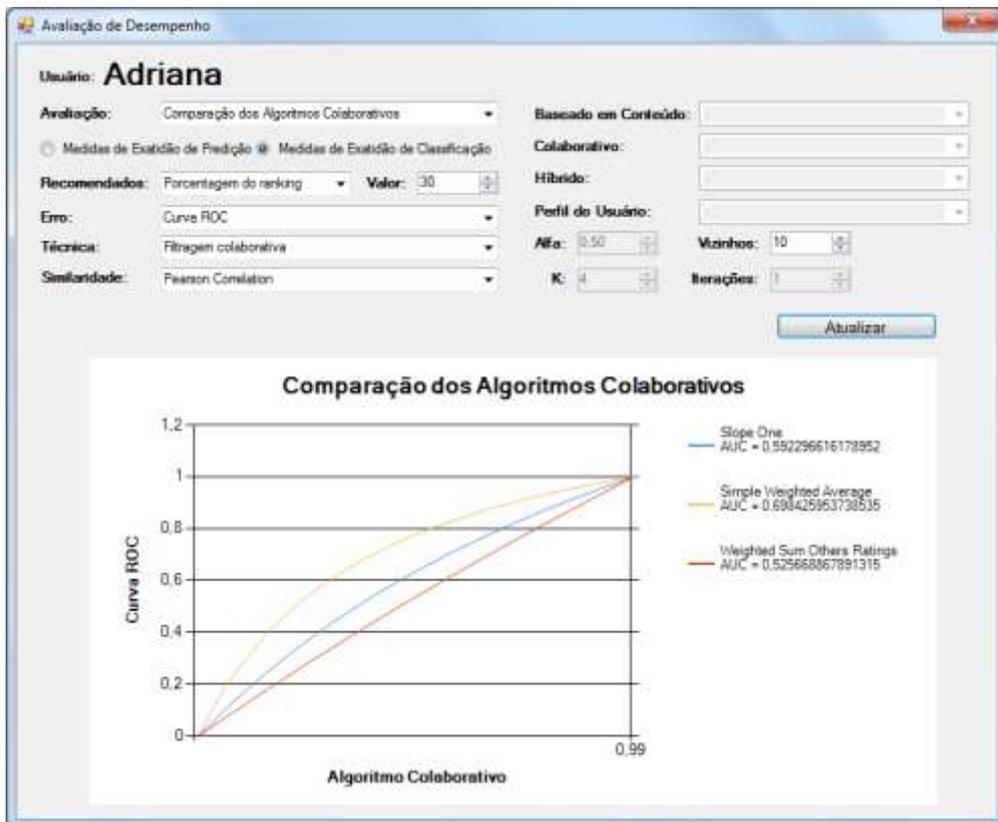


Figura 44 - Experimento 1: Curva ROC dos algoritmos de predição colaborativa

5.2 Experimento 2 – Algoritmo de Similaridade

Em seguida, fixando-se este algoritmo de predição colaborativa Simple Weighted Average, realizou-se o segundo experimento, a fim de determinar qual algoritmo de similaridade colaborativa apresentava os melhores resultados: o Cosine Similarity, o AdjustedCosine Similarity ou o Pearson Correlation. Novamente fixou-se o tamanho da vizinhança igual a 10. Pode-se observar que a similaridade possui um grande impacto no grau de exatidão das estimativas, já que alterando-se o algoritmo de similaridade adotado, os valores de erro variam bastante. Por ter sido o algoritmo de similaridade que obteve os melhores resultados (figura 45 e figura 46), o Pearson Correlation foi o utilizado nas próximas análises.

