

UNIVERSIDADE DO ESTADO DE SANTA CATARINA - UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT
MESTRADO EM COMPUTAÇÃO APLICADA

EDUARDO JOSÉ DE BORBA

**SISTEMA DE RECOMENDAÇÃO SENSÍVEL AO TEMPO EM
AMBIENTES VIRTUAIS DE APRENDIZAGEM**

JOINVILLE

2017

EDUARDO JOSÉ DE BORBA

**SISTEMA DE RECOMENDAÇÃO SENSÍVEL AO TEMPO EM
AMBIENTES VIRTUAIS DE APRENDIZAGEM**

Qualificação apresentada ao Programa de Pós-Graduação em Computação Aplicada, da Universidade do Estado de Santa Catarina, como requisito parcial para a obtenção do grau de Mestre em Computação Aplicada.

Orientador: Dra. Isabela Gasparini
Coorientador: Dr. Daniel Lichtnow

JOINVILLE

2017

EDUARDO JOSÉ DE BORBA

**SISTEMA DE RECOMENDAÇÃO SENSÍVEL AO TEMPO EM AMBIENTES
VIRTUAIS DE APRENDIZAGEM**

Qualificação apresentada ao Programa de Pós-Graduação em Computação Aplicada, da Universidade do Estado de Santa Catarina, como requisito parcial para a obtenção do grau de Mestre em Computação Aplicada.

Banca Examinadora

Orientadora:

Dra. Isabela Gasparini
Universidade do Estado de Santa Catarina
(UDESC)

Coorientador:

Dr. Daniel Lichtnow
Universidade Federal de Santa Marina
(UFSM)

Membros:

Dr. Rui Jorge Tramontin Junior	Dr. Thiago T. Primo (UFPel)
Universidade do Estado de Santa Catarina (UDESC)	Universidade Federal de Pelotas (UFPel)

Dr. José Palazzo M. de Oliveira (UFRGS)
Universidade Federal do Rio Grande do
Sul (UFRGS)

Joinville, 11 de Dezembro de 2017

RESUMO

Sistemas de Recomendação (SR) são ferramentas de software que sugerem itens para os usuários de forma automatizada e personalizada, sem a necessidade do usuário formular uma consulta para encontrar os itens do seu interesse. Esses sistemas são explorados em Ambientes Virtuais de Aprendizagem (AVA) com o objetivo de reduzir alguns problemas existentes nesses ambientes quando a quantidade de materiais disponíveis é grande, tais como: sobrecarga cognitiva, dificuldade de encontrar os materiais do seu interesse e muitos materiais nunca serem utilizados. Pesquisadores da área argumentam que os algoritmos de SRs tradicionais não são suficientes para os AVAs, sendo necessário um nível maior de personalização a situação do usuário, como considerar dimensões do contexto. O objetivo desse trabalho é a criação de perfis de usuário que levem em conta a mudança dos interesses destes usuários ao longo do tempo. O algoritmo proposto combina a (1) similaridade do perfil do usuário (representado pelos materiais acessados pelo usuário) com os itens disponíveis para a recomendação com a (2) recência do acesso ou uso desse materiais, além da (3) informação se aquele item disponível para a recomendação já foi acessado ou não. A proposta leva em conta que o ritmo de estudo dos alunos pode ser diferente, portanto a recência é considerada em relação a sequência de itens acessados e não ao tempo absoluto (em segundos) desde o acesso. A proposta desse trabalho será incorporada ao ambiente AdaptWeb[®] e será avaliado através de um minicurso de algoritmos ministrado no ambiente. O algoritmo proposto será comparado a abordagem Baseada em Conteúdo tradicional através de um experimento utilizando uma estratégia *Between Subjects*. O objetivo do experimento é verificar se existe diferença na percepção do usuário sobre a qualidade das recomendações do algoritmo proposto em relação a abordagem Baseada em Conteúdo tradicional.

Palavras-chaves: Sistema de Recomendação; Sensível ao Tempo; Sensível ao Contexto; Ambiente Virtual de Aprendizagem; AdaptWeb[®].

ABSTRACT

Recommender Systems (RS) are software tools that provide items as suggestions to users automatically and personalized to his interests, without the need to formulate a search argument to achieve this. These systems are applied to Virtual Learning Environments (VLE) aiming to reduce some drawbacks existing in these environments when the number of available items is huge, e.g., cognitive overload, difficulty finding items of user's interest or some materials never get used. Researchers in this area argue that traditional RS approaches are not enough for VLE, being required a major level of personalization to user's current context. This work's goal is to create user profiles that take into account the changing of user's interests with the time. These profiles are going to be applied in the recommendation algorithm proposed in this work. The proposed algorithm combines (1) similarity between user profile (represented by materials accessed by the user) with the items available to recommendation with (2) the recency of materials accessed by the user and (3) the information about if the item available to be recommended were accessed or not. The proposal takes into account that each learner can have a different study rhythm, therefore the recency considers the sequence of items accessed and not the absolute time (in seconds) from the access. The proposal of this work will be incorporated to the AdaptWeb[®] environment and will be evaluated through an algorithms course in the environment. The proposal will be compared to the Content-Based traditional approach through an experiment using a Between Subjects strategy. The objective of this experiment is to verify if there are differences in user's perception of quality recommendations between the proposal when compared with Content-Based approach.

Keywords: Recommender System; Time-Aware; Context-Aware; Virtual Learning Environment; AdaptWeb[®].

LISTA DE ILUSTRAÇÕES

Figura 1 – Construtos do framework de avaliação de SRs pela perspectiva do usuário	28
Figura 2 – Estrutura do AdaptWeb®	51
Figura 3 – Proposta de Interface de Recomendação	53
Figura 4 – Explicação da recomendação	54

LISTA DE TABELAS

Tabela 1 – Comparação dos trabalhos relacionados	41
Tabela 2 – Cenário de Uso: comportamento do algoritmo após o acesso de itens relacionados a pilhas.	47
Tabela 3 – Cenário de Uso: comportamento do algoritmo após o acesso de itens relacionados a filas.	47
Tabela 4 – Cenário de Uso: comportamento do algoritmo após o aluno voltar a acessar itens relacionados a pilhas.	48

LISTA DE ABREVIATURAS E SIGLAS

AdaptWeb®	Ambiente de Ensino-Aprendizagem Adaptativo na Web
AVA	Ambiente Virtual de Aprendizagem
CCT	Centro de Ciências Tecnológicas
PPGCA	Programa de Pós-Graduação em Computação Aplicada
SR	Sistema de Recomendação
TCLE	Termo de Consentimento Livre e Esclarecido
UDESC	Universidade do Estado de Santa Catarina
UFRGS	Universidade Federal do Rio Grande do Sul
UFSM	Universidade Federal de Santa Maria
XML	Extensible Markup Language

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Problema	12
1.2	Objetivos	12
1.2.1	Objetivo Geral	12
1.2.2	Objetivos Específicos	12
1.3	Escopo	12
1.4	Metodologia	13
1.5	Estrutura	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Sistemas de Recomendação	14
2.1.1	Baseada em Conteúdo	15
2.1.2	Filtragem Colaborativa	17
2.1.3	Baseada em Conhecimento	19
2.1.4	Híbrida	20
2.2	Sistemas de Recomendação Sensíveis ao Contexto	20
2.3	Sistemas de Recomendação Sensível ao Tempo	22
2.3.1	Restriction	23
2.3.2	Micro-profile	24
2.3.3	Bias	24
2.3.4	Decay	24
2.3.5	Time Rating	25
2.3.6	Novelty	25
2.3.7	Sequence	25
2.4	Projeto de Interface de Recomendações	25
2.5	Avaliação de Sistemas de Recomendação	26
2.6	Considerações sobre o capítulo	28
3	TRABALHOS RELACIONADOS	30
3.1	Fan et al. 2015	30
3.2	Luo et al. 2010	32
3.3	Benčič e Bieliková 2012	34
3.4	Hawalah e Fasli 2014	35
3.5	Qiao e Zhang 2012	37
3.6	Kushwaha et al. 2016	38

3.7	Wei, Khoury e Fong 2013	39
3.8	Considerações sobre o capítulo	40
4	SR SENSÍVEL AO TEMPO	43
4.1	Algoritmo Proposto	43
4.2	Cenário de Uso	46
4.3	Discussão Sobre o Algoritmo Proposto	48
4.4	Considerações sobre o capítulo	49
5	PLANEJAMENTO DO EXPERIMENTO	50
5.1	Descrição do Ambiente AdaptWeb®	50
5.1.1	Estrutura do AdaptWeb®	50
5.1.2	Sistema de Recomendação no AdaptWeb®	51
5.1.3	Apresentação das Recomendações	52
5.2	Definição do experimento	53
5.3	Teste piloto	55
5.4	Considerações sobre o capítulo	56
6	CONSIDERAÇÕES FINAIS	57
6.1	Cronograma	58
6.2	Resultados Parciais	58
	REFERÊNCIAS	60
	APÊNDICES	63
	APÊNDICE A – CRONOGRAMA DAS ATIVIDADES	64
	ANEXOS	65
	ANEXO A – 60 QUESTÕES DO FRAMEWORK DE AVALIAÇÃO DE SISTEMAS DE RECOMENDAÇÃO RESQUE	66
A.1	Quality of Recommended Items	66
A.1.1	Accuracy	66
A.1.2	Relative Accuracy	66
A.1.3	Familiarity	66
A.1.4	Attractiveness	66
A.1.5	Enjoyability	66

A.1.6	Novelty	66
A.1.7	Diversity	67
A.1.8	Context Compatibility	67
A.2	Interaction Adequacy	67
A.3	Interface Adequacy	67
A.4	Perceived Ease of Use	67
A.4.1	Ease of Initial Learning	67
A.4.2	Ease of Preference Elicitation	68
A.4.3	Ease of Preference Revision	68
A.4.4	Ease of Decision Making	68
A.5	Perceived Usefulness	68
A.6	Control/Transparency	69
A.7	Attitudes	69
A.8	Behavioral Intentions	69
A.8.1	Intention to Use the System	69
A.8.2	Continuance and Frequency	69
A.8.3	Recommendation to Friends	69
A.8.4	Purchase Intention	70

1 INTRODUÇÃO

Um Ambiente Virtual de Aprendizagem (AVA) é um ambiente computacional com a finalidade de integrar diversas mídias (e.g., vídeos, apresentações, textos) e dar suporte à educação online (DRACHSLER et al., 2015). Esses ambientes, além de simularem uma sala de aula permitindo o relacionamento professor-aluno e aluno-aluno, disponibilizam conteúdos e materiais para os usuários poderem acessar.

Quando a quantidade de materiais disponíveis nos AVAs é muito grande, existem alguns problemas que podem acontecer. São eles:

- O aluno sofrer uma sobrecarga cognitiva, aumentando o esforço necessário para compreender o ambiente e encontrar os itens de seu interesse, atrapalhando o processo de aprendizagem;
- O aluno não encontrar um material que seja de seu interesse, devido a enorme quantidade de materiais disponíveis;
- Parte do material disponibilizado, que poderia auxiliar os alunos no processo de aprendizagem, nunca ser utilizado.

Com o objetivo de reduzir esses problemas, pesquisadores tem aplicado técnicas de personalização para selecionar os melhores itens para cada estudante, considerando o seu conhecimento, objetivos, preferências e necessidades (BRUSILOVSKY, 1998). Os Sistemas de Recomendação são uma alternativa para reduzir esses problemas, sugerindo itens para o usuário utilizando informações sobre seus interesses e sobre os itens disponíveis (ADOMAVICIUS; TUZHILIN, 2005).

Porém, pesquisadores da área argumentam que no domínio educacional os SRs tradicionais (aqueles que consideram apenas informações sobre os usuários e os itens para recomendar) não são suficientes (VERBERT et al., 2012; DRACHSLER et al., 2015). Verbert et al. (2012) afirmam que nessa área é necessário um nível maior de personalização, como utilizar informações do contexto do usuário na recomendação.

Apesar de existir uma grande quantidade de trabalhos utilizando o contexto em SRs no domínio educacional, como pode ser visto em Verbert et al. (2012) e Drachsler et al. (2015), pouco foi encontrado da aplicação do contexto temporal nesse domínio (BORBA; GASPARINI; LICHTNOW, 2017). O contexto temporal é relevante pois leva em consideração a variação dos interesses do usuário com o passar do tempo. Além disso, os SR Sensíveis ao Tempo demonstraram bons resultados em outros domínios de aplicação (CAMPOS; DÍEZ; CANTADOR, 2014).

1.1 PROBLEMA

Como dito anteriormente, foram encontrados poucos trabalhos sobre o uso do contexto temporal em SRs para AVAs. Além disso, nos trabalhos encontrados as propostas não foram avaliadas em ambientes reais de uso, não sendo possível avaliar o impacto dos SRs Sensíveis ao Tempo nesse domínio. Assim, a pergunta a ser respondida por este trabalho é: **"Como considerar a variação dos interesses do aluno com o passar do tempo em um AVA para recomendar itens para ele?"**.

1.2 OBJETIVOS

Foram definidos objetivos geral e específicos para orientar o processo de pesquisa desse trabalho buscando responder a questão de pesquisa definida acima.

1.2.1 Objetivo Geral

Criar modelos de usuário e algoritmos para sistemas de recomendação voltados a AVAs que levem em conta a mudança dos interesses destes usuários ao longo do tempo.

1.2.2 Objetivos Específicos

- Definir formas de levar em conta aspectos temporais em um algoritmo de recomendação;
- Definir como usar o aspecto temporal em sistemas de recomendação para ambiente educacionais;
- Implementar o algoritmo de recomendação proposto em um AVA;
- Considerar diretrizes de apresentação de recomendação ao desenvolver a interface do sistema de recomendação em um AVA;
- Avaliar a percepção do usuário sobre o algoritmo proposto em um ambiente real de uso.

1.3 ESCOPO

Esse trabalho não irá considerar outras dimensões do contexto além do tempo na recomendação, e dentro do uso do contexto temporal apenas a categoria de *Decay* será aplicada nesse trabalho, que está relacionada a perda de interesse por itens acessados anteriormente. Além disso, a única abordagem de recomendação utilizada

será a Baseada em Conteúdo, apesar de as categorias de Sistemas de Recomendação Sensíveis ao Tempo poderem ser aplicadas em quaisquer abordagens de recomendação. A avaliação do Sistema de Recomendação proposto será feita apenas em ambientes educacionais, mesmo sendo possível aplicá-lo em outros domínios de aplicação. Não será avaliado, nesse trabalho, o impacto da proposta na aprendizagem dos alunos, pois apesar de o objetivo final ser a melhoria da aprendizagem, esse tipo de avaliação exigiria uma comparação de grupos com e sem o uso do SR utilizando-se de algum método confiável de medir a aprendizagem, algo que é ainda muito discutido por pedagogos (LUCKESI, 2014).

1.4 METODOLOGIA

A pesquisa desse trabalho é de natureza Aplicada, pois busca gerar conhecimentos através da implementação e experimentação de SRs em um ambiente real de uso. A abordagem do problema deste trabalho é quantitativa, através de questionários utilizando escala de Likert para quantificar a opinião dos alunos. Os objetivos dessa pesquisa tem caráter Explicativo, visando identificar fatores que influenciam a percepção dos alunos sobre a qualidade das recomendações. O procedimento utilizado para o desenvolvimento dessa pesquisa é Experimental, onde o objeto de estudo é a percepção dos alunos sobre a qualidade das recomendações e a variável é o algoritmo de recomendação utilizado.

1.5 ESTRUTURA

Este trabalho está estruturado da seguinte forma: o Capítulo 2 conceitua os Sistemas de Recomendação (SR), as suas abordagens, as formas de avaliação e a apresentação das recomendações; o Capítulo 3 descreve os trabalhos relacionados que utilizam a categoria *Decay* nas recomendações e compara com a proposta desse trabalho; o Capítulo 4 apresenta em detalhe a proposta desse trabalho; o Capítulo 5 apresenta o experimento utilizado para avaliação dessa proposta. Por último, o Capítulo 6 apresenta as considerações parciais deste trabalho, recapitulando o que foi feito até o momento.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo são apresentados os principais conceitos relacionados a proposta desse trabalho. Primeiramente são apresentados os Sistemas de Recomendação e as suas abordagens tradicionais, seguidos pelos Sistemas de Recomendação Sensíveis ao Contexto e mais especificamente os Sistemas de Recomendação Sensíveis ao Tempo. Em seguida são apresentados aspectos relacionados ao projeto de interfaces de recomendação e formas de avaliação de Sistemas de Recomendação.

2.1 SISTEMAS DE RECOMENDAÇÃO

Sistemas de Recomendação (SRs) se tornaram uma importante área de pesquisa a partir dos anos 90, quando começaram a surgir os primeiros trabalhos na área de filtragem colaborativa (ADOMAVICIUS; TUZHILIN, 2005). Os SRs são ferramentas computacionais que provém sugestões de itens personalizadas aos usuários (RICCI; ROKACH; SHAPIRA, 2011). Isso significa que o usuário recebe como recomendação um conjunto diferente de itens de acordo com as suas preferências e necessidades. Nos últimos anos, o interesse na aplicação de SRs têm crescido fortemente (ADOMAVICIUS; TUZHILIN, 2005; BEEL et al., 2016). Exemplos dessas aplicações são: recomendação de Livros, CDs, DVDs, etc., em sites de *e-commerce* como Amazon e EBAY; recomendações de filmes em sites como MovieLens e Netflix; recomendação de músicas em sites de *streaming* como Last.fm ou Spotify; recomendação de amigos ou de postagens em redes sociais como Facebook ou Twitter; entre outras.

SRs estão representados formalmente na Equação 2.1.

$$F : U \times I \rightarrow R \quad (2.1)$$

Onde F é a função que busca prever o interesse do usuário pelos itens existentes, U representa o conjunto dos usuários, I representa o conjunto dos itens e R representa a lista ordenada dos itens pelo interesse previsto para o usuário ativo (o usuário que irá receber a recomendação). O objetivo do SR então é conseguir prever de maneira mais correta, com as informações disponíveis, os itens que serão de maior interesse do usuário.

Existem duas formas de capturar os interesses do usuário pelos itens acessados dentro do sistema: (1) Explícita, na qual o usuário indica explicitamente o seu interesse pelo item que acabou de acessar, geralmente com uma nota 1 a 5 ou apenas uma indicação de interesse positivo/negativo para o item; (2) Implícita, na qual o

usuário não precisa indicar o seu interesse pelo item, essa informação é capturada implicitamente através do seu comportamento e das suas interações dentro do sistema.

Os SRs podem ser classificados de acordo com a forma como as recomendações são realizadas (abordagem). As principais abordagens citadas na literatura são (TORRES, 2004; ADOMAVICIUS; TUZHILIN, 2005; RICCI; ROKACH; SHAPIRA, 2011; BOBADILLA et al., 2013): Baseada em Conteúdo, Filtragem Colaborativa, Baseada em Conhecimento e Híbrida. Nas subseções a seguir são descritas cada uma dessas abordagens.

2.1.1 Baseada em Conteúdo

Segundo Ricci, Rokach e Shapira (2011), essa é uma abordagem na qual o usuário recebe recomendações de itens similares aos que se interessou no passado. Consiste em avaliar a semelhança entre um item e os interesses do usuário. Os métodos dessa abordagem tentam prever o grau de utilidade de um item para um usuário com base na utilidade que o usuário determinou para os itens similares ao item (ADOMAVICIUS; TUZHILIN, 2005).

A abordagem Baseada em Conteúdo tem suas raízes na Recuperação da Informação (ADOMAVICIUS; TUZHILIN, 2005). Na abordagem Baseada em Conteúdo, tem-se um conjunto de atributos descrevendo um item e um conjunto de atributos descrevendo os gostos e preferências do usuário. A descrição de um item frequentemente é realizada através de palavras-chave definidas automaticamente por meio de algoritmos usados na área de Recuperação da Informação (ADOMAVICIUS; TUZHILIN, 2005). Já a descrição das preferências do usuário, como dito anteriormente, pode ser capturada de duas formas: implícita, através do seu comportamento no ambiente e de itens que acessou; ou explícita, onde o usuário informa suas preferências ao sistema, por exemplo, respondendo a questionários (ADOMAVICIUS; TUZHILIN, 2005). Dessa forma, os SRs de itens textuais (e.g., documentos) são os que mais utilizam a abordagem Baseada em Conteúdo, devido à facilidade da aplicação das técnicas de Recuperação da Informação nesse tipo de item.

Dentro da área de Recuperação da Informação uma forma de medir a similaridade de itens em um SR é o Cosseno. O cálculo da similaridade por Cosseno foi definido por Salton nos anos 60 (SALTON, 1964). Nessa técnica, cada documento é representado por um vetor de termos $\vec{d}_J = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$. Os vetores são dispostos em um espaço vetorial de t dimensões, onde t é o número de termos, e documentos próximos nesse espaço são considerados semelhantes. Para verificar essa proximi-

dade utiliza-se a Equação 2.2 (CHRISTOPHER; PRABHAKAR; HINRICH, 2008).

$$sim(d_1, d_2) = \frac{\sum_{i=1}^t w_{1,i} \times w_{2,i}}{\sqrt{\sum_{i=1}^t w_{1,i}^2 \sum_{i=1}^t w_{2,i}^2}} \quad (2.2)$$

Onde: $sim(d_1, d_2)$ é o resultado da distância dos vetores, variando de $[0, 1]$; $w_{1,i}$ é o termo presente na posição i do item 1; $w_{2,i}$ é o termo presente na posição i do item 2. Por exemplo, se tivermos três vetores: $u = \{1, 1\}$ representando o usuário, $i_1 = \{0, 1\}$ e $i_2 = \{1, 1\}$ representando itens. Os vetores com 1 na primeira posição indicam que o item ou usuário que estão representando possuem o primeiro termo, enquanto o 0 indica que não possuem. O mesmo funciona para a segunda posição em diante. Ao calcular a similaridade entre esses itens, temos $sim(u, i_1) \approx 0.71$ e $sim(u, i_2) = 1$, identificando que o item representado por i_2 é mais similar às preferências do usuário u .

Outra técnica de Recuperação da Informação é o TF-IDF (*Term-Frequency Inverse Document Frequency*). Essa técnica é utilizada para identificar termos importantes em um documento (CHRISTOPHER; PRABHAKAR; HINRICH, 2008) e pode ser utilizada para a descoberta das palavras-chave que descrevem um item. É utilizada a fórmula da Equação 2.3 para o cálculo dos pesos de cada termo do documento (CHRISTOPHER; PRABHAKAR; HINRICH, 2008).

$$tf-idf_{t,d} = tf_{t,d} \times idf_{t,d} \quad (2.3)$$

Onde: $tf-idf_{t,d}$ representa o peso do termo t no documento d ; $tf_{t,d}$ é o número de vezes que o termo t aparece no documento d ; e $idf_{t,d}$ representa o Inverse Document Frequency do termo t , sendo o responsável por identificar termos que aparecem em muitos documentos diferentes (CHRISTOPHER; PRABHAKAR; HINRICH, 2008). Os termos que aparecem em muitos documentos tendem a perder sua importância. O $idf_{t,d}$ é calculado através da Equação 2.4 (CHRISTOPHER; PRABHAKAR; HINRICH, 2008).

$$idf_{t,d} = \log\left(\frac{N}{d_f}\right) \quad (2.4)$$

Onde: N é o número total de documentos em uma coleção; e d_f é o número de documentos onde aparece o termo t .

A principal vantagem da abordagem Baseada em Conteúdo é não necessitar da opinião de outros usuários para a recomendação (RICCI; ROKACH; SHAPIRA, 2011). As principais desvantagens são: a Partida Fria, em que o sistema não terá informações suficientes sobre os usuários novos para realizar uma boa recomendação;

e a Superespecialização, na qual o usuário recebe sempre itens semelhantes aos que já viu (LOPS; GEMMIS; SEMERARO, 2011).

2.1.2 Filtragem Colaborativa

Nessa abordagem o usuário receberá como recomendação itens que usuários com os mesmos interesses que ele se interessaram no passado, ou seja, é a automação do processo de "boca-a-boca"(JANNACH et al., 2010). A técnica de Filtragem Colaborativa tenta prever a utilidade do item para o usuário, com base na utilidade do mesmo produto para um conjunto de usuários possuidores de características semelhantes às suas (JANNACH et al., 2010).

Existem duas variações básicas da Filtragem Colaborativa: Usuário-Usuário, onde a similaridade entre os usuários é analisada; Item-Item, onde a similaridade entre itens a serem recomendados é analisada (JANNACH et al., 2010).

Para Torres (2004), que considera a variação Usuário-Usuário, a Filtragem Colaborativa ocorre, resumidamente, da seguinte forma:

1. As opiniões das pessoas sobre itens são armazenadas;
2. Baseado nessas opiniões, pessoas com perfil semelhantes (vizinhos) são agrupados;
3. Itens bem avaliados pelos vizinhos são recomendados ao usuário.

Existem duas estratégias para medir a similaridade entre os usuários: Coeficiente de Pearson e Cosseno (TORRES, 2004). Levando em consideração que os usuários são representados pelas notas que deram aos itens, utiliza-se um cálculo matemático para medir a similaridade entre o perfil dos usuários (TORRES, 2004).