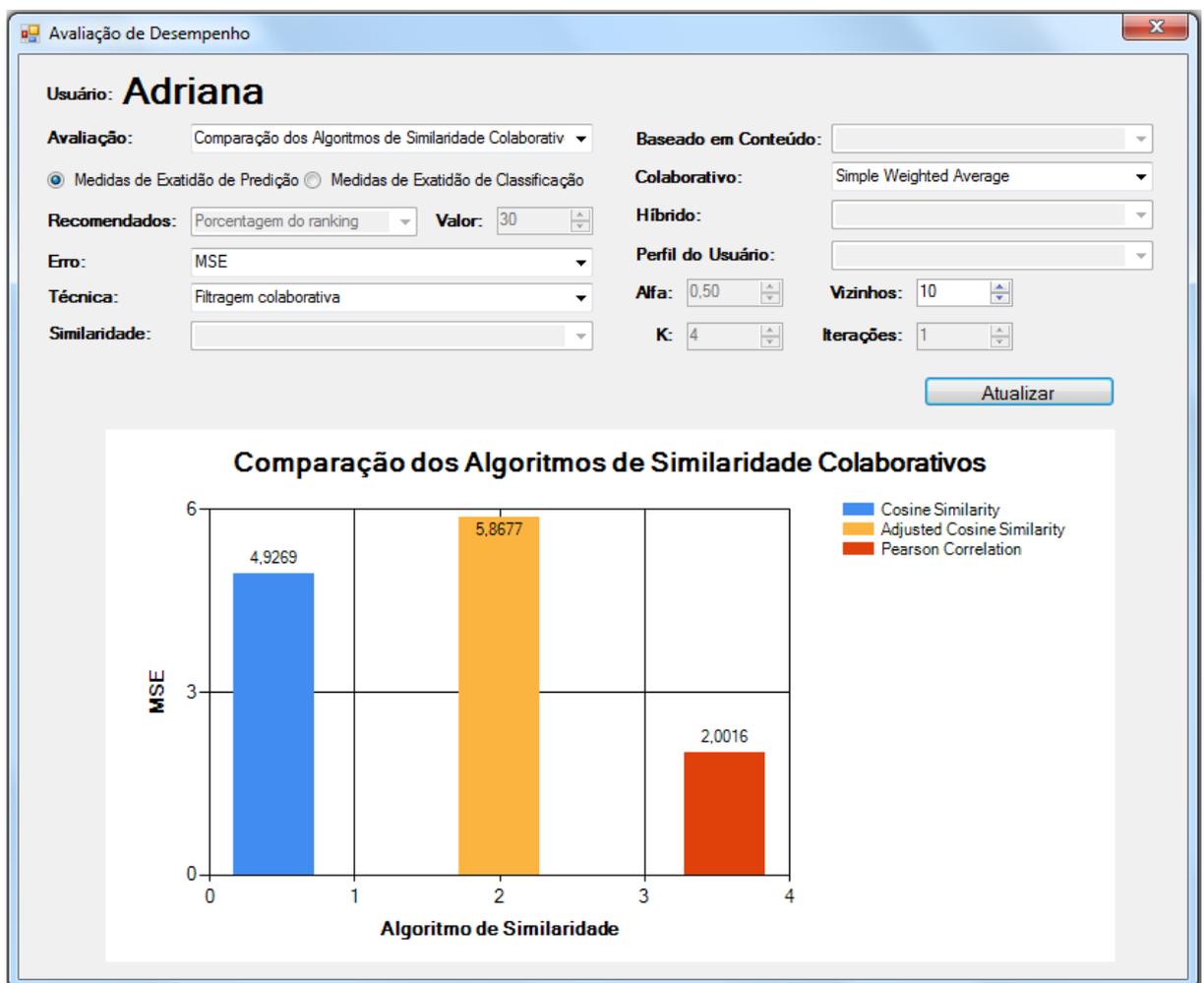


Figura 45 - Experimento 2: MSE dos algoritmos de similaridade colaborativa

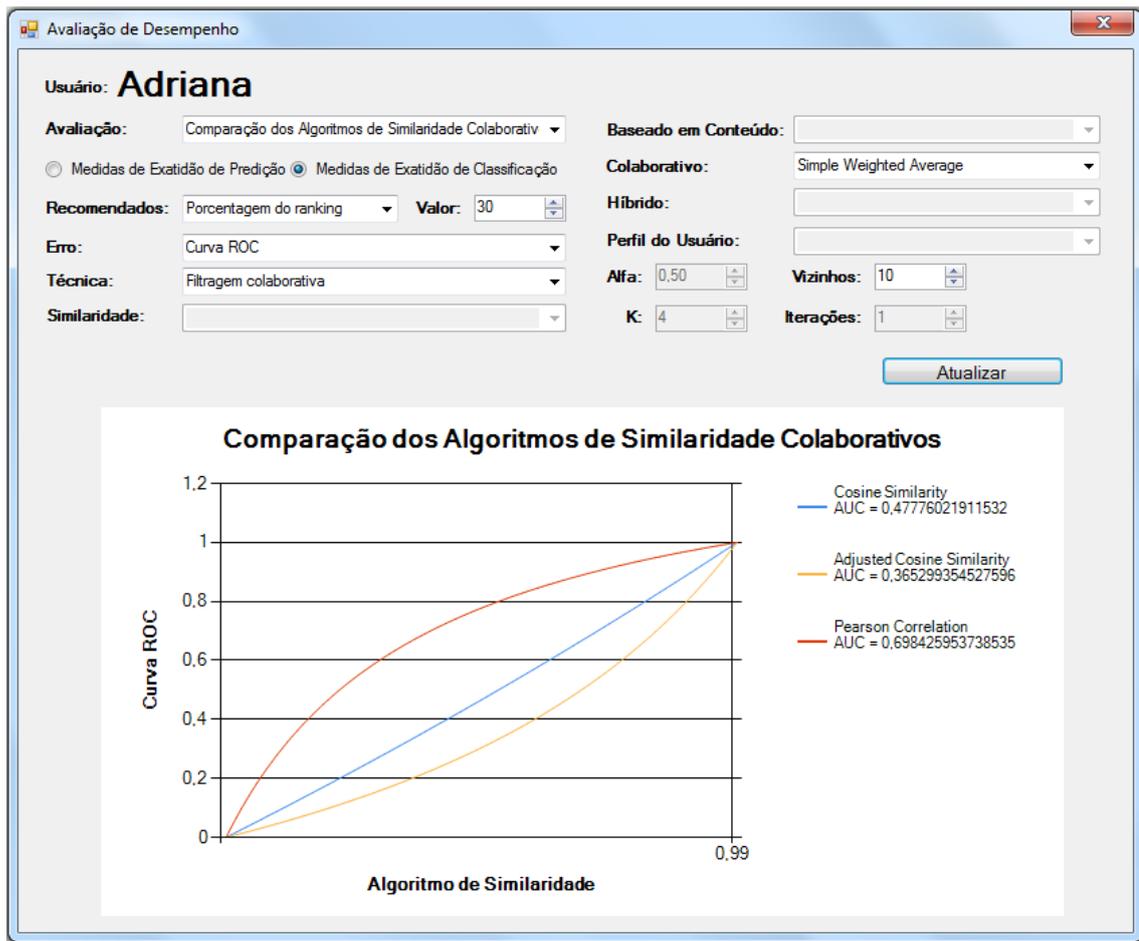


Figura 46 - Experimento 2: Curva ROC dos algoritmos de similaridade colaborativa

5.3 Experimento 3 – Tamanho da Vizinhança

O terceiro experimento consistiu em determinar qual o tamanho da vizinhança que melhores resultados trazia ao algoritmo de similaridade Pearson Correlation no algoritmo colaborativo Simple Weighted Average. Foi possível perceber que, inicialmente, o valor do erro cai consideravelmente à medida que o número de vizinhos aumenta. Entretanto, após atingir um valor mínimo, o erro aumenta e se estabiliza. Isso ocorre porque com um grande número de vizinhos, o algoritmo começa a considerar as avaliações de usuários que não são tão similares ao usuário ao qual se quer fazer a recomendação. Isto acaba degradando a qualidade da previsão. Em termos de custo de processamento, à medida que a vizinhança aumenta, o custo computacional do algoritmo também aumenta. Neste exemplo esta variável foi variada de 5 em 5 entre 0 e 30 e o menor valor do MSE (figura 47) e a maior AUC (figura 48) foram obtidos com a quantidade de vizinhos igual a 10. Assim este é considerado, neste exemplo, o valor ótimo para esta variável.

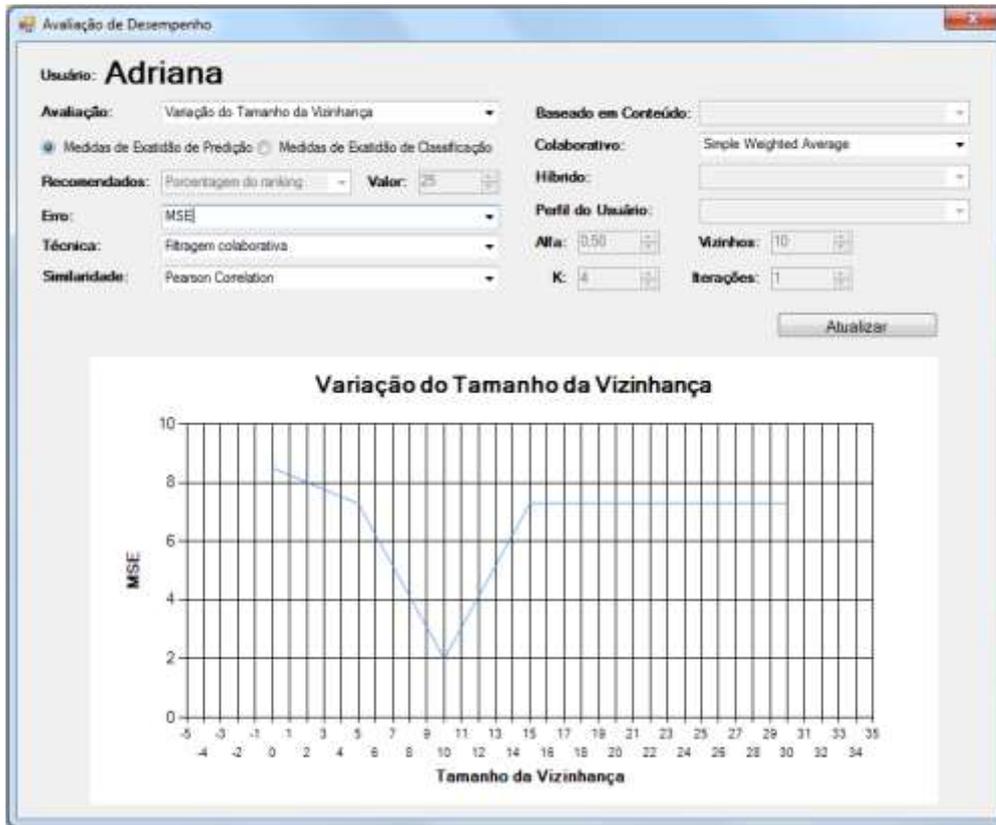


Figura 47 - Experimento 3: MSE da variação da quantidade de vizinhos

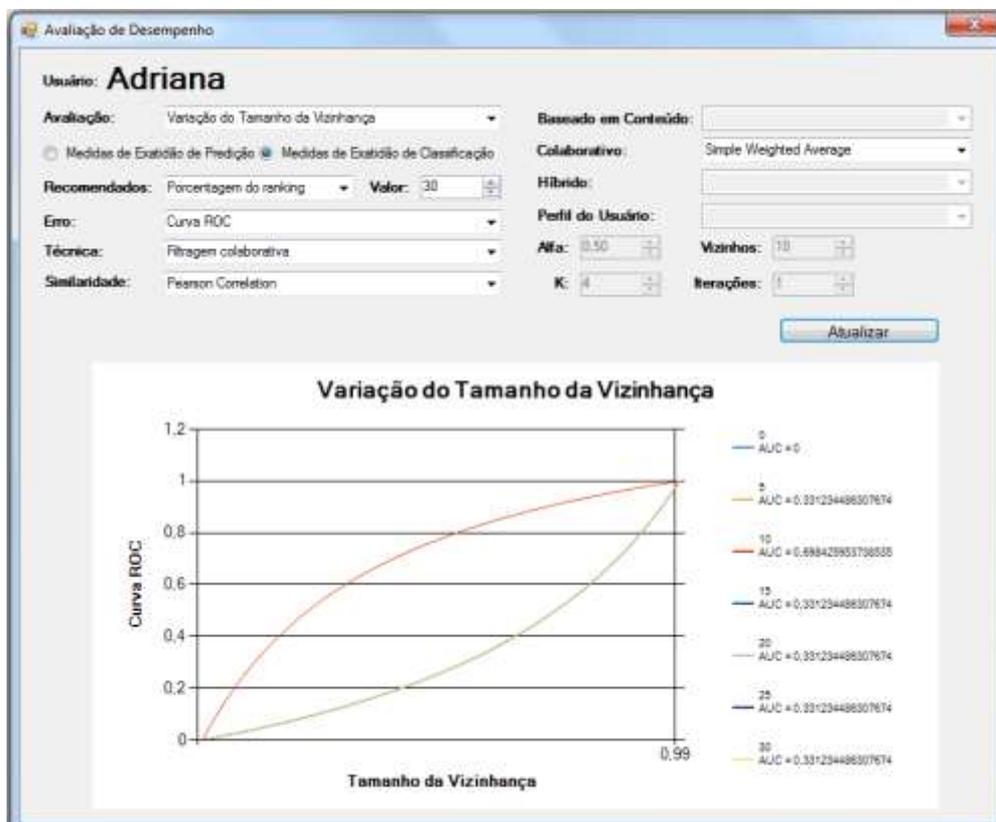


Figura 48 – Experimento 3: Curva ROC da variação da quantidade de vizinhos

5.4 Experimento 4 – Parâmetro K

Como quarto experimento, decidiu-se variar a quantidade de centroides que representariam o perfil do usuário para a técnica de filtragem baseada em conteúdo. O objetivo era verificar qual valor de K traria melhores resultados de recomendação ao usuário. Lembrando que escolher o valor máximo de K equivale a escolher a opção de todos os itens avaliados representarem o perfil do usuário logado no sistema. Para este experimento já foi fixado o valor de 30 iterações do algoritmo de clustering, por considerá-lo suficiente. O valor de K foi variado de 2 em 2 entre 1 e o número máximo de itens avaliados pelo usuário. Como era de se esperar, quanto menor o valor de K menor o tempo necessário para o cálculo da filtragem baseada em conteúdo entre o perfil do usuário e os itens ainda não avaliados por ele, o que implica em um menor tempo de resposta da recomendação. Após a realização do experimento, observou-se que o valor máximo de K apresentou menor MSE (figura 49) e maior AUC (figura 50). Sendo assim optou-se por escolhê-lo como valor ótimo para esta variável, ou seja, o perfil do usuário será mesmo representado por todos os itens já avaliados por ele.