O Coeficiente de Pearson é um coeficiente bastante utilizado em modelos econômicos e mede a força do relacionamento de duas variáveis (TORRES, 2004). Esse coeficiente varia no intervalo $[-1, 1]$, sendo -1 indica ausência de correlação e $+1$ indica forte correlação. O cálculo é então feito de acordo com a Equação 2.5 (TORRES, 2004).

$$w_{a,u} = \frac{\sum_{i=1}^m (r_{a,i} - \bar{r}_a) * (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^m (r_{a,i} - \bar{r}_a)^2} \sqrt{\sum_{i=1}^m (r_{u,i} - \bar{r}_u)^2}} \quad (2.5)$$

Na fórmula, $w_{a,u}$ representa a correlação entre o usuário u e um determinado usuário a , onde: $r_{a,i}$ é a avaliação do usuário a para o item i ; \bar{r}_a é a média de todas as avaliações do usuário a ; $r_{u,i}$ é a avaliação do usuário u para o item i ; \bar{r}_u é a média de

todas as avaliações do usuário u . A similaridade é calculada apenas com itens que os dois usuários avaliaram.

Com o aumento da quantidade de usuários e de itens, se torna um desafio para a Filtragem Colaborativa Usuário-Usuário realizar uma recomendação, principalmente pela dificuldade de identificar a vizinhança com tantos usuários (JANNACH et al., 2010). A estratégia Item-Item é uma solução para ser utilizada nesse contexto, permitindo a computação das similaridades a acontecer *off-line* (JANNACH et al., 2011). A ideia principal da estratégia Item-Item é prever a nota que o usuário daria para um item com base nas notas que ele deu para itens semelhantes àquele. Para essa estratégia, o cálculo da similaridade pelo Cosseno, semelhante ao já citado, é uma métrica padrão e a que apresenta os melhores resultados (JANNACH et al., 2010). Esse cálculo da similaridade, ao invés de comparar as notas de cada um dos usuários, considera vetores com as notas de cada item para identificar essa similaridade.

As pessoas que apresentaram preferências similares no passado tendem a concordar no futuro (RICCI; ROKACH; SHAPIRA, 2011). Por isso essa abordagem tende a realizar recomendações que serão bem aceitas pelos usuários.

Como essa abordagem não considera a descrição dos itens e sim as notas desses, uma vantagem dessa abordagem é que as recomendações realizadas podem ser bastante interessantes e inesperadas ao usuário (RICCI; ROKACH; SHAPIRA, 2011).

Por outro lado, a abordagem colaborativa também possui a desvantagem da Partida Fria. Existem dois tipos de Partida Fria nessa abordagem (ADOMAVICIUS; TUZHILIN, 2005): a Partida Fria do Usuário e do Item. A Partida Fria do Usuário é a dificuldade que o sistema encontra para recomendar um item para um usuário que não avaliou nenhum item ainda. A Partida Fria do Item ocorre para um novo item no sistema, que não será recomendado enquanto não for avaliado por algum usuário.

Além disso, outras desvantagens são (ADOMAVICIUS; TUZHILIN, 2005):

- Esparsidade: quanto maior a quantidade de usuários e de itens disponíveis, mais esparsa ficará a tabela com as notas dos usuários e mais difícil será realizar as comparações. Pode ser difícil prever com precisão usuários com os mesmos gostos, pois cada usuário poderá avaliar conjuntos muito diferentes de itens;
- Necessidade de uma comunidade de usuários ativa: para essa abordagem é necessário ter uma grande quantidade de usuários ativos no sistema ao mesmo. No caso de um sistema com poucos usuários pode acontecer também a esparsidade pois os usuários acessam e avaliam itens diferentes e não é possível calcular a similaridade entre eles;

- Ovelha Cinza: para usuários que possuem gostos distintos demais, se torna um desafio realizar recomendações interessantes para ele. O principal motivo é que o sistema não consegue definir outros usuários semelhantes a ele para gerar recomendações;
- Escalabilidade: com o aumento do número de usuários, o custo computacional se torna alto;
- Confiabilidade: essa abordagem é dependente da confiabilidade das avaliações realizadas pelos usuários, se estas forem realizadas de forma incorreta irão diminuir a eficiência da abordagem. Outra coisa a ser considerada é a reputação dos usuários: usuários com maior reputação podem ter suas avaliações mais consideradas (maior peso) que as avaliações de outros usuários.

2.1.3 Baseada em Conhecimento

A abordagem Baseada em Conhecimento recomenda itens aos usuários com base no conhecimento que o sistema possui sobre como características de um item se encaixam nas necessidades de um usuário e o quão útil esse item será (RICCI; ROKACH; SHAPIRA, 2011). Geralmente são utilizadas formas de representar esse conhecimento que sejam de fácil interpretação por computadores, como Ontologias por exemplo (BURKE, 2002). O sistema então recebe como entrada a descrição das necessidades e interesses do usuário e o papel do sistema é realizar uma combinação entre essas necessidades e os itens.

Os SRs Baseado em Caso (Case-Based) são um exemplo de SR da abordagem Baseada em Conhecimento. Nesse sistema uma função de similaridade estima o quanto a necessidade de um usuário (descrição de um problema) combina com uma determinada recomendação (solução do problema) (RICCI; ROKACH; SHAPIRA, 2011). Essa similaridade é o grau de utilidade da recomendação.

Outro exemplo da abordagem Baseada em Conhecimento são os SR Baseados em Restrição. Nessa abordagem os itens que não atendam a certas restrições são automaticamente eliminados dos itens a serem recomendados. Segundo Ricci, Rokach e Shapira (2011), a principal diferença entre um SR Baseado em Caso e um Baseado em Restrição está no fato de o Baseado em Caso considerar a similaridade entre as necessidades do usuário e o item enquanto a baseada em restrições possui regras específicas para tratar cada uma das necessidades do usuário.

A abordagem Baseada em Conhecimento costuma funcionar melhor que outras (e.g., Filtragem Colaborativa ou Baseada em Conteúdo) no início do desenvolvimento, porém se ela não for equipada com a capacidade de aprender mais sobre o usuário, ela será rapidamente ultrapassada por métodos mais simples que explo-

ram a interação do usuário com o sistema (RICCI; ROKACH; SHAPIRA, 2011). Essa abordagem é empregada em conjunto com as outras abordagens com o objetivo de aprimorar a qualidade das recomendações (BURKE, 2002).

2.1.4 Híbrida

Essa abordagem utiliza uma combinação das diversas abordagens para recomendar itens ao usuário. O objetivo é reunir as vantagens das abordagens e tentar eliminar suas desvantagens (BURKE, 2002). Alguns exemplos de algoritmos que utilizam a abordagem híbrida foram dados por Burke (2002):

- *Weighted*: a recomendação é o resultado da execução das abordagens de recomendação em conjunto. Essas abordagens podem ser executadas linearmente, uma após a outra, para definir os melhores itens a serem recomendados, ou cada abordagem pode ter pesos diferentes, tornando o resultado de um mais importante que o resultado do outro.
- *Switching*: ocorre uma alternância entre as abordagens, em certos momentos uma delas é utilizada e em outros momentos outra é utilizada. O sistema deverá possuir alguns critérios para definir qual abordagem irá utilizar.
- *Mixed*: as mencionadas são utilizadas separadamente e os resultados aparecem em um mesmo ranking. Esse tipo de abordagem é utilizado quando se deseja realizar um grande número de recomendações diferentes simultaneamente.
- *Feature combination*: considera as informações da colaboração como uma característica e utiliza a abordagem Baseada em Conteúdo para realizar a recomendação.
- *Cascade*: uma abordagem é utilizada primeiro para gerar um ranking e outra abordagem refina o resultado dado por esta.
- *Feature augmentation*: uma abordagem é utilizada para produzir um ranking ou uma classificação para cada item e o resultado será considerado na execução de outra abordagem.

2.2 SISTEMAS DE RECOMENDAÇÃO SENSÍVEIS AO CONTEXTO

SRs tradicionais consideram apenas as relações entre os usuários e os itens para recomendar, mas não consideram o contexto em que os usuários estão. De acordo com Dey (2001), o contexto é qualquer informação que pode ser usada para caracterizar a situação de uma entidade. As principais entidades em SRs são o usuário que irá receber uma recomendação e os itens que serão recomendados.

SRs Sensíveis ao Contexto estão formalmente definidos na Equação 2.6.

$$F : U \times I \times C \rightarrow R \quad (2.6)$$

Onde F é a função que prediz o interesse em um item ainda não utilizado pelo usuário, U representa o conjunto dos usuários, I representa o conjunto dos itens, C representa o contexto da interação e R representa o conjunto de itens ordenado pelo interesse previsto do usuário para os itens disponíveis.

Vários autores definem conjuntos de dimensões que podem representar o contexto (SCHILIT; ADAMS; WANT, 1994; CHEN; KOTZ et al., 2000; ZIMMERMANN; LORENZ; OPPERMAN, 2007) e que diferem pouco entre si. Nesse trabalho, adotou-se a definição de Schmidt, Beigl e Gellersen (1999), que é uma das mais completas encontradas. Os autores definem essas sete dimensões para representar o contexto (SCHMIDT; BEIGL; GELLERSEN, 1999):

- Informações sobre o usuário, e.g., hábitos do usuário, estado emocional, etc.;
- Ambiente social do usuário, e.g., co-localização com outros usuários, interação em redes sociais, etc.;
- Tarefas do usuários, e.g., objetivos gerais, se é uma tarefa definida previamente (pelo professor, por exemplo) ou aleatória, etc.;
- Localização, e.g., posição absoluta, se o usuário está em casa, no trabalho ou na universidade, etc.;
- Condições do ambiente, e.g., barulho, luminosidade, etc.;
- Infraestrutura, e.g., velocidade da internet, tipo de dispositivo utilizado, etc.;
- Tempo, e.g., *timestamp* de ocorrência de uma interação, dia da semana no qual o usuário pede uma recomendação, etc.

Sobre a aplicação do contexto em SRs, Adomavicius e Tuzhilin (2011) definem três paradigmas de uso das dimensões do contexto no processo de recomendação:

- Pré-Filtragem Contextual, na qual o contexto filtra os dados que representam o usuário e esses dados servem como entrada para um algoritmo tradicional de recomendação;
- Pós-Filtragem Contextual, na qual uma abordagem tradicional de recomendação é utilizada para gerar uma lista de itens a ser recomendados e depois esses itens são filtrados de acordo com o contexto do usuário;

- Modelagem Contextual, na qual o contexto é aplicado diretamente no algoritmo de recomendação, gerando um algoritmo diferente dos tradicionais.

Verbert et al. (2012) dizem que, em ambientes educacionais, as abordagens tradicionais de SRs não são suficientes para recomendar de forma apropriada para os estudantes, porque esse domínio oferece algumas características específicas que não são cobertas por essas abordagens. Por exemplo, é muito mais perigoso recomendar um item ruim para um estudante, que pode desmotivá-lo nos seus estudos, do que recomendar um produto ruim em um site de *e-commerce*. De acordo com Verbert et al. (2012) esse domínio requer um nível maior de personalização.

Aplicar algumas dimensões do contexto é uma alternativa para melhorar a personalização em ambientes educacionais, recomendando materiais adequados para a situação atual do usuário. Por exemplo, considerar o histórico de aprendizagem do aluno, as condições do ambiente e a acessibilidade dos recursos (VERBERT et al., 2012).

Na Seção 2.3 é apresentado um tipo específico de SRs Sensíveis ao Contexto que utilizam a dimensão temporal para recomendar, chamados de SRs Sensíveis ao Tempo. Esse tipo de SR pode também aplicar outras dimensões do contexto em conjunto a questão temporal.

2.3 SISTEMAS DE RECOMENDAÇÃO SENSÍVEL AO TEMPO

Dentre as dimensões do contexto citadas na seção 2.2, o tempo tem uma vantagem de ser fácil de capturar, considerando que praticamente todos os dispositivos tem um relógio que pode capturar o tempo no qual alguma interação ocorreu. Além disso, trabalhos na área demonstraram que o contexto temporal tem potencial para melhorar a qualidade das recomendações (CAMPOS; DÍEZ; CANTADOR, 2014). Esse tipo de SR é chamado de SR Sensível ao Tempo.

SRs Sensíveis ao Tempo estão formalmente definidos na Equação 2.7.

$$F : U \times I \times T \rightarrow R \quad (2.7)$$

Onde F é a função que prediz o interesse do usuário por item ainda não utilizado, U representa o conjunto de usuários, I representa o conjunto de itens, T representa o contexto temporal e R representa o conjunto de itens ordenado pelo interesse previsto do usuário para os itens disponíveis.

De acordo com o dicionário Michaelis (2011), o tempo é um "Período de momentos, de horas, de dias, de semanas, de meses, de anos etc. no qual os eventos se

sucedem, dando-se a noção de presente, passado e futuro". Com essa informação é possível para um sistema computacional estabelecer uma ordem para os eventos que ocorrem.

O Tempo pode ser representado de uma variável contínua ou categórica. A representação contínua utiliza o exato momento em que os itens foram consumidos/avaliados (CAMPOS; DÍEZ; CANTADOR, 2014), por exemplo: "8 de outubro de 2017, 16:15:03". Enquanto na representação categórica as variáveis são calculadas em relação a períodos de interesse (CAMPOS; DÍEZ; CANTADOR, 2014), e.g., Dias da semana: Domingo, Segunda, Terça, ... ou Estações do ano: Primavera, Verão, Outono, Inverno. Além disso, o tempo pode ser representado por diferentes unidades de tempo, e.g., segundos, minutos, horas, meses, anos, etc., e as unidades tempo são hierárquicas, e.g., um dia tem 24 horas, uma hora tem 60 minutos e 1 minuto tem 60 segundos.

Um mapeamento sistemático foi conduzido sobre os SR Sensíveis ao Tempo (BORBA; GASPARINI; LICHTNOW, 2017) utilizando a metodologia de Petersen et al. (2008). Esse mapeamento sistemático não foi restringido apenas trabalhos na área educacional. A principal questão de pesquisa desse mapeamento foi: Como o contexto temporal é utilizado em SRs Sensíveis ao Contexto? Para responder a essa questão de pesquisa principal, três questões de pesquisa secundárias foram definidas, são elas: (1) Como os algoritmos de recomendação utilizam o tempo? (2) Qual é a diferença entre o uso do tempo em diferentes domínios de aplicação? (3) Que outras dimensões são utilizadas juntamente com o contexto temporal?

Após o processo de seleção dos artigos, 88 trabalhos fizeram parte do estudo e foram considerados para responder as questões de pesquisa. Entre os resultados do mapeamento sistemático desenvolvido em Borba, Gasparini e Lichtnow (2017), o principal foi a definição de sete categorias de SRs Sensíveis ao Tempo. Essa categorização foi feita a partir do agrupamento dos artigos que utilizam o tempo de forma semelhante. A partir disso foi possível identificar as sete principais formas de utilizar o tempo nos algoritmos de recomendação que são descritas nas próximas subseções.

2.3.1 Restriction

Na categoria *Restriction*, o tempo é utilizado para restringir que itens serão utilizados. Isso significa que o SR compara variáveis de tempo relacionadas aos itens e ao usuário para restringir quais itens irão aparecer na lista de recomendações. Existe pelo menos duas formas de restrição para se utilizar: (1) o SR compara o tempo disponível pelo usuário com o tempo necessário para consumir um determinado item, e.g., a duração dos filmes que serão recomendados e o tempo que o usuário tem até o seu próximo compromisso; (2) o SR compara o tempo atual (data e hora) com o

horário de funcionamento dos itens que serão recomendados, e.g., na recomendação de restaurantes onde só faz sentido recomendar locais que estejam servindo no momento.

2.3.2 Micro-profile

Na categoria *Micro-Profile*, o usuário possui perfis distintos para cada período de tempo. Nessa categoria, o tempo deve ser utilizado de forma categórica, onde as categorias que serão utilizadas dependem da aplicação onde for aplicada. É possível, por exemplo, que o usuário possua um perfil para dias da semana e outro perfil para finais de semana, ou então um perfil para a manhã, outro para a tarde e outro para a noite. O objetivo é que as recomendações serão realizadas considerando apenas as interações do usuário que aconteceram no mesmo contexto temporal em que ele está no momento, e.g., recomendar programas de TV para o usuário em um domingo a noite considerando apenas quais programas ele costuma acessar em um domingo a noite.

2.3.3 Bias

Na categoria *Bias*, o tempo é utilizado para agregar informação na matriz Usuários x Itens normalmente utilizada pela Filtragem Colaborativa. Essa matriz é comumente utilizada com apenas duas dimensões que são os Usuários e o Itens e os valores dessa matriz são as notas dadas pelos usuários para os itens. Ao incorporar o tempo nessa matriz, é possível realizar uma comparação mais precisa entre os usuários do sistema e assim prever o interesse do usuário ativo para os itens ainda não acessados. Dessa forma, usuários que avaliaram os mesmos itens com notas semelhantes e em contextos temporais semelhantes serão considerados vizinhos do usuário ativo e o algoritmo de recomendação tem uma maior chance de acertar nos interesses do usuário.

2.3.4 Decay

Na categoria *Decay*, o tempo é utilizado como um fator de decaimento na importância das interações do usuário, i.e., interações (itens consumidos, avaliações, etc.) mais antigas tem um peso menor para o algoritmo de recomendação do que as interações mais atuais. Os algoritmos dessa categoria consideram que o interesse do usuário varia com o tempo e é importante considerar que os interesses mais atuais do usuário representam melhor o seu perfil do que interesses mais antigos. É importante notar que as interações antigas não são ignoradas pelo algoritmo de recomendação com *Decay*, é apenas dado um peso menor para essas interações.

2.3.5 Time Rating

Na categoria *Time Rating*, o tempo é considerado pelo SR para inferir as preferências do usuário. Nessa categoria o SR utiliza uma estratégia implícita para capturar o interesse do usuário que considera o tempo que o usuário passou em determinado item. A categoria toma como princípio que itens no qual o usuário passou pouco tempo não são do seu interesse, enquanto itens em que ele passou mais tempo indicam os seus interesses. Essa forma de capturar é interessante pois o usuário não precisa explicitamente dar notas aos itens, dessa forma é possível capturar um feedback do usuário para todos os itens acessados por ele.

2.3.6 Novelty

Na categoria *Novelty*, o SR considera que itens mais novos serão mais relevantes para os usuários do que itens mais antigos. Nessa categoria, existem pelo menos duas estratégias que podem ser utilizadas: (1) o SR possui uma idade limite definida (por exemplo, duas semanas) e itens que sejam mais velhos que isso serão retirados da lista de recomendação; (2) o SR não ignora itens antigos, porém os itens novos possuem um peso maior e, se dois itens similares estiverem para ser recomendados, o mais novo é o escolhido mesmo que o mais antigo esteja mais de acordo com o perfil do usuário. Essa categoria é mais comum em domínios onde novos itens tendem a ser mais relevantes que itens antigos, e.g., redes sociais, notícias, etc.

2.3.7 Sequence

Na categoria *Sequence*, o SR observa itens que são geralmente consumidos juntos em uma determinada ordem e utiliza essa informação para recomendar. Dessa forma, quando o SR encontra um padrão nos acessos de um usuário que já é conhecido, é possível utilizar os próximos itens da sequência como recomendações para o usuário. Essa categoria que os usuários tendem a seguir algum padrão de acesso (trajetória) enquanto interagem com o sistema.

2.4 PROJETO DE INTERFACE DE RECOMENDAÇÕES

No trabalho de Pu, Chen e Hu (2012) os autores argumentam que apenas a eficiência do algoritmo não garante que o usuário estará satisfeito com o sistema, será leal e continuará utilizando-o ou que os itens serão "convertidos" (nesse sentido, os autores se referem a conversão como a aceitar a recomendação dada e utilizar/comprar/assistir/etc. o item recomendado). Os autores afirmam que a percepção do usuário sobre a qualidade da recomendação é afetada tanto pela qualidade das recomendações, que é responsabilidade do algoritmo de recomendação, quanto pela eficiência

na apresentação das recomendações, explicando a razão daquelas recomendações e inspirando a confiança do usuário nas suas decisões. Para isso, os autores defendem uma avaliação do SR pela perspectiva usuário, de forma a avaliar não somente o algoritmo de recomendação mas o SR como um todo (PU; CHEN; HU, 2012).

Além disso, Pu, Chen e Hu (2012) definem um conjunto de vinte diretrizes para o design de um SR bem aceito pelos usuários. Essas diretrizes foram criadas a partir da combinação do resultado de vários trabalhos que executaram experimentos com participação de usuários (i.e., Estudos com usuários) para avaliar a interface de SRs. As principais diretrizes levadas em conta por esse trabalho são (PU; CHEN; HU, 2012):

- Diretriz 14: Considere aprimorar a acurácia percebida pelo usuário com um *layout* mais atrativo, rótulos mais efetivos, e explicando como o sistema gerou as recomendações. Fazendo isso pode aumentar a percepção do usuário sobre a eficiência do sistema, sua satisfação com o sistema em geral, sua prontidão para aceitar os itens recomendados e a sua confiança no sistema.
- Diretriz 18: Considere fornecer como recomendação itens compatíveis ao contexto do usuário. Essa característica pode estar altamente relacionada com a percepção de utilidade do sistema e da satisfação do usuário.
- Diretriz 19: Considere explicar porque o sistema recomendou determinados itens. Esses aspectos podem estar altamente relacionados com a satisfação do usuário, a percepção de controle, as intenções do usuário inspiradas pela confiança do usuário, como a intenção de retornar ao sistema.
- Diretriz 20: Considere fornecer informação suficiente relacionadas aos itens recomendados, controlar a qualidade das informações e da estrutura de navegação.

2.5 AVALIAÇÃO DE SISTEMAS DE RECOMENDAÇÃO

A avaliação de SR são divididas em três categorias (SHANI; GUNAWARDANA, 2011):

- *Offline*: avaliação do método de recomendação através de uma base de dados, simulando as ações dos usuários sem necessitar da participação dos mesmos. Essa avaliação geralmente utiliza uma estratégia onde a base de dados é separada em base de treinamento e base de testes. A base de treinamento terão as notas dadas pelo usuário que serão repassadas ao algoritmo de recomendação como forma de construir o perfil. A base de teste contém os itens para os quais

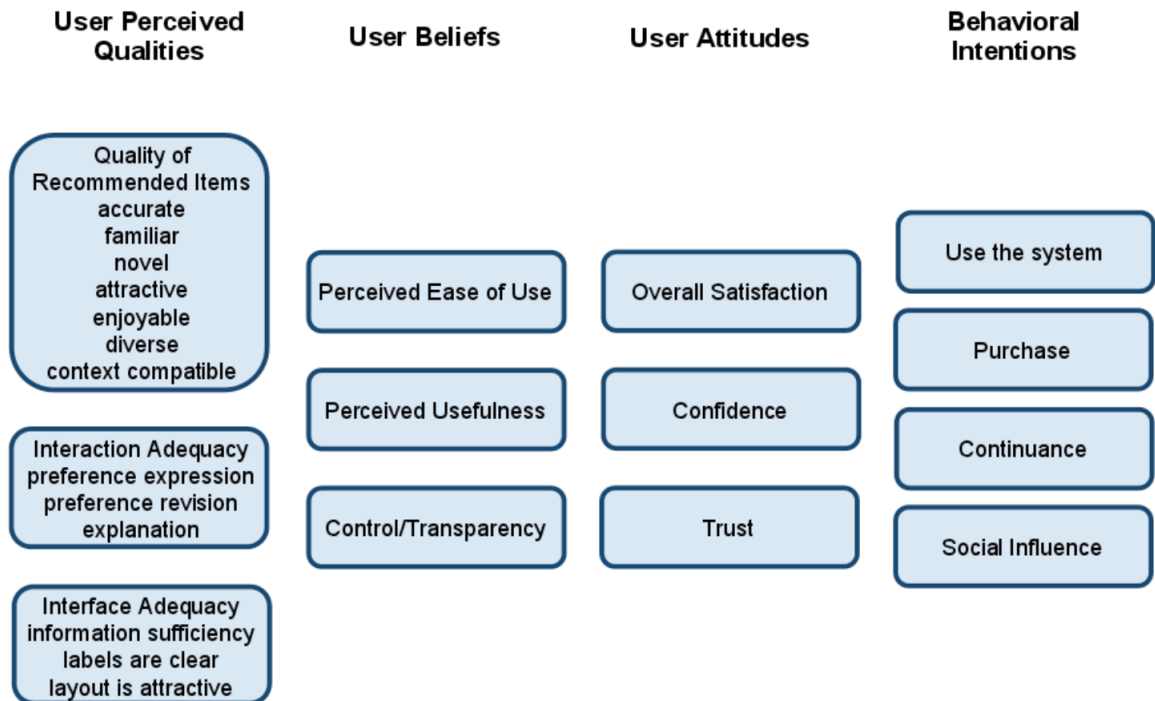
o algoritmo de recomendação irá buscar prever o interesse do usuário. As métricas tradicionalmente utilizadas para medir a eficiência do algoritmo de recomendação nesse tipo de avaliação são: (1) *Mean Absolute Error* (MAE) e (2) *Root Mean Square Error* (RMSE), utilizadas para calcular o quão próximas as notas previstas estão das notas reais; (3) *Precision*, definida pela divisão do número de itens relevantes recomendados pelo número total de itens recomendados; (4) *Recall*, definida pela divisão do número de itens relevantes recomendados pelo número total de itens relevantes existentes; e (5) *F-measure*, definida pela média harmônica entre Precisão e Recall.;

- Estudos com os usuários, no qual um pequeno grupo de usuários é convidado a participar de um experimento e realiza tarefas específicas relacionadas ao SR. As métricas mais comuns de SR podem ser utilizados em conjunto com medidas qualitativas para mensurar a satisfação dos usuários, por exemplo através da observação, questionários, entrevistas, etc.;
- Uso real do sistema, na qual o SR é avaliado em situações reais de uso, com uma quantidade grande de usuários, e os dados para a avaliação (quantitativa) são capturados de forma automática, por exemplo ferramentas de *Web Analytics*, registros de Logs, notas dadas pelos usuários, etc.