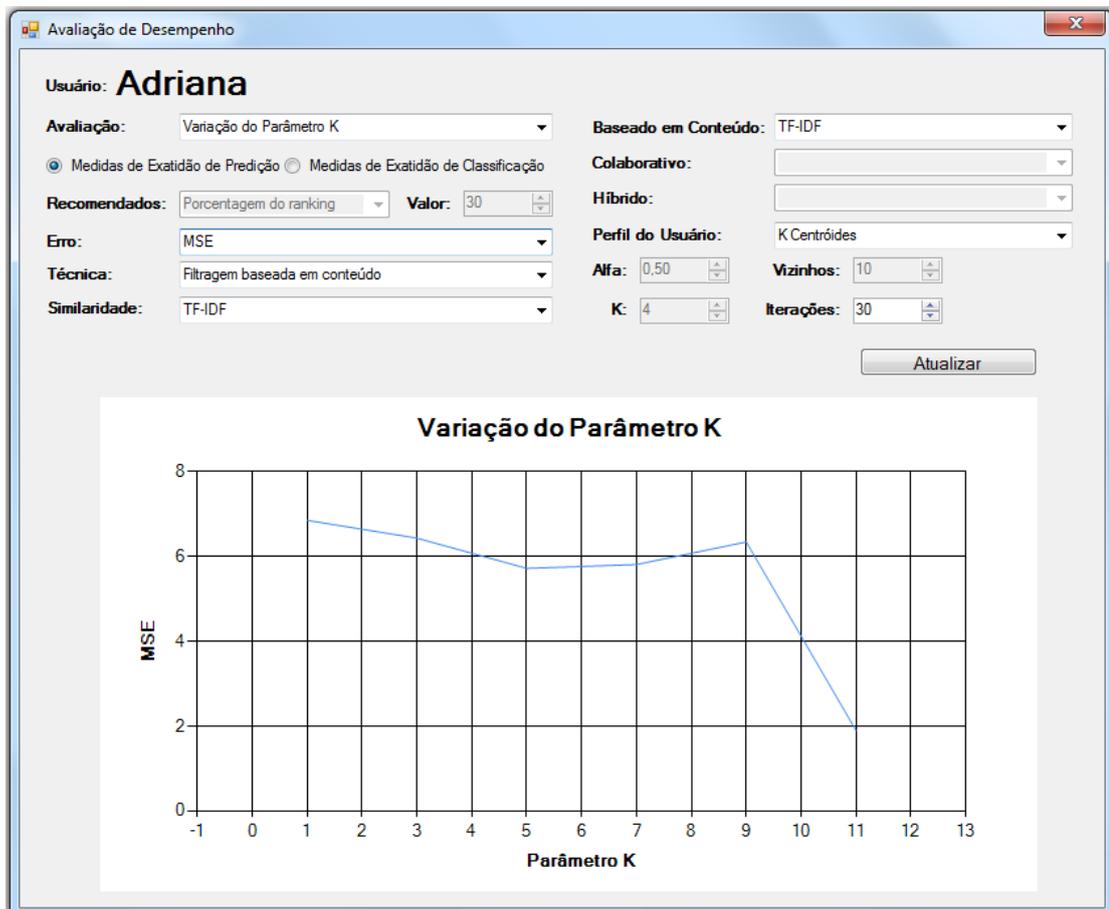


Figura 49 - Experimento 4: MSE da variação do parâmetro K

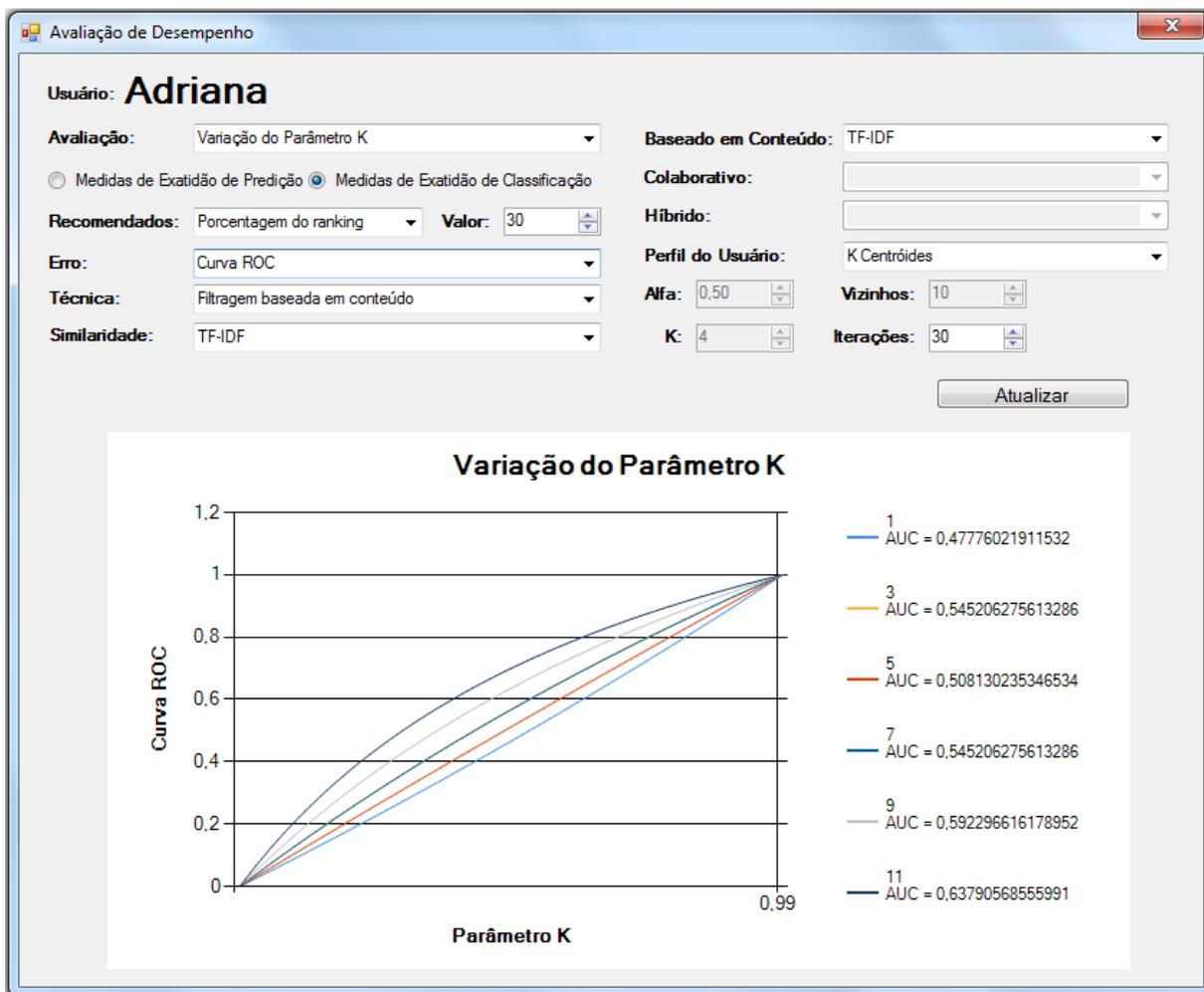


Figura 50 - Experimento 4: Curva ROC da variação do parâmetro K

5.5 Experimento 5 – Parâmetro α

Os resultados dos quatro experimentos iniciais foram utilizados como entrada para os demais experimentos. O quinto experimento variou o parâmetro α da filtragem híbrida ponderada. Lembrando que o valor de α igual a zero equivale ao uso da filtragem baseada em conteúdo. O valor de α igual a 1 equivale ao uso da filtragem colaborativa. Como em um experimento posterior já serão comparadas as três técnicas de filtragem, estes dois valores (0 e 1) não foram utilizados neste experimento. Sendo assim, neste experimento este parâmetro foi variado de 0.1 em 0.1 entre 0.1 e 0.9, identificando-se o valor 0.5 como valor ótimo (figura 51 e figura 52). Isto quer dizer que os melhores resultados se dão quando a filtragem híbrida ponderada considera de forma equitativa os resultados de suas partes colaborativa e baseada em conteúdo.