Pu, Chen e Hu (2011) propõe um framework para a avaliação de SRs utilizando **Estudos com os usuários**, com o objetivo de realizar uma avaliação do SR através da perspectiva do usuário. Esse *framework* se chama *Recommender Systems' Quality of User Experience* (ResQue) e foi proposto com base em outras ferramentas para avaliação centrada no usuário não exclusivas de SR: *Technology Acceptance Model* (TAM), que define três construtos (Facilidade de Uso Percebida, Utilidade Percebida e Intenções do Usuário em Utilizar o Sistema) e *Software Usability Measurement Inventory* (SUMI), que consiste de cinco construtos (Eficiência, Influência, Ajuda, Controle, Capacidade de Aprendizado) e um questionário de 50 questões.

O *framework* proposto por Pu, Chen e Hu (2011) consiste em quatro construtos: (1) Qualidades Percebidas pelos Usuários, (2) Crenças/Opiniões do Usuário, (3) Atitudes/Propósitos do Usuário, (4) Intenções Comportamentais. Para cada um dos construtos, vários aspectos são avaliados, como pode ser visto na Figura 1. Os autores definem ainda um conjunto de 60 questões para aplicar nessa avaliação como pode ser visto no Anexo A. Nesse questionário as questões são afirmações nas quais o usuário deve se posicionar em uma escala de Likert de 5 pontos, de "Discordo totalmente" até "Concordo totalmente". Os autores ainda afirmam que o conjunto de questões aplicado pode ser reduzido para um subconjunto com 15 questões (questões com asterisco no Anexo A).

Figura 1 – Construtos do framework de avaliação de SRs pela perspectiva do usuário



Fonte: Pu, Chen e Hu (2011)

2.6 CONSIDERAÇÕES SOBRE O CAPÍTULO

Nesse capítulo foram apresentados os principais conceitos relacionados a Sistemas de Recomendação (SRs) Sensíveis ao Tempo. Foram apresentadas desde as abordagens tradicionais, passando pelos SR Sensíveis Contexto e os seus paradigmas até as categorias de SRs Sensíveis ao Tempo definidas em Borba, Gasparini e Lichtnow (2017).

Dentre as categorias de SRs Sensíveis ao Tempo apresentadas na Seção 2.3, a utilizada por esse trabalho é o *Decay*. Nessa categoria é considerado que o interesse do usuário por um item acessado diminui com o passar do tempo. Neste sentido, considerando que o acesso a um item é um indicativo do interesse do usuário, os itens acessados mais recentemente tem um peso maior na identificação dos interesses do usuário, i.e., na definição do perfil do usuário.

Além disso, foram apresentadas as formas de avaliação de um SR como definido por Shani e Gunawardana (2011). Neste trabalho será utilizada uma avaliação pela perspectiva do usuário, como orientado por Pu, Chen e Hu (2012). Por isso, é apresentado o *framework* definido por Pu, Chen e Hu (2011) que busca avaliar a experiência do usuário através de um questionário com 60 questões divididas nos qua-

tro construtos do *framework*. Para ser utilizado, será necessário selecionar quais das questões serão utilizadas, pois como citado por Pu, Chen e Hu (2011) nem todas as questões se aplicam a todos os SRs, e traduzir essas questões para tornar possível a aplicação.

3 TRABALHOS RELACIONADOS

Todos os trabalhos descritos neste capítulo foram selecionados dentre os artigos analisados no mapeamento sistemático da literatura (BORBA; GASPARINI; LICHTNOW, 2017) sendo considerados aqueles estão enquadrados dentro da categoria de *Decay* no que se refere ao uso do tempo no algoritmo de recomendação, independente do domínio de aplicação. No total sete trabalhos foram selecionados, sendo apenas um da área educacional.

3.1 FAN ET AL. 2015

O trabalho de Fan et al. (2015) realiza a recomendação de *web services*, considerando a avaliação do serviço através da medição do *Quality of Service* (QoS). QoS considera características do serviço como tempo de resposta, disponibilidade, taxa de serviço, etc. Os autores consideram que a capacidade de prever a qualidade de um serviço diminui conforme o tempo que passou da última invocação desse serviço, devido a possíveis encerramentos do serviço, falhas na rede, etc. Por isso, a recomendação de serviços dos autores combina técnicas de similaridade com uma função de decaimento que considera que a QoS diminui com o passar do tempo. O modelo de decaimento proposto considera que as invocações mais recentes de dois usuários a um serviço devem ter um impacto maior no cálculo da similaridade entre os usuários. A função de decaimento de um item k é definida na Equação 3.1.

$$f(t_{i,k}, t_{j,k}) = e^{-\alpha|t_{atual} - \Delta t|} \quad (3.1)$$

Onde: α é o fator de decaimento; e Δt é a variação de tempo combinando o acesso do usuários u_i e u_j e é calculada com a fórmula apresentada na Equação 3.2.

$$\Delta t = \frac{(\Delta t_i + \Delta t_j)}{2} \quad (3.2)$$

Onde Δt_i é o intervalo de tempo entre a invocação do serviço pelo usuário u_i ao serviço e o tempo atual e Δt_j é o intervalo de tempo entre a invocação do mesmo serviço pelo usuário u_j e o tempo atual. Assim, a similaridade dos usuários diminui quanto maior for o Δt .

A nota do serviço k é considerada como a combinação do QoS com a função

de decaimento e é calculada através da Equação 3.3.

$$r_{u_i, s_k, t} = r_{u_i, s_k} \cdot f(t_{i,k}, t_{j,k}) \quad (3.3)$$

Os autores utilizam a nota resultado da função apresentada anteriormente para calcular a similaridade dos itens utilizando o coeficiente de Pearson. A Equação 3.4 apresenta o coeficiente de Pearson utilizado pelos autores que considera a função de decaimento.

$$sim(u_i, u_j, t) = \frac{\sum_{s_k \in w_{u_i, u_j}} (r_{u_i, s_k, t} - \bar{r}_{u_i})(r_{u_j, s_k, t} - \bar{r}_{u_j})}{\sum_{s_k \in w_{u_i, u_j}} (r_{u_i, s_k, t} - \bar{r}_{u_i})^2 \sum_{s_k \in w_{u_i, u_j}} (r_{u_j, s_k, t} - \bar{r}_{u_j})^2} \quad (3.4)$$

Onde w_{u_i, u_j} é um conjunto dos itens em comum invocados pelos usuários u_i e u_j . Utilizando essa fórmula de similaridade é possível calcular a similaridade entre os usuários e encontrar os que são mais similares.

A proposta dos autores considera também a localização desses usuários para calcular a similaridade. Quanto mais próximos eles estão, mais similares eles são considerados.

Foram realizados experimentos com a base de dados WS-Dream comparando o algoritmo proposto com outros 6 algoritmos: Recomendação considerando todos os usuários (RBA), Filtragem Colaborativa Usuário-Usuário utilizando a Correlação de Pearson (UPCC), Filtragem Colaborativa Item-Item utilizando a Correlação de Pearson (IPCC), *Context-Aware Service Recommender* (CASR), Método que considera a preferência do usuário de acordo com a sua localização (CASR-UP) e UPCC que considera o decaimento (ITRP-WS). A avaliação foi realizada utilizando uma estratégia *offline* e as métricas utilizadas para a comparação foram MAE e RMSE.

Primeiramente foi realizado um experimento verificando a influência dos parâmetros dos algoritmos. Esses parâmetros são considerados para decidir quais são os itens considerados significantes, ou seja, depois de calculada nota prevista para os itens, decidir qual o limite mínimo para que um item seja recomendado. O segundo experimento avaliou a influência da proporção da base de treino e de teste no experimento. E por último, para os algoritmos que utilizavam a Filtragem Colaborativa foi avaliado a influência da quantidade de k-vizinhos considerada na eficiência do algoritmo. Os resultados desses experimentos mostraram que o algoritmo proposto foi melhor que os outros 6 algoritmos analisados.

3.2 LUO ET AL. 2010

O trabalho de Luo et al. (2010) propõe um modelo de recomendação sensível ao contexto para ambientes de aprendizagem pervasivos. Esse modelo utiliza uma abordagem híbrida (baseada em conteúdo com filtragem colaborativa) com uma personalização pelo contexto. Três tipos de contexto são definidos:

1. Contexto do aluno, que possui as seguintes dimensões: tipo de dispositivo, ambiente (localização), perfil (informações pessoais como nome e afiliação), preferências (recursos pelo qual o aluno tem interesse), processo de aprendizagem (histórico de materiais acessados), pedido de acesso (a recursos educacionais massivos).
2. Contexto do serviço, que possui as seguintes dimensões: ambiente (localização), perfil (nome, parâmetros, retornos), QoS (parâmetros de qualidade do serviço, como carga de trabalho, reputação, disponibilidade, segurança, etc.).
3. Contexto do recurso, que segue o *China ELearning Technology Standard* que define as dimensões de um recurso educacional. Esse padrão define as dimensões como sendo: perfil (informações sobre o recurso como Título, Assunto, Palavras-chave), criador (nome, organização), audiência (tipo de educação, nível de ensino).

O modelo de recomendação proposto pode ser dividido em dois passos: *Logic-Based Resource Relevant Degree* e *Situation-Based Recourse Relevant Degree*.

Na etapa do *Logic-based Resource Relevant Degree* é feita uma análise do histórico de recursos acessados pelo aluno e as suas preferências. Esse passo combina a abordagem baseada em conteúdo, filtragem colaborativa e os padrões sequenciais de acesso.

A abordagem baseada em conteúdo considera as múltiplas dimensões dos recursos como assunto, assunto secundário, nível de ensino, etc. Nessa abordagem, é inserido um conceito de *Preference Energy* (PE) para refletir a variação do interesse do usuário com o passar do tempo. O PE indica que o interesse de um usuário a um item acessado diminui com o passar do tempo. Os autores definem a diminuição da PE pela fórmula apresentada na Equação 3.5.

$$PE_{attenuation}(x) = e^{-\lambda(x-1)}, \text{ com } x \geq 1 \quad (3.5)$$

Onde x é a ordem do recurso na lista de acessos do usuário e λ é o parâmetro de decaimento. Esse valor do PE, combinado com as avaliações feitas pelos usuários

para os itens, são utilizadas para gerar uma *Individual Preference Tree* que auxilia o cálculo da similaridade dos recursos candidatos a serem recomendados.

A *Individual Preference Tree* utilizada pela abordagem baseada em conteúdo também é considerada pelo algoritmo de filtragem colaborativa definida pelos autores para encontrar os k-vizinhos mais similares. Dessa forma, não só usuários que acessaram os mesmos itens podem ser considerados vizinhos (como na filtragem colaborativa tradicional), mas também usuários que acessaram itens similares entre si (mesmo assunto, palavras-chave, etc.) e os avaliaram de forma similar.

O último método de recomendação utilizado pela etapa chamada *Logic-based Resource Relevant Degree* utiliza os padrões sequencias de acesso dos usuários aos recursos. O algoritmo utilizado para a mineração dos padrões sequencias é o PrefixSpan, que procura sequências (ou subsequências) que apareceram em pelo menos μ acessos. Baseado na árvore de padrões sequenciais resultantes do algoritmo de mineração, é calculado quais os itens mais prováveis de serem acessados de acordo a sequência atual do usuário. A proposta dos autores define que o algoritmo de mineração de sequências deve ser executado de forma *offline*, para garantir a resposta em um tempo hábil.

A etapa de *Logic-based Resource Relevant Degree* combina o conjunto de recursos recomendados dos três algoritmos descritos, removendo da lista os recursos já acessados pelo usuário.

Na etapa de *Situation-based Resource Relevant Degree* é considerado que mesmo um recurso que combine com as preferências do usuário pode não ser adequados para a recomendação se o contexto do usuário (dispositivo, ambiente) não for adequado para utilizar o recurso. Para isso, no contexto dos recursos é descrito quais os dispositivos no qual a utilização do recurso é adequada e no contexto do usuário é descrito qual o dispositivo do usuário. Também é considerado o grau de satisfação no tempo para acessar um determinado recurso. Isso pode ser calculado pelo tamanho do recurso e a velocidade de internet do usuário. Combinando essas duas características é possível ter uma recomendação mais adequada a situação atual do aluno.