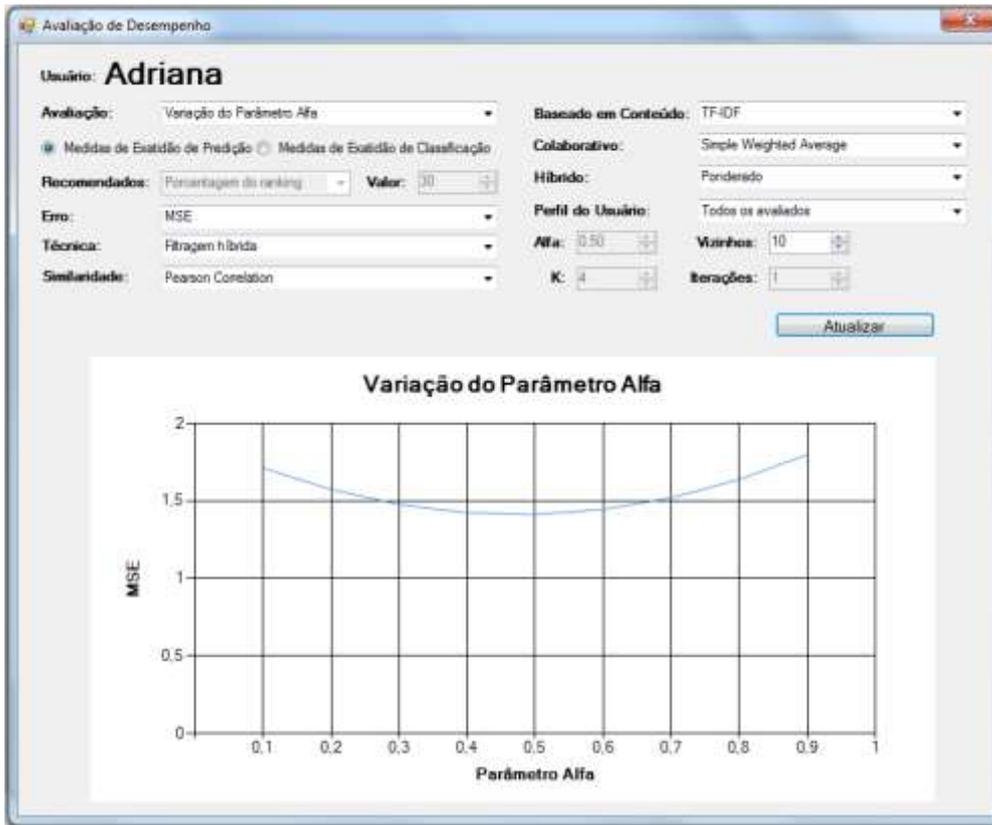


Figura 51 - Experimento 5: MSE da variação do parâmetro α

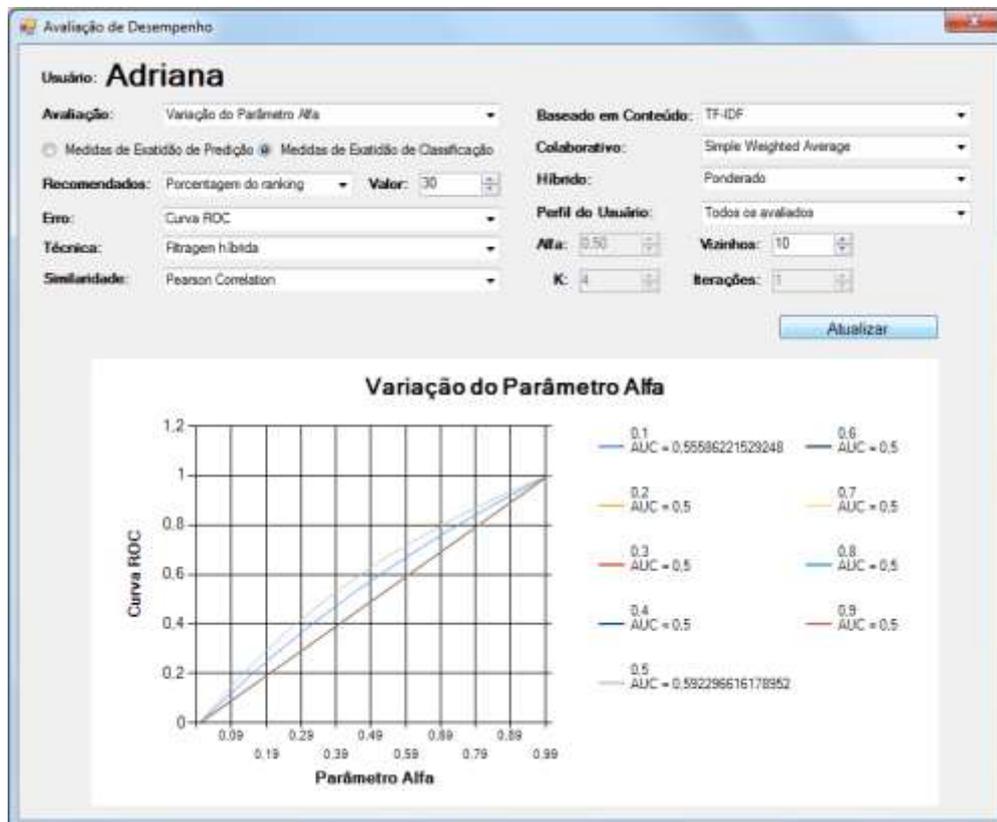


Figura 52 – Experimento 5: Curva ROC da variação do parâmetro α

5.6 Experimento 6 – Algoritmo Híbrido

O sexto experimento objetivou comparar qual das duas abordagens de filtragem híbrida implementadas tinha melhor desempenho, a ponderada (com o valor de α definido no experimento anterior) ou a que também considerava a heurística da quantidade de avaliações. Observando a figura 53 e a figura 54, é possível concluir que a segunda abordagem apresentou os melhores resultados. Isto é ainda mais perceptível ao se considerar um usuário novo no sistema. Supondo que este usuário não tenha avaliado ainda nenhum item, ele não terá um perfil montado para o cálculo da filtragem baseada em conteúdo e não haverá como calcular sua similaridade com outros usuários para a filtragem colaborativa. Neste exemplo a heurística da quantidade de avaliações assume o papel principal na recomendação. Isso não se aplica apenas a usuários que não avaliaram nenhum item, mas também a usuários que avaliaram poucos itens. No caso destes últimos, apesar de haver como montar o seu perfil e calcular a similaridade com os demais usuários, as poucas informações disponíveis no sistema tornariam a recomendação pobre e pouco confiável.

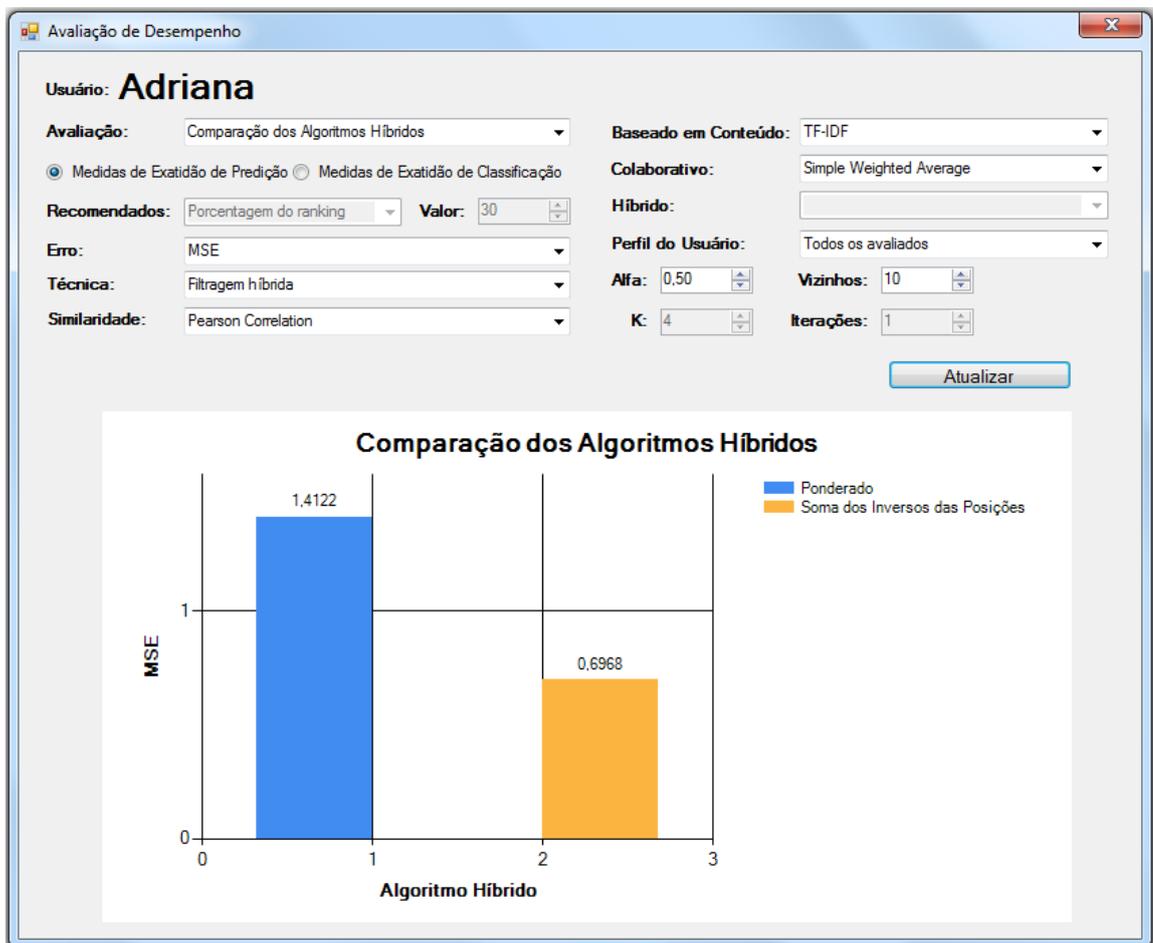


Figura 53 - Experimento 6: MSE dos algoritmos híbridos

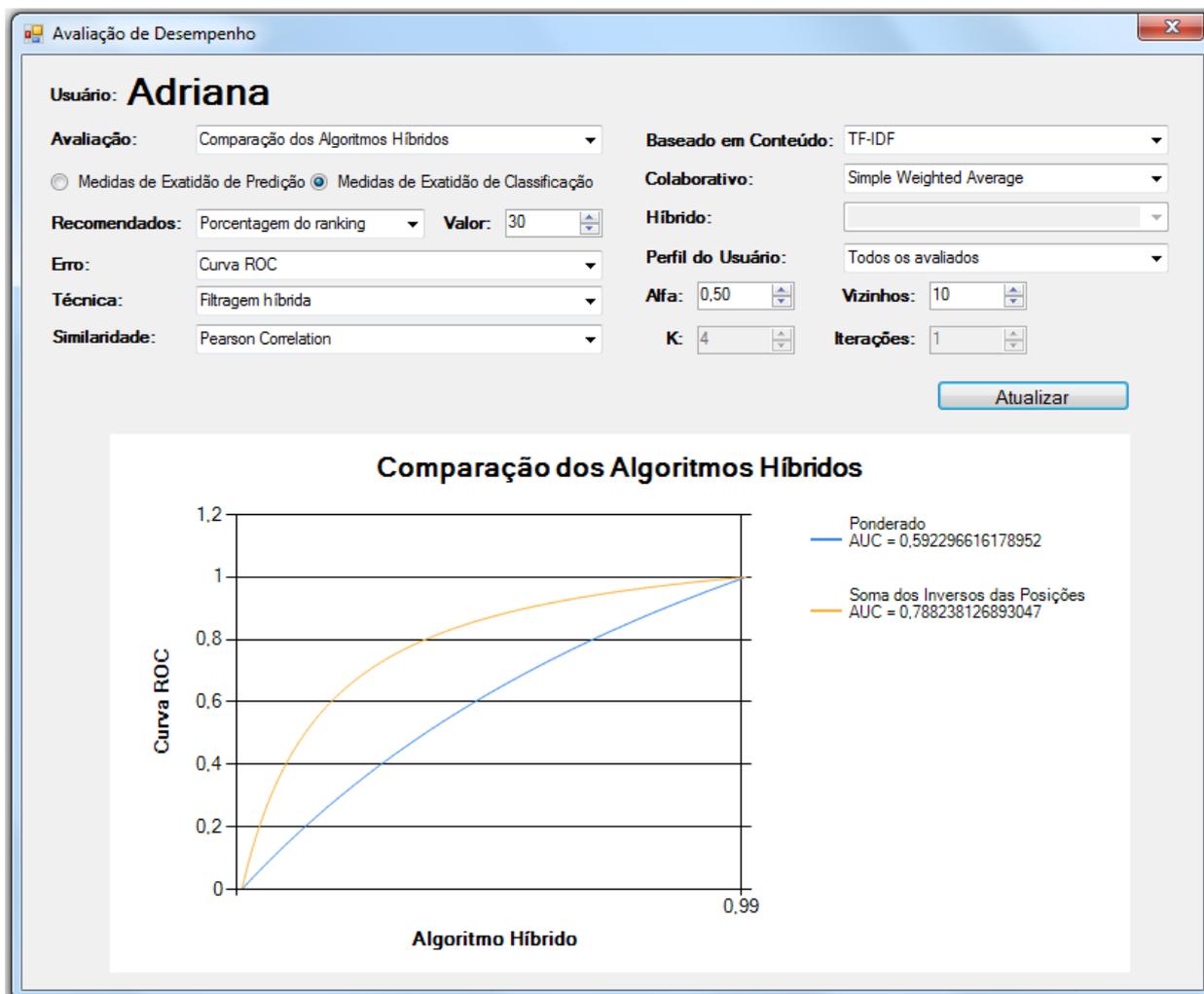


Figura 54 - Experimento 6: Curva ROC dos algoritmos híbridos

5.7 Experimento 7 – Técnica de Filtragem

O último experimento realizado objetivou comparar os resultados obtidos com as técnicas de filtragem baseada em conteúdo, colaborativa e híbrida, a fim de verificar se esta última de fato minimizava as principais desvantagens e mantinha as principais vantagens de suas componentes. Desta vez, além do MSE (figura 55) e da curva ROC (figura 56), foram utilizadas também as métricas MAE (figura 57) e Precisão (figura 58). Após a realização desse experimento, foi possível perceber que de fato a filtragem híbrida apresenta melhores resultados (menor MSE, maior AUC, menor MAE e maior precisão) do que as suas componentes.