O algoritmo de recomendação proposto pelos autores então calcula uma lista de recursos candidatos a recomendação utilizando os algoritmos de *Logic-Based Resource Relevant Degree* e remove dessa lista os recursos que não satisfaçam o dispositivo do usuário e a satisfação mínima com o tempo de resposta esperado.

Esse algoritmo foi avaliado através de uma avaliação *offline* utilizando o base de dados do Movielens, onde foram adicionados dados de contexto às interações existentes na base. As métricas utilizadas para avaliar o algoritmo foram *Precision*, Utilidade e Validade (razão entre a quantidade de itens apropriados para o tipo de

conexão do usuário e o total de itens existentes). A avaliação foi feita em relação aos seguintes algoritmos tradicionais de recomendação: Algoritmo Baseado em Conteúdo utilizando o modelo de espaço vetorial, Filtragem Colaborativa combinando a abordagem Usuário-Usuário com a Item-Item, Abordagem Híbrida. Em comparação aos algoritmos tradicionais o algoritmo proposto teve melhores resultados no experimento realizado.

3.3 BENČIČ E BIELIKOVÁ 2012

O sistema de recomendação proposto por Bencic e Bielikova (2012) busca recomendar ações aos usuários no momento que for propício, de acordo com o contexto do usuário, e não apenas quando uma ação do interesse do usuário é encontrada. Uma ação se refere a qualquer coisa que seja utilizada pelo usuário final de uma aplicação.

O método proposto para a recomendação representa o contexto do usuário através de símbolos, onde cada símbolo é composto de duas partes – onde uma representa a dimensão e a outra representa a situação particular. Por exemplo, *Clima : Limpo*. Para cada símbolo do contexto do usuário é atribuído um valor no intervalo $(0, 1)$ que indica a convicção de que o usuário está naquele contexto.

A convicção de que o usuário está em determinado contexto é observada de tempos em tempos. Esse intervalo depende da velocidade de conexão do dispositivo do usuário, nível da bateria, etc. A convicção do usuário estar em determinado contexto diminui com o passar o tempo (supondo que uma nova observação demore a acontecer). Por isso, os autores utilizam uma função de decaimento para essa convicção conforme a Equação 3.6.

$$CF_t = \frac{CF_b}{(1 + r)^t} \quad (3.6)$$

Onde CF_t é a convicção calculada em função do tempo t , CF_b é a convicção base, r é o fator de decaimento e t é o tempo em horas passado desde a última observação.

As ações são modeladas através de um conjunto de regras. As regras são definidas automaticamente através do feedback do usuário e são representadas pelos antecedentes (em que situação a regra se aplica) e a consequência (a ação associada aquela situação). As regras também possuem um decaimento na convicção com o passar o tempo. Porém, nesse caso o decaimento não é constante, como acontece para o decaimento da convicção do contexto do usuário. Para as regras, o fator de decaimento é calculado para que a convicção não chegue a zero com muita ao acontecer

um longo período sem observações.

Combinando as convicções nas regras criadas com as convicções no contexto do usuário, são encontradas as ações com maior probabilidade de serem adequadas. O modelo de recomendação considera não apenas a última observação, mas sim uma combinação das últimas observações e suas respectivas convicções (com o fator de decaimento aplicado).

A avaliação do sistema foi feita realizando simulações de possíveis interações de um usuário imaginário em um ambiente de recomendação de notícias durante o período de um mês. Nessa simulação, o objetivo foi calcular as recomendações de notícias para três perfis de usuários: um usuário que lê notícias todos os dias pela manhã; outra que lê notícias apenas nas segundas pela manhã e sextas a noite; e outro que começa lendo as notícias apenas nas segundas pela manhã e muda o seu comportamento com o passar do tempo para a leitura as sextas a noite. Os resultados mostraram que o método conseguiu compreender o comportamento dos três tipos de usuário com um *Precision* e um *Recall* de quase 100%.

3.4 HAWALAH E FASLI 2014

O trabalho de Hawalah e Fasli (2014) propõe um método de recomendação utilizando o contexto do usuário representado através de ontologias. O algoritmo proposto pode se adequar a diversos domínios, de forma que o contexto seja incorporado aos interesses do usuário independente do que são os itens que serão recomendados. Além disso, o método considera não só o contexto atual do usuário, mas também os contextos capturados anteriormente. Os autores separam o método em três fases: Extração da informação, Aprender o perfil do usuário e Personalização.

A etapa de Extração da informação é realizada por um agente de captura dos dados que é genérico o suficiente para ser adaptado de acordo com o domínio. Em determinados domínios pode ser utilizado uma coleta perguntando explicitamente os interesses e o contexto ao usuário, enquanto em outros domínios é mais adequado capturar de forma implícita pela navegação do usuário.

A informação bruta capturada (seja de forma explícita ou implícita) é processada pelo agente extrator, responsável por extrair informação de mais alto nível. Esse agente está associado a dois tipos de bases de conhecimento: ontologias e taxonomias. O agente realiza um mapeamento dos itens que o usuário demonstrou interesse em conceitos da ontologia de referência, enquanto também extrai dimensões do contexto de mais alto nível utilizando-se das taxonomias de contexto.

A segunda etapa, responsável por compreender o perfil do usuário, utiliza a abordagem de Pré-filtragem Contextual para definir qual a parte do perfil do usuário

é relevante. É utilizada um método de micro-perfis, onde as informações do perfil do usuário (itens acessados, notas dadas) que aconteceram em contextos similares ao atual são consideradas mais relevantes para a recomendação. Para isso, é calculado a importância dos conceitos em cada contexto possível, de acordo com as informações do perfil do usuário. Nesse cálculo, é considerada a frequência com que o conceito aparece naquele determinado contexto, bem como a frequência com que esse conceito aparece em outros contextos e a frequência com que outros conceitos aparecem nesse contexto.

Ainda no cálculo da importância do conceito em determinado contexto, é considerado que os interesses do usuário podem mudar com o tempo. Para isso, é incluído na fórmula o fator de Recência (do inglês *Recency*), de forma que os interesses demonstrados pelo usuário mais recentemente são considerados mais importantes. A fórmula apresentada na Equação 3.7 apresenta o cálculo da recência.

$$Recency(c_j, ce_l) = \frac{1}{(1 + \log(d_t - d_l) \times \alpha)} \quad (3.7)$$

Onde d_t é a data de inicialização do cálculo, d_l é a data da última ocorrência do conceito c_j no contexto ce_l e α é o fator de decaimento.

Dessa forma, como resultado dessa etapa temos os interesses do usuário em cada contexto e o peso de cada um. Essas informações são utilizadas para construir ontologias de perfil contextual personalizada (CPOP, do inglês, *Contextual Personalized Ontology Profile*), sendo uma CPOP para cada contexto.

Na terceira etapa de personalização, a ontologia gerada na etapa anterior é utilizada para inferir outros conceitos que o usuário pode ter interesse, além dos já presentes do seu perfil. Isso é realizado utilizando a técnica de *Spreading Activation*, que explora as ontologias buscando encontrar os novos conceitos relacionados ao perfil do usuário. Utilizando essa técnica, é gerada uma lista de recomendações para cada CPOP, que são combinadas para gerar a lista final de recomendações para o usuário.

A avaliação do trabalho de Hawalah e Fasli (2014) foi através de um Estudo com Usuários, visando uma avaliação centrada no usuário como descrito por Kelly et al. (2009). O objetivo da avaliação é verificar se a recomendação contextual proposta no trabalho fornece uma recomendação mais eficiente do que os métodos tradicionais. 24 usuários participaram da avaliação, onde eles utilizaram um sistema por 30 dias, numa estratégia *Between-subjects*, i.e., cada grupo testa apenas uma versão do sistema. No total, 4 versões do sistema foram testadas: o algoritmo proposto (CAPS), o algoritmo proposto sem o uso do contexto (CAPS-C), um método de recomendação personalizado tradicional chamado pelos autores de Simple-P e um método de

recomendação não personalizado chamado de Non-P.

O resultado da avaliação analisou a nota dada pelos usuários para os itens em uma escala de likert de 1 a 4. Sendo itens com grau 1 e 2 considerados uma recomendação ruim e os itens com grau 3 e 4 considerados uma boa recomendação. Com isso foi possível calcular a *Precision at N* (P@N). O algoritmo proposto possui o melhor resultado de P@N entre os algoritmos testados.

3.5 QIAO E ZHANG 2012

O trabalho de Qiao e Zhang (2015) propõe um algoritmo de recomendação que considera as informações contextuais disponíveis em dispositivos móveis, como tempo, localização, tipo do dispositivo, etc. O algoritmo de recomendação é genérico, ou seja, sem um domínio de aplicação definido.

O objetivo dos autores é combinar a Filtragem Colaborativa com o contexto do usuário, considerando a variação temporal nos interesses do usuário. Para tal, são encontrados os k usuários com os interesses similares ao usuário atual, considerando o contexto do usuário, através de técnicas de clusterização. Após encontrados os k vizinhos, utiliza-se uma fórmula para a predição das notas para itens ainda não acessados pelo usuário com a incorporação de uma função temporal, como apresentado na Equação 3.8. Os itens com as notas previstas mais altas são recomendados ao usuário.

$$p_{u,i} = \bar{r}_u + \frac{\sum_{v \in U} \text{sim}(u,v)(r_{u,i} - \bar{r}_v)f(t_{ni})}{\sum_{v \in U} \text{sim}(u,v)f(t_{ni})} \quad (3.8)$$

Sendo $p_{u,i}$ a nota prevista do usuário u para o item i , \bar{r}_u a média das notas dadas pelo usuário u , $\text{sim}(u,v)$ a similaridade entre o usuário u e o seu vizinho v , $r_{u,i}$ o grau de interesse do usuário u pelo item i , t é o tempo passado desde que o usuário u utilizou o item i . A função f é a função exponencial de decaimento que representa a diminuição do interesse do usuário por um determinado item. A função de decaimento é definida na Equação 3.9.

$$f(t_{ni}) = e^{-t_{ni}} \quad (3.9)$$

A avaliação desse algoritmo foi realizada utilizando a base do MovieLens – 100 K, através da métrica MAE. Quando comparado a Filtragem Colaborativa tradicional o algoritmo proposto alcançou melhores resultados, ou seja, teve uma taxa de erro menor.

3.6 KUSHWAHA ET AL. 2016

O trabalho de Kushwaha et al. (2016) propõe uma versão modificada da técnica de *Joint Matrix Factorization* para uso em um sistema de recomendação de músicas incorporado ao Last.fm. A proposta busca reduzir a esparsidade dos dados e melhorar a qualidade das recomendações. Para isso, é incorporado informações como descrição dos itens, perfil do usuário e o seu contexto na matriz das notas dadas pelos Usuários para os Itens comumente utilizada na Filtragem Colaborativa. Assim, a matriz utilizada pelo algoritmo possui uma maior dimensionalidade e complexidade. A técnica de fatoração de matrizes é essencial para reduzir a dimensão da matriz utilizada e permitir a extração de informações latentes importantes para a recomendação.

Além disso, no algoritmo proposto por Kushwaha et al. (2016) é considerado uma estratégia de *tag-based user similarity matrix*, i.e., uma variação da Filtragem Colaborativa que utiliza as tags colocadas pelos usuários nos Artistas e Músicas como forma de comparar os usuários e encontrar os mais similares. Nessa estratégia é considerada a variação temporal da informação. Os autores consideram o decaimento da importância das *tags* colocadas pelo usuário nos itens com o passar do tempo. A fórmula utilizada pelos autores para representar o decaimento foi baseada em Iofciu e Demartini (2009), que pode ser vista na Equação 3.10.

$$postScore_i = \lambda^{\Delta Time_i} \quad (3.10)$$

Onde $postScore_i$ é a importância temporal da *tag* i , λ é o fator de decaimento, que deve ser menor do que 1 e foi utilizado por Kushwaha et al. (2016) como 0.9 e $\Delta Time_i$ é o tempo passado desde a interação. Além disso, é considerada a especificidade da *tag* para a nota final da *tag*. A fórmula da especificidade é definida na Equação 3.11

$$tagSpecificity_i = \log(50 + tagCount_i) \quad (3.11)$$

Onde $tagSpecificity_i$ é o fator de especificidade da *tag* e $tagCount_i$ representa quantas vezes a *tag* foi adicionada ao item i . A nota final da *tag* é calculada conforme a Equação 3.12.

$$tagScore_i = \frac{\sum_1^n postScore}{tagSpecificity_i} \quad (3.12)$$

A fórmula acima combina a importância temporal da *tag* com a sua especificidade. A nota final da *tag* é incorporada na matriz latente resultado da fatoração de

matrizes para servir como fator de decaimento para a importância das *tags*, representando a variação nos interesses do usuário.