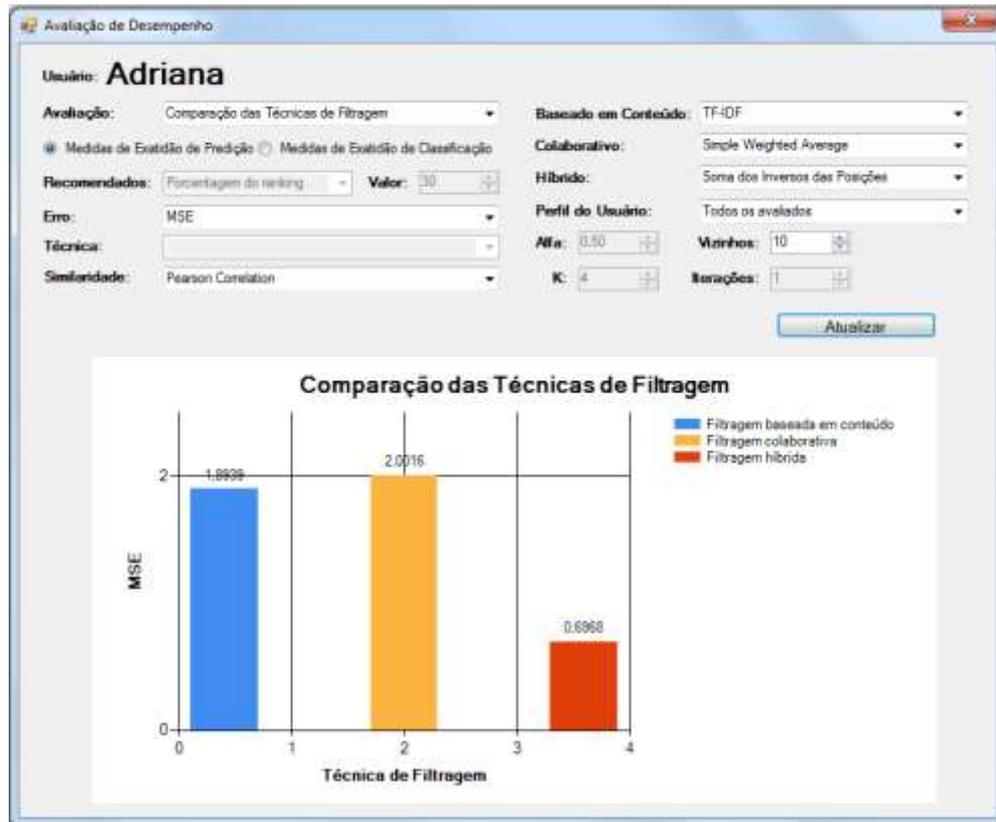


Figura 55 - Experimento 7: MSE das técnicas de filtragem

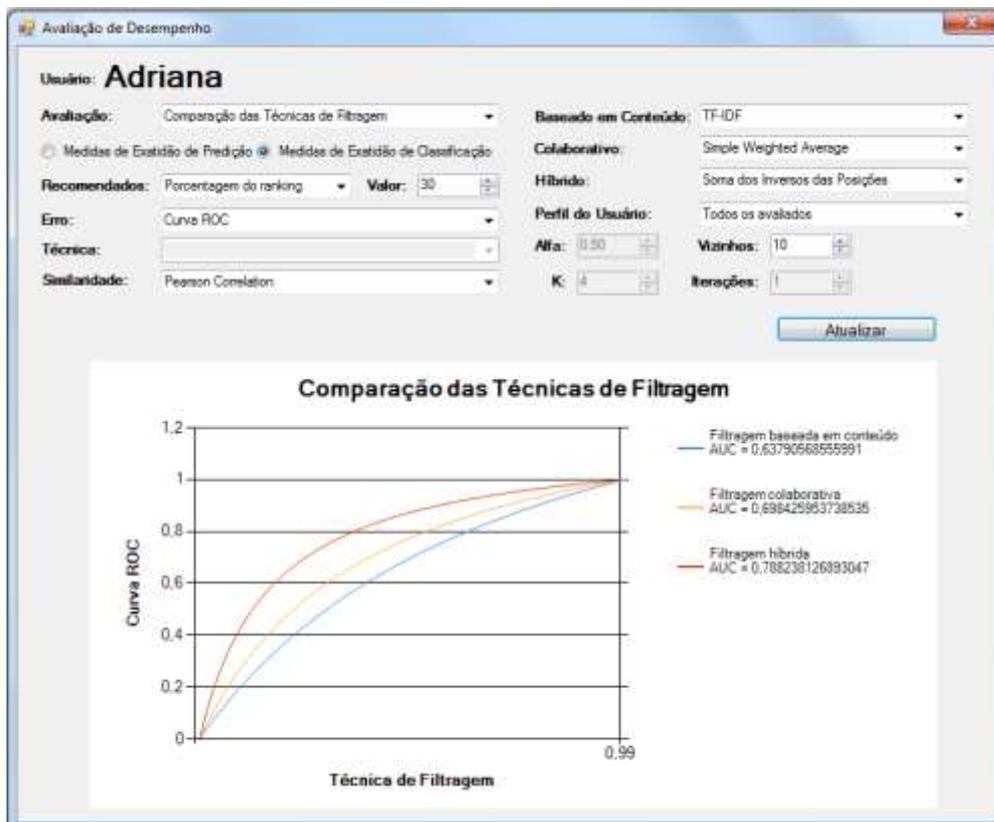


Figura 56 - Experimento 7: Curva ROC das técnicas de filtragem



Figura 57 - Experimento 7: MAE das técnicas de filtragem



Figura 58 - Experimento 7: Precisão das técnicas de filtragem

5.8 Experimento 8 – Problema da Superespecialização

Este experimento objetivou verificar se a técnica de filtragem híbrida resolve o problema da superespecialização apresentado pela técnica de filtragem baseada em conteúdo. Como já foi visto, este problema consiste em serem recomendados apenas itens similares aos já avaliados pelo usuário. A usuária Jhenifer Lamim avaliou apenas um item no sistema, o livro “Os Cães nunca deixam de Amar” (figura 59).

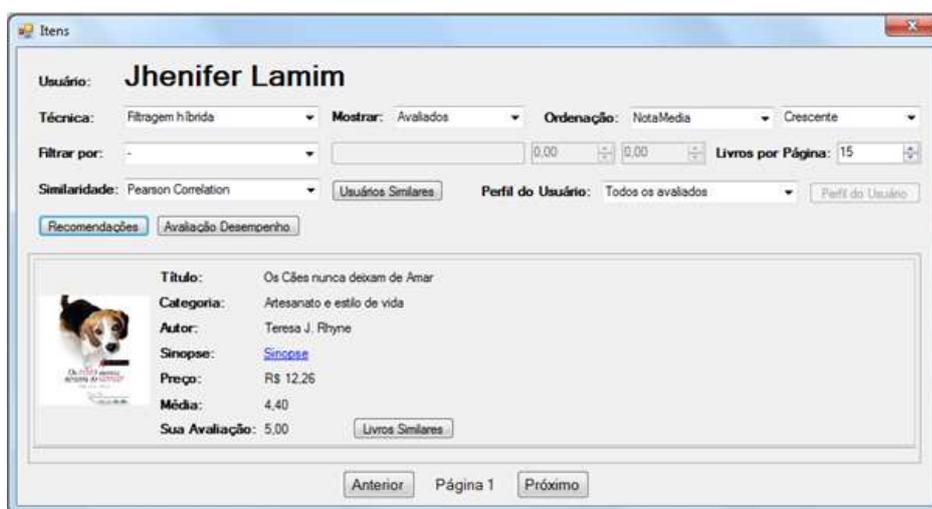


Figura 59 – Experimento 8: Avaliações

Na filtragem baseada em conteúdo lhe são recomendados itens sobre o mesmo assunto (cães), como pode ser visualizado na figura 60.

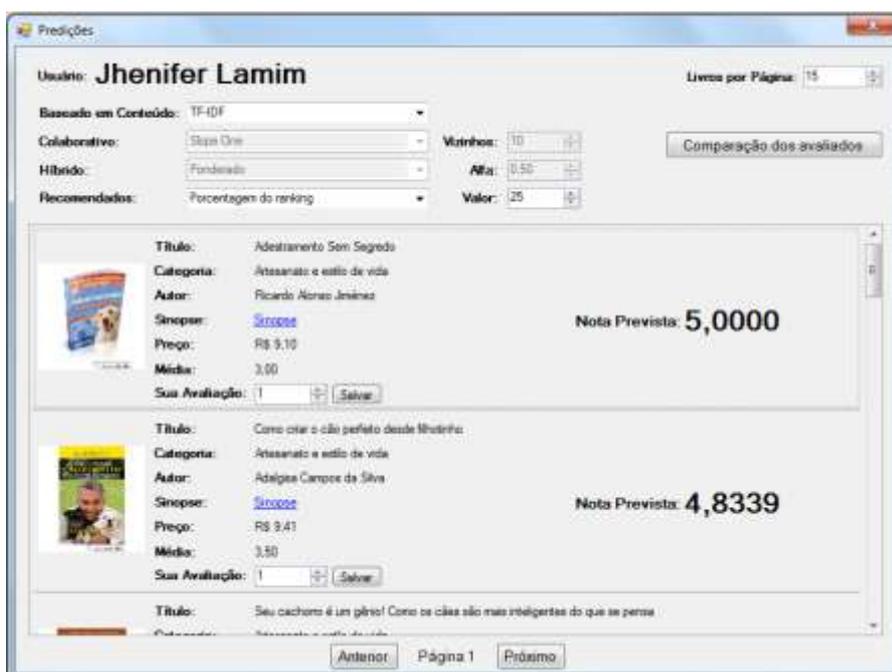


Figura 60 – Experimento 8: Recomendação baseada em conteúdo

Já a filtragem colaborativa permite a recomendação de itens sobre outros assuntos, considerados novidades para o usuário, como pode ser visto na figura 61.

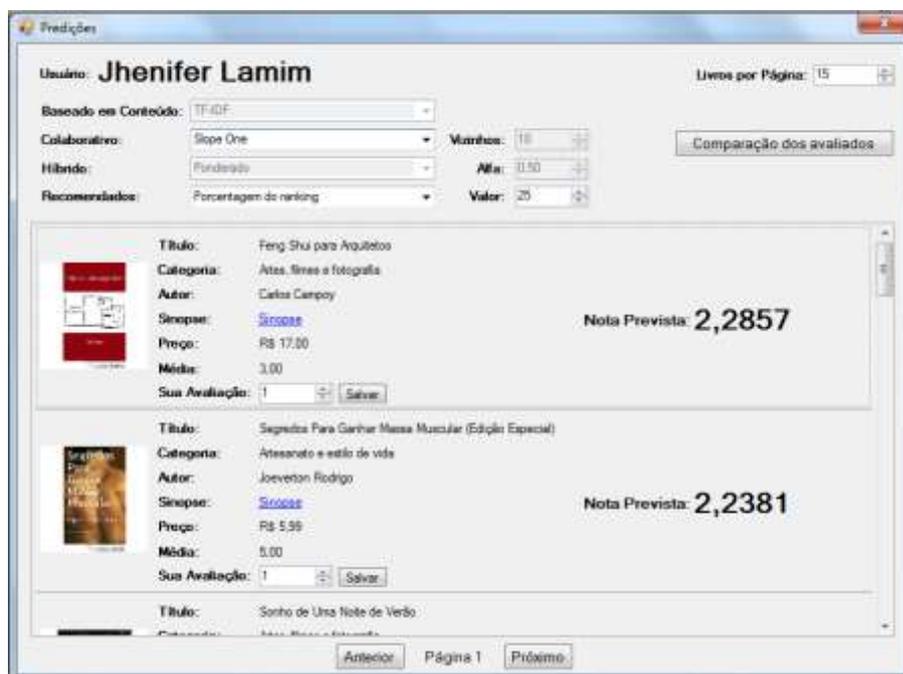


Figura 61 – Experimento 8: Recomendação colaborativa

Devido à sua componente colaborativa, este problema foi minimizado na filtragem híbrida, permitindo assim que sejam recomendadas novidades aos usuários, o que é importante, já que seus interesses podem variar com o tempo (figura 62).