O algoritmo proposto pelos autores foi avaliado utilizando uma base de dados do próprio Last.fm, combinado com a base de dados do DBpedia para a captura de informações sobre os artistas, compositores e músicas. A avaliação considerou a métrica RMSE para as previsões para as notas que o usuário daria para um determinado item comparada com a nota real. Quando comparado com outros dois algoritmos de recomendação que incorporam informação social para recomendação (chamados BPMFSR e Sorec), o algoritmo proposto por Kushwaha et al. (2016) se saiu melhor em 3 das 6 condições de experimento realizadas.

3.7 WEI, KHOURY E FONG 2013

Wei, Khoury e Fong (2013) descrevem uma proposta de recomendação que utiliza a Filtragem Colaborativa e que aplica um decaimento temporal na importância das interações dos usuários. No trabalho, é proposto um serviço para a recomendação de propagandas em redes sociais, que leva em consideração a confiança entre usuários, a reputação dos usuários e as relações entre usuários. Os autores afirmam que usuários com uma recomendação gerada com base em usuários conhecidos pelo usuário atual serão mais bem aceitas, assim como usuários no qual ele confia e nos usuários com alta reputação (especialistas).

O algoritmo de recomendação considera que a importância da relação entre os usuários diminui gradativamente com o tempo. Então, um comentário realizado hoje deve ter um peso maior que um comentário realizado há um mês atrás quando for avaliada a relação entre dois usuários. O decaimento é incluído diretamente na fórmula utilizada para realizar a comparação de similaridade entre dois itens, como pode ser visto a seguir na Equação 3.13.

$$s_{i,j}(t) = \frac{\sum_{u \in U_i^t \cap U_j^t} (f_{ui}^\alpha(t) \cdot r_{ui})(f_{uj}^\alpha(t) \cdot r_{uj})}{\sqrt{\sum_{u \in U_i^t} (f_{ui}^\alpha(t) \cdot r_{ui})^2 \sum_{u \in U_j^t} (f_{uj}^\alpha(t) \cdot r_{uj})^2}} \quad (3.13)$$

Onde a função f é definida pelos autores como relevância temporal do item e pode ser vista na Equação 3.14

$$f_{uj}^\alpha(t) = e^{-\alpha(t-t_{ui})} \quad (3.14)$$

Onde fator α é responsável por controlar a taxa de decaimento, t representa o tempo atual e t_{ui} representa o tempo no qual o usuário u utilizou o item i .

Além do decaimento na relevância das relações entre os usuários, os autores também incluem a confiança entre os usuários e a reputação de especialistas da área na fórmula de similaridade visando melhorar a qualidade das recomendações.

O algoritmo proposto foi avaliado utilizando três bases de dados: MovieLens, Facebook e Delicious. O algoritmo foi comparado a outros dois algoritmos: Filtragem Colaborativa usando correlação de Pearson e usando correlação de Pearson com efeito temporal. As métricas utilizadas foram *Mean Absolute Error* e *Root Mean Square Deviation*. Os resultados mostraram que o algoritmo proposto melhorou significativamente a qualidade das recomendações.

3.8 CONSIDERAÇÕES SOBRE O CAPÍTULO

Nesse capítulo foram apresentados sete trabalhos relacionados que utilizam Sistemas de Recomendação (SRs) com o uso do *Decay*. A tabela 1 apresenta um resumo dos trabalhos apresentados e uma comparação com a proposta desse trabalho.

Pode-se observar que dos trabalhos relacionados, apenas um é da área educacional, enquanto os outros são aplicados em outros domínios de aplicação. Além disso, nenhum dos trabalhos apresentados utiliza a abordagem Baseada em Conteúdo. A abordagem mais comumente utilizada nos trabalhos relacionados é a Filtragem Colaborativa, aparecendo em três trabalhos sozinhas e em mais dois outros trabalhos combinados com outras abordagens (Abordagem Híbrida).

Dos trabalhos relacionados, apenas um não considera uma Constante de decaimento. Essa Constante de decaimento geralmente é um valor entre 0 e 1 que define a velocidade com que o peso das interações do usuário diminui. Esse valor é definido pelos autores de forma empírica, não tendo um método para a definição dessa constante. No trabalho de Qiao e Zhang (2015), que não utiliza uma Constante de decaimento, é utilizada uma função exponencial para o decaimento do interesse do usuário. Na proposta do presente trabalho também não é utilizada uma Constante de decaimento e é utilizada uma função de decaimento linear.

Também é possível observar que apenas um trabalho não utiliza o Tempo absoluto na função de decaimento. Isso significa que a maioria dos trabalhos relacionados considera o tempo passado (e.g., em segundos, minutos, horas, etc.) desde a interação para calcular a importância desta. O trabalho de Luo et al. (2010) considera a sequência de itens acessados, dando uma maior importância para os itens mais recentes de acordo com a posição do item na lista ordenada pelo tempo de acesso. A proposta desse trabalho utiliza uma estratégia similar a de Luo et al. (2010) nesse aspecto.

Sobre a avaliação, pode-se observar que a maioria utiliza uma avaliação *Of-*

Tabela 1 – Comparação dos trabalhos relacionados.

Trabalho	Domínio	Abordagem	Constante de decaimento	Tempo absoluto	Avaliação	Algoritmos comparados
Fan et al. (2015)	Web Services	Filtragem Colaborativa	Sim	Sim	Offline, com a base WS-Dream	RBA, UPCC, IPCC, CASR, CASR-UP e ITRP-WS
Luo et al. (2010)	AVA	Híbrida	Sim	Não	Offline, com a base do Movielens	Abordagens tradicionais: Baseada em Conteúdo, Filtragem Colaborativa e Híbrida
Bencic e Bielikova (2012)	Genérico	Baseada em Conhecimento	Sim	Sim	Simulações	Nenhum
Hawalih e Fasli (2014)	Genérico	Baseada em Conhecimento	Sim	Sim	Estudos com usuários, com 24 usuários durante 30 dias	CAPS-C, Simple-P e Non-P
Qiao e Zhang (2015)	Genérico	Filtragem Colaborativa	Não	Sim	Offline, com a base do Movielens	Filtragem Colaborativa tradicional
Kushwaha et al. (2016)	Músicas	Híbrida	Sim	Sim	Offline, com a base da Last.fm	BPMFSR e Sorec
Wei, Khoury e Fong (2013)	Propagandas	Filtragem Colaborativa	Sim	Sim	Offline, com as bases do Movielens, Facebook e Delicious	Filtragem Colaborativa usando correlação de Pearson com e sem efeito temporal
Proposta	AVA	Baseada em Conteúdo	Não	Não	Estudos com usuários, com pelo menos 60 usuários durante 45 dias	Abordagem Baseada em Conteúdo tradicional

Fonte: O autor.

ffline. Bencic e Bielikova (2012) realiza uma simulação para demonstrar o algoritmo de recomendação, que segundo a definição de Shani e Gunawardana (2011) não se encaixaria como uma avaliação de SR. O trabalho de Luo et al. (2010), que é da área educacional assim como a proposta desse trabalho, utiliza uma avaliação *offline* com a base do Movielens ¹. Essa avaliação, apesar de ser realizada de forma correta e apresentar bons resultados, não garante que o SR desenvolvido por Luo et al. (2010) teria um bom resultado em um ambiente real de uso. Por isso, na proposta desse trabalho será realizada uma avaliação com participação de usuários em um ambiente educacional durante uma situação real de uso.

Na avaliação citada anteriormente, o objetivo será comparar o algoritmo proposto neste trabalho com a abordagem Baseada em Conteúdo tradicional. É possível observar pelos trabalhos relacionados que comparar a proposta com os algoritmos mais tradicionais da área é algo comumente realizado (FAN et al., 2015; LUO et al., 2010; HAWALAH; FASLI, 2014; QIAO; ZHANG, 2015; WEI; KHOURY; FONG, 2013).

¹ Uma base de dados com notas dadas por usuários à filmes. Disponível em: <https://grouplens.org/datasets/movielens/>.

4 SR SENSÍVEL AO TEMPO

Neste capítulo é apresentado o SR Sensível ao Tempo para ambientes educacionais proposto nesse trabalho. O algoritmo proposto considera o decaimento nos interesses do usuário com o passar do tempo (categoria do *Decay* - Seção 2.3.4). Além do algoritmo proposto, são apresentados cenários que ilustram o uso desse algoritmo e discussões sobre as suas vantagens.

4.1 ALGORITMO PROPOSTO

O algoritmo de recomendação proposto nesse trabalho combina a Abordagem Baseada em Conteúdo com o uso do contexto temporal através da categoria *Decay* (ver seção 2.3.4). Para o SR, o perfil do usuário é composto pelos itens acessados por ele. A cada item acessado pelo usuário, são armazenadas as palavra-chaves do item juntamente com o timestamp do acesso e a sequência desse item no histórico do usuário, começando em 1 para o primeiro item acessado. Os itens que serão recomendados também são representados no algoritmo de recomendação por um conjunto de palavras-chave. O cálculo da relevância de um determinado item i para um usuário u no algoritmo proposto está representado na Equação 4.1.

$$F(u, i) = S(u, i) \cdot R(I_{u,i}) + A(i) \quad (4.1)$$

Onde: F é a função que calcula a relevância de um item i para um usuário u ; S é a função de similaridade entre o perfil do usuário u (representado através de conjunto de palavras-chave dos itens acessados) e o item i (representado pelo conjunto de palavras-chave que o caracterizam); R é o maior valor de recência dos itens do conjunto $I_{u,i}$ (itens acessados pelo usuário u e com alguma similaridade com o item i); A é uma função que retorna 1 se o item i nunca foi acessado pelo usuário u e 0 se o item já foi acessado.

A função S de similaridade entre o usuário u e o item i pode ser calculada utilizando funções de similaridade como o Cosseno (ver Seção 2.1.1). A função de recência R , responsável pelo *Decay*, está definida na Equação 4.2.

$$R(I_{u,i}) = \max_{\{j \in I_{u,i}\}} \frac{x_j}{|I_u|} \quad (4.2)$$

Onde: x_j é a posição do item j na lista de itens acessados pelo usuário u ordenada de forma crescente pelo *timestamp* do acesso; e $|I_u|$ é a quantidade de itens

acessados pelo usuário u . Essa fórmula considera que mais de um item acessado pelo usuário pode ser similar ao item i , por isso a fórmula retorna a maior recência de todos os itens presentes no perfil do usuário que são similares a i . A seguir temos um exemplo do uso da fórmula da recência considerando apenas um item, que pode ser estendida para o uso com vários itens e aplicada a função max como descrito na fórmula acima.

Considerando um usuário u que acessou três itens diferentes nos seguintes *timestamps* (em *epoch*): Item A acessado em 1503670382; Item B acessado em 1500027182; Item C acessado em 1508051582. A posição desses itens no histórico do usuário u é: Item A - posição 2, Item B - posição 1 e Item C - posição 3. As equações 4.3, 4.4 e 4.5 apresentam o resultado do cálculo de recência para cada um dos itens:

$$R(A) = \max \frac{2}{3} = 0.\bar{6} \quad (4.3)$$

$$R(B) = \max \frac{1}{3} = 0.\bar{3} \quad (4.4)$$

$$R(C) = \max \frac{3}{3} = 1.0 \quad (4.5)$$

No Algoritmo 1 é possível observar o pseudocódigo do algoritmo proposto. Nesse algoritmo estão representadas em mais alto nível as etapas da recomendação. Primeiramente, são buscados no banco de dados os itens acessados pelo usuário e os itens candidatos a recomendação. Os itens candidatos a recomendação podem ser todos os itens disponíveis no ambiente ou um subconjunto de itens, de acordo com a necessidade da aplicação.

O cálculo da recência para cada item do perfil do usuário é feito também na fase de inicialização, pois esse cálculo só precisa ser realizado uma vez. Depois disso, para cada item candidato a ser recomendado para o usuário, é calculada a similaridade do perfil do usuário (composto pelas palavras-chave dos itens acessados por ele) com as palavras-chave do item candidato. Além disso, é buscada a lista de itens acessados pelo usuário que são similares ao item candidato. Nessa etapa, também é encontrado o Item Acessado pelo usuário mais recente que é similar ao item candidato, para utilizar o seu valor de recência no cálculo da Relevância.

O cálculo da Relevância para o item candidato é realizado utilizando a Equação 4.1. Os cinco itens com a maior Relevância são retornados pelo algoritmo para serem recomendados ao usuário.