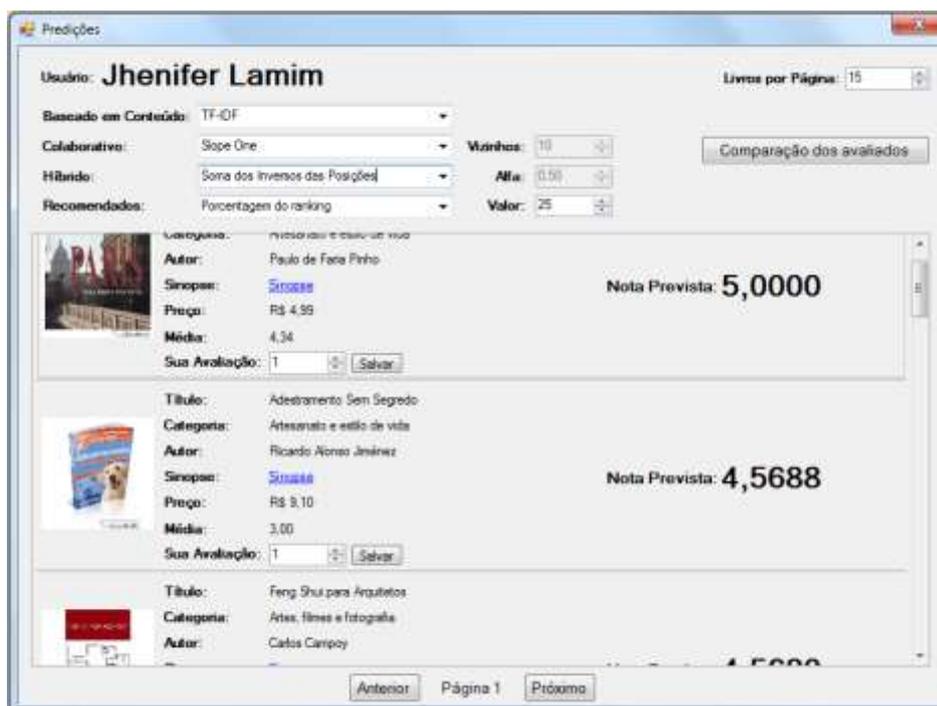


Figura 62 - Experimento 8: Recomendação híbrida

Uma análise mais geral, incluindo outros experimentos, permitiu perceber que a filtragem híbrida também apresenta melhores resultados do que suas componentes diante do problema do novo usuário ou do novo item. Um usuário foi considerado novo neste sistema quando possuísse menos da metade da média do número de avaliações feitas por todos os usuários. De forma similar, um item foi considerado novo no sistema quando possuísse menos da metade da média do número de avaliações recebidas por todos os itens.

O problema do novo item foi minimizado ou resolvido pela parte baseada em conteúdo da filtragem híbrida, já que mesmo que o item não tenha recebido avaliações ele pode ser recomendado, caso seus atributos sejam suficientemente semelhantes aos dos itens já avaliados pelo usuário. Já o problema do novo usuário ocorre nas duas componentes. Nelas o usuário precisa ter avaliado pelo menos um item, a fim de utilizar o valor desta avaliação (filtragem colaborativa) ou o conteúdo do item avaliado (filtragem baseada em conteúdo). A filtragem híbrida com a heurística de também considerar o número de avaliações dos itens apresenta melhores resultados e minimiza este problema, recomendando ao usuário os itens mais populares do sistema.

Apesar de tudo isso, com os testes que foram realizados também percebeu-se que nem sempre é possível realizar boas recomendações, pois existem alguns problemas inerentes aos sistemas de recomendação. Mesmo melhorando os resultados obtidos por suas componentes, a filtragem híbrida ainda apresenta mau uso dos recursos computacionais, o que compromete o tempo de resposta e o uso de memória. Estes problemas acontecem porque a velocidade de resposta e a alocação de recursos de memória são variáveis diretamente proporcionais à quantidade de itens e quantidade de usuários do sistema. Assim, à medida que o número de itens e de usuários cresce, aumentam também os problemas de escalabilidade, desempenho e memória.

6. Considerações Finais

O trabalho aqui descrito se propôs a estudar e analisar, a partir do desenvolvimento de um protótipo de um sistema de recomendação, algumas técnicas de filtragem híbridas, juntamente com suas componentes: a filtragem baseada em conteúdo e a filtragem colaborativa. As tecnologias e a estrutura deste protótipo foram definidas após um intenso trabalho inicial de pesquisa bibliográfica e análise da literatura existente nessa área.

Neste trabalho foram abordadas algumas das principais limitações deste tipo de sistema. Através do estudo e avaliação de algumas medidas de desempenho conhecidas na área de Sistemas de Recomendação, foi possível concluir que a técnica de filtragem híbrida de fato apresenta melhores resultados do que as suas componentes. Isso já era esperado porque estas componentes são, em geral, complementares, de tal forma que os pontos fracos existentes em uma são suprimidos pelas vantagens da outra. Porém, um problema geral dessas medidas de avaliação de desempenho, segundo muitos pesquisadores, é que mesmo que sejam feitas recomendações relevantes e que atendam as preferências dos usuários, os sistemas não lhes oferecem novidades, o que tem se percebido ser um importante fator de fidelidade.

Os sistemas de recomendação têm obtido progressos significativos nos últimos anos, com muitas técnicas e estratégias de recomendação sendo propostas e muitos sistemas comerciais sendo desenvolvidos. Diversas extensões tem sido pesquisadas para melhorar os sistemas de recomendação, dentre elas a incorporação de informação contextual, o suporte a avaliações multicritério, a criação de recomendações mais flexíveis e menos intrusivas, além da criação de métodos que representem melhor o comportamento do usuário e as informações sobre os itens. Entretanto, apesar destes avanços, muitas melhorias ainda são necessárias para deixá-los mais efetivos. Além de envolver estas extensões e métodos, em trabalhos futuros poderiam ser realizados maiores estudos no tocante à melhoria da performance dos algoritmos de similaridade, previsão e recomendação que foram implementados.

Por fim, analisando-se os objetivos que foram traçados no início deste projeto, é possível concluir que eles foram satisfatoriamente atingidos.

Referências Bibliográficas

- [1] ADOMAVICIUS, G.; TUZHILIN, A. **Toward the next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions**. IEEE Transactions on Knowledge and Data Engineering – TKDE, v. 17, n. 6, pp. 734-749, 2005.
- [2] AMAZON, **eBooks da Amazon**, Disponível em <http://www.amazon.com.br/b/ref=s9_hps_gw_brwse?node=6071434011>, Acesso: 24 nov. 2013.
- [3] AVILA, C. M. O. **Desenvolvimento de um Sistema de Recomendação de Artigos Científicos e Avaliação de Métodos de Extração de Palavras-Chave**. 2006. 53 p. Dissertação (Pós-Graduação em Informática) - Universidade Católica de Pelotas, 2006.
- [4] BALABANOVIĆ, M.; SHOHAM, Y. **Fab: content-based, collaborative recommendation**. Communications of the ACM, v. 40, n. 3, pp. 66-72, 1997.
- [5] BELKIN, N. J.; CROFT, W. B. **Information filtering and information retrieval: two sides of the same coin?** Communications of the ACM, v. 35, n. 12, pp. 29-38, 1992.
- [6] BENNET, J.; LANNING, S. **The netflix prize**. In Proceedings of KDD Cup and Workshop, 2007.
- [7] BEZERRA, B.L. D. **Uma solução em filtragem de informação para sistemas de recomendação baseada em análise de dados simbólicos**. 2004. 108 p. Dissertação (Pós-Graduação em Ciência da Computação) - Centro de Informática, Universidade Federal de Pernambuco, 2004.
- [8] BREESE, J. S.; HECKERMAN, D.; KADIE, C. **Empirical analysis of predictive algorithms for collaborative filtering**. Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, pp. 43-52, Morgan Kaufmann Publishers Inc., 1998.
- [9] BURKE, R. **Hybrid Recommender Systems: Survey and Experiments**. Disponível em <<http://josquin.cti.depaul.edu/~rburke/pubs/burke-umuai02.pdf>>. Acesso: 23 nov. 2013.
- [10] CLAYPOOL, M. et al. **Combining content-based and collaborative filters in an online newspaper**. Proceedings of ACM SIGIR workshop on recommender systems, Citeseer, 1999.
- [11] CREMONESI, P.; KOREN, Y.; TURRIN, R. **Performance of recommender algorithms on top-n recommendation tasks**. Proceedings of the fourth ACM conference on Recommender systems, pp. 39-46, ACM, 2010.
- [12] DESHPANDE, M.; KARYPIS, G. **Item-based top-n recommendation algorithms**. ACM Transactions on Information Systems, v. 22, n. 1, pp. 143-177, 2004.
- [13] FILHO, V. M. **e-Recommender: Sistema Inteligente de Recomendação para Comércio Eletrônico**. 2006. 58 f. Monografia apresentada na Universidade de Pernambuco para obtenção do grau de bacharel em Engenharia da Computação.

- [14] GOLDBERG, D. et al. **Using collaborative filtering to weave an information tapestry**. Communications of the ACM, v. 35, n. 12, pp. 61-70, 1992.
- [15] HERLOCKER, J. et al. **Evaluating Collaborative Filtering Recommender Systems**, ACM Transaction on Information Systems, v. 22, n. 1, pp. 5-53, 2004.
- [16] KIM, B. M.; LI, Q.; PARK, C. S. **A new approach for combining content-based and collaborative filters**. Disponível em <<http://fife.swufe.edu.cn/BILab/paper/JIIS-a%20new%20approach.pdf>>. Acesso: 24 nov. 2013.
- [17] KIM, H. N. et al. **Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation**, Electronic Commerce Research and Applications, v. 9, n. 1, pp. 73–83, 2010.
- [18] KONSTAN, J. A. **Introduction to Recommender Systems: Algorithms and Evaluation**. ACM Transactions in Information Systems, v. 22, n. 1, 2004.
- [19] LÁZARO, A. S. **Análise e seleção de algoritmos de filtragem de informação para solução do problema cold-start item**. 2010. 66 p. Monografia apresentada na Universidade Federal de Lavras para obtenção do grau de bacharel em Sistemas de Informação.
- [20] LINDEN, G.; SMITH, B; YORK, J. **Amazon.com recommendations: item-to-item collaborative filtering**, IEEE Internet Computing, pp. 76-80, 2003.
- [21] MIZZARO, S. **Relevance: The whole history**. Journal of the American society for information science, v. 48, n. 9, pp. 810-832, 1997.
- [22] PARK, S. T.; CHU, W. **Pairwise preference regression for cold-start recommendation**. In RecSys, New York, 2009.
- [23] REATEGUI, E. B.; CAZELLA, S. C. **Sistemas de Recomendação**. XXV Congresso da Sociedade de Computação: A Universalidade da Computação – um agente de inovação e conhecimento. Unisinos, São Leopoldo, 43 p, 2005.
- [24] RESNICK, P. et al. **GroupLens: an open architecture for collaborative filtering of netnews**. Proceedings of the 1994 ACM conference on Computer supported cooperative work, pp. 175-186, ACM, 1994.
- [25] RICCI, F. et al. **Recommender Systems Handbook**, Springer, 2010.
- [26] SALTON, G. **Automatic Text Processing: The Transformation, Analysis, and Retrieval of**. Addison-Wesley, 1989. ISBN 0201122278.
- [27] SCHAFER, J. B. et al. **Collaborative filtering recommender systems**. In: (Ed.). The adaptive web, pp.291-324, Springer, 2007. ISBN 3540720782.
- [28] SCHAFER, J. B.; KONSTAN, J. A.; RIEDL, J. **E-commerce recommendation applications**. In: (Ed.). Applications of Data Mining to Electronic Commerce, pp.115-153, Springer, 2001. ISBN 1461356482.

- [29] SCHEIN, A. I. et al. **Methods and metrics for cold-start recommendations**. Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 253-260, ACM, 2002.
- [30] **Sistemas de recomendação**. Disponível em <<http://www.slideshare.net/Marciobds/sistemas-de-recomendao-27410622>>. Acesso: 25 nov. 2013.
- [31] SU, X; KHOSHGOFTAAR, T. M. **A Survey of Collaborative Filtering Techniques**, 19 p. Jun Hong, 2009.
- [32] SWETS, J. A. **Signal Detection Theory and ROC Analysis in Psychology and Diagnostics: Collected Papers**. New Jersey: LEA, 1996.
- [33] XU, R.; WUNSCH II, D. **Survey of clustering algorithms**, IEEE Transactions on Neural Networks, v.16, n.3, pp. 645-678, 2005.
- [34] ZANETTE, L. R. **Sistema de Recomendação de itens baseado na rede de confiança do usuário**. 2008. 178 p. Dissertação (Pós-Graduação em Informática) - Universidade Federal do Rio de Janeiro, 2008.

Apêndice

Prova por Indução

A média aritmética entre n números positivos é sempre maior ou igual à média geométrica destes mesmos números. Como a média geométrica corresponde à raiz enésima do produto entre estes números, temos que o produto total de n números positivos é máximo quando a média geométrica deles é máxima, ou seja, quando ela é igual a média aritmética deles.

$$\frac{a_1 + a_2 + \dots + a_n}{n} \geq \sqrt[n]{a_1 \cdot a_2 \cdot \dots \cdot a_n}$$

Caso base:

$$n = 2$$

$$\frac{a_1 + a_2}{2} \geq \sqrt{a_1 \cdot a_2}$$

$$a_1 + a_2 \geq 2 \sqrt{a_1 \cdot a_2}$$

$$a_1 - 2 \sqrt{a_1 \cdot a_2} + a_2 \geq 0$$

$$\text{Por produtos notáveis: } (\sqrt{a_1} - \sqrt{a_2})^2 \geq 0$$

O que é verdade para todo número real ($x^2 \geq 0$)

Hipótese de Indução:

$$n = k$$

$$\frac{a_1 + a_2 + \dots + a_k}{k} \geq \sqrt[k]{a_1 \cdot a_2 \cdot \dots \cdot a_k}$$

Tese de Indução:

$$n = k - 1$$

$$\frac{a_1 + a_2 + \dots + a_{k-1}}{k - 1} \geq \sqrt[k-1]{a_1 \cdot a_2 \cdot \dots \cdot a_{k-1}}$$

Fazemos

$$\frac{a_1 + a_2 + \dots + a_{k-1}}{k - 1} = p$$

$$\sqrt[k-1]{a_1 \cdot a_2 \cdot \dots \cdot a_{k-1}} = q$$

Assim queremos provar que: $p \geq q$

Adicionando o termo q nas médias aritmética e geométrica dos $n - 1$ termos da tese, passamos a ter k termos:

$$\frac{a_1 + a_2 + \dots + a_{k-1} + q}{k} \geq \sqrt[k]{a_1 \cdot a_2 \cdot \dots \cdot a_{k-1} \cdot q}$$

Como

$$q = \sqrt[k-1]{a_1 \cdot a_2 \cdot \dots \cdot a_{k-1}}$$

$$q^{k-1} = a_1 \cdot a_2 \cdot \dots \cdot a_{k-1}$$

Assim:

$$\frac{a_1 + a_2 + \dots + a_{k-1} + q}{k} \geq \sqrt[k]{q^{k-1} \cdot q}$$

$$\frac{a_1 + a_2 + \dots + a_{k-1} + q}{k} \geq \sqrt[k]{q^k}$$

$$\frac{a_1 + a_2 + \dots + a_{k-1} + q}{k} \geq q$$

Como:

$$p = \frac{a_1 + a_2 + \dots + a_{k-1}}{k-1}$$

Então:

$$p(k-1) = a_1 + a_2 + \dots + a_{k-1}$$

Assim:

$$\frac{p(k-1) + q}{k} \geq q$$

$$p(k-1) + q \geq qk$$

$$p(k-1) \geq qk - q$$

$$p(k-1) \geq q(k-1)$$

$$p \geq q$$

Conforme queríamos demonstrar.

Como a média geométrica é máxima quando é igual à média aritmética, o produto total é máximo nesse caso:

$$\frac{a_1 + a_2 + \dots + a_n}{n} \geq \sqrt[n]{a_1 \cdot a_2 \cdot \dots \cdot a_n}$$

E sabe-se que isso só é verdade quando todos os números positivos em questão são iguais. Assim, quanto maior o produto entre os números positivos melhor distribuídos estão estes valores, sendo igualmente distribuídos quando o produto é máximo.