No Algoritmo 2 é demonstrado o cálculo da recência para cada um dos itens do perfil do usuário, que é utilizado pelo Algoritmo 1 descrito anteriormente. Para esse

algoritmoRecomendacaoDecay (*Usuário*)**input** : Usuário que irá receber a recomendação.**output:** Lista com cinco itens recomendados para o usuário.

/* Inicialização

*/

itens = buscarItensCandidatos();

itensAcessados = buscarItensAcessados(Usuário);

recênciaItensAcessados = calcularRecência(itensAcessados);

relevâncias = {};

/* Computação das recomendações

*/

foreach *item* \in *itens* **do**

| similaridade = calcularSimilaridade(itensAcessados, item);

| itensSimilares = buscarItensSimilares(itensAcessados, item);

| recência = calcularMaxRecência(itensSimilares,

| recênciaItensAcessados);

| relevâncias[item] = similaridade * recência + itensAcessados.inclui?(item);

end**return** relevâncias.max(5);**Algoritmo 1:** Pseudocódigo do Algoritmo de Recomendação Proposto.

cálculo, o próprio registro de item acessado pelo usuário possui a informação da posição em que se encontra no histórico do usuário e é aplicada uma versão simplificada da Equação 4.2. A função *max* presente na Equação 4.2 não é aplicada nesse momento, sendo calculada apenas a recência para cada item individualmente. No método descrito no Algoritmo 3 é que a função *max* será aplicada para encontrar a maior recência entre os itens similares ao item candidato à recomendação. O algoritmo tem um processo que difere da Equação 4.2 justamente para que recência não seja recalculada várias vezes e assim melhore o seu tempo de processamento.

calcularRecência (*itensAcessados*)**input** : Lista de registros de acesso do usuário.**output:** Vetor Associativo onde as chaves são os itens acessados pelo usuário e os valores são as recências calculadas para esses itens.

recênciaItensAcessados = {};

quantidadeItensAcessados = itensAcessados.tamanho();

foreach *itemAcessado* \in *itensAcessados* **do**

| recênciaItensAcessados[item] = itemAcessado.posição() /

| quantidadeItensAcessados;

end**return** recênciaItensAcessados;**Algoritmo 2:** Pseudocódigo do cálculo da recência para cada item presente no perfil do usuário.

calcularMaxRecência (*itensSimilares, recênciaItensAcessados*)

input : Lista de itens similares ao item candidato a recomendação e vetor associativo com a recência calculada para cada item do perfil do usuário.

output: Valor máximo de recência entre itens similares.

$recênciaItensSimilares = recênciaItensAcessados.find(itensSimilares);$

return $recênciaItensSimilares.max(1);$

Algoritmo 3: Pseudocódigo do cálculo da recência máxima entre os itens do perfil do usuário similares ao item candidato.

4.2 CENÁRIO DE USO

Um cenário de uso do *Decay* é apresentado a seguir. *O aluno Pedro está matriculado em uma disciplina de Estrutura de Dados que possui quatro tópicos, sendo eles Pilhas, Filas, Listas e Árvores. Nessa disciplina, para cada um dos conteúdos é aplicada uma prova para avaliar os conhecimentos dos alunos. Para cada conteúdo da disciplina, existem 20 materiais que pode se acessado normalmente pelo aluno e mais 10 materiais relacionados que só podem ser acessados quando recomendados. Cada um desses itens é representado pela palavra-chave do conceito em que está relacionado. Até o momento da primeira prova sobre o conteúdo de Pilhas, Pedro acessou apenas materiais relacionados a Pilhas. O algoritmo de recomendação utilizando a abordagem Baseada em Conteúdo recomenda para Pedro apenas materiais relacionados a Pilhas. Após a primeira prova, Pedro começa a acessar materiais relacionados a Filas, o segundo tópico da disciplina. Em uma abordagem Baseada em Conteúdo tradicional as recomendações continuariam sendo sobre o conteúdo de Pilhas por um bom tempo, pois o perfil de Pedro seria em grande parte composto por materiais acessados sobre esse assunto. Porém, com o uso do **Decay**, no momento em que Pedro começar a acessar materiais sobre Filas o algoritmo de recomendação dará um peso maior para esses materiais (sem ignorar os itens acessados anteriormente) e em pouco tempo Pedro já estará recebendo recomendações sobre o novo tópico estudado. Da mesma forma, se Pedro voltar a acessar conteúdos anteriores para relembrar algum conceito, o SR também perceberá isso e recomendará itens relacionados ao primeiro conteúdo novamente.*

Agora será apresentado em mais detalhe qual seria o comportamento do algoritmo proposto nesse cenário de uso apresentado. Ao estudar o primeiro assunto da disciplina, Pedro acessa os conteúdos relacionados a Pilhas e no seu perfil o SR formará o seu perfil com vários itens representados pela palavras-chave "Pilha". O SR, ao gerar uma recomendação, irá encontrar um valor de similaridade alto entre o perfil do usuário e os itens candidatos a recomendação que também possuam a palavra-chave "Pilha". Como todos os itens do perfil do usuário nesse momento tem a palavra-chave "Pilha", a recência não terá um impacto importante nesse momento,

pois independente do item que for mais recente a recomendação será a mesma. Na Tabela 2 é possível observar de forma resumida o comportamento do algoritmo (estamos considerando que nenhum item recomendado tenha sido acessado).

Tabela 2 – Cenário de Uso: comportamento do algoritmo após o acesso de itens relacionados a pilhas.

Retornos	Itens relacionados a pilhas	Itens relacionados a filas	Itens relacionados a listas	Itens relacionados a árvores
Similaridade	1.0	0.0	0.0	0.0
Recência	1.0	0.0	0.0	0.0
Acessado ou não	1.0	1.0	1.0	1.0
Relevância	2.0	1.0	1.0	1.0

Fonte: O autor.

Dessa forma, como é possível ver na Tabela 2 os itens relacionados a pilhas terão uma maior Relevância e serão os itens recomendados. A partir do momento que o usuário comece a acessar itens relacionados a filas, o algoritmo passará a dar uma importância maior a itens relacionados a esse segundo tópico. Na Tabela 3 é possível observar o comportamento do algoritmo nesse caso.

Tabela 3 – Cenário de Uso: comportamento do algoritmo após o acesso de itens relacionados a filas.

Retornos	Itens relacionados a pilhas	Itens relacionados a filas	Itens relacionados a listas	Itens relacionados a árvores
Similaridade	0.7	0.3	0.0	0.0
Recência	0.35	1.0	0.0	0.0
Acessado ou não	1.0	1.0	1.0	1.0
Relevância	1.245	1.3	1.0	1.0

Fonte: O autor.

É possível observar na Tabela 3 que, mesmo o perfil do usuário sendo ainda mais similar a itens com a palavra-chave "Pilha", o impacto da Recência no algoritmo fez com a Relevância dos itens relacionados a filas seja maior e, dessa forma, esses itens sejam recomendados. Supondo que o aluno Pedro decida voltar a acessar itens relacionados a pilhas para recapitular os conteúdos. O comportamento do algoritmo nesse caso é apresentado na Tabela 4.

Como pode ser observado na Tabela 4, algoritmo percebe que o aluno voltou a um tópico já estudado anteriormente e voltaria a recomendar itens relacionados a isso. Como o perfil do aluno é composto ainda por muitos itens relacionados ao tópico de Pilhas acessado primeiramente, ao voltar a acessar esses itens o algoritmo voltará a recomendar esses itens ao aluno.

Tabela 4 – Cenário de Uso: comportamento do algoritmo após o aluno voltar a acessar itens relacionados a pilhas.

Retornos	Itens relacionados a pilhas	Itens relacionados a filas	Itens relacionados a listas	Itens relacionados a árvores
Similaridade	0.75	0.25	0.0	0.0
Recência	1.0	0.7	0.0	0.0
Acessado ou não	1.0	1.0	1.0	1.0
Relevância	1.75	1.175	1.0	1.0

Fonte: O autor.

4.3 DISCUSSÃO SOBRE O ALGORITMO PROPOSTO

Um ponto negativo da Filtragem Colaborativa é a necessidade de uma comunidade de usuários ativa, que nem sempre é possível em um AVA onde as turmas muitas vezes são menores (entre 10 e 100 alunos). A Abordagem Baseada em Conteúdo é considerada para o SR proposto porque essa abordagem permite a recomendação em um sistema que não possui uma comunidade ativa e pode suprir as necessidades desse domínio.

Os SRs Sensíveis ao Tempo tem uma vantagem em relação à outros SRs Sensíveis ao Contexto por a informação temporal ser mais simples de capturar e manipular que outras informações contextuais, e.g., localização. Além disso, esses tipos de algoritmos estão sendo explorados em outros domínios de aplicação, como pode ser visto nos 88 artigos analisados no Mapeamento Sistemático realizado (BORBA; GASPARINI; LICHTNOW, 2017), e demonstraram bons resultados. Por isso, esse trabalho busca aplicar o contexto temporal no algoritmo de recomendação na área educacional (na qual foram encontrados apenas quatro trabalhos) e avaliar os resultados.

A escolha do *Decay* se justifica como forma de procurar minimizar uma das grandes desvantagens da abordagem Baseada em Conteúdo: a Superespecialização. Na abordagem Baseada em Conteúdo as recomendações seriam geradas levando em conta todos os itens acessados pelo usuário igualmente. Enquanto ao aplicar o *Decay*, os itens acessados mais recentemente possuem um grau de importância maior para o algoritmo do que itens acessados anteriormente. Dessa forma, mesmo que o usuário tenha acessado muitos materiais sobre determinado assunto, ao começar a acessar materiais sobre outro assunto o algoritmo de recomendação consegue rapidamente se adaptar e gerar recomendações sobre esse novo conteúdo.

O algoritmo proposto nesse trabalho considera o decaimento no peso dos itens do perfil do usuário em função da posição do item na sequência de materiais acessados, como mostrado na Seção 4.1, e não na quantidade de tempo passada em segundos como feito na maioria dos trabalhos relacionados mostrados no Capítulo

3. Isso é uma vantagem, pois no domínio educacional o passar do tempo não é tão relevante quanto em outros domínios, como na recomendação de *Web Services* por exemplo.

Os alunos em um AVA podem ter ritmos de estudo diferentes e, por isso, é assumido nesse trabalho que faz mais sentido analisar quantos itens foram acessados desde do acesso de determinado material do que o tempo passado desde a interação. Dessa forma, esse algoritmo não considera se o usuário acessou todos os itens do seu perfil em um único dia, ou se ele acessou metade do materiais na primeira semana do curso e a outra metade na última semana ou se ele acessou alguns materiais todos os dias durante o curso.

Além disso, no algoritmo proposto nesse trabalho não é definido um fator de decaimento único para os todos os usuários como nos trabalhos relacionados apresentados no Capítulo 3. Isto porque é necessário considerar que cada aluno tem um estilo de aprendizagem diferente e o decaimento para um aluno poderia ser diferente do decaimento para outros alunos. E não foram encontrados nos trabalhos relacionados que mostrem um modelo para calcular o fator de decaimento de forma personalizada para cada usuário. Em geral, os autores utilizam o fator de decaimento escolhido de forma empírica e aplicam o mesmo fator para todos os usuários.

4.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

Nesse capítulo é apresentada a proposta desse trabalho. O algoritmo proposto utiliza a abordagem Baseada em Conteúdo em conjunto com o uso do tempo através da categoria *Decay*, ou seja, é dado um peso menor na recomendação para os itens acessados a mais tempo pelo usuário e um peso maior para os itens mais recentes. No Sistema de Recomendação (SR) proposto é combinada a similaridade entre o perfil do usuário e os itens, a recência dos itens acessados pelo usuário e se os links recomendados já foram ou não acessados. O algoritmo proposto será avaliado utilizando o experimento descrito no Capítulo 5.

5 PLANEJAMENTO DO EXPERIMENTO

Neste capítulo é apresentado o experimento que será realizado para a avaliação da proposta apresentada no Capítulo 4. Para avaliar a proposta deste trabalho irá utilizar uma avaliação pela perspectiva do usuário, que segundo a definição de Shani e Gunawardana (2011) se encaixa em um Estudo com usuários. O algoritmo proposto será incorporado ao ambiente AdaptWeb[®] avaliado em uma situação real de uso em um minicurso de algoritmos. Esse capítulo explica em mais detalhes o ambiente AdaptWeb[®] as mudanças propostas para o ambiente e o experimento planejado.

5.1 DESCRIÇÃO DO AMBIENTE ADAPTWEB[®]

O AdaptWeb[®] (Ambiente de Ensino-Aprendizagem Adaptativo na Web) é um sistema open source que consiste em um AVA capaz de adaptar o conteúdo, a apresentação e a navegação em determinado curso às características e preferências do aluno (GASPARINI et al., 2009). A Subseção 5.1.1 apresenta a Estrutura Geral do AdaptWeb[®].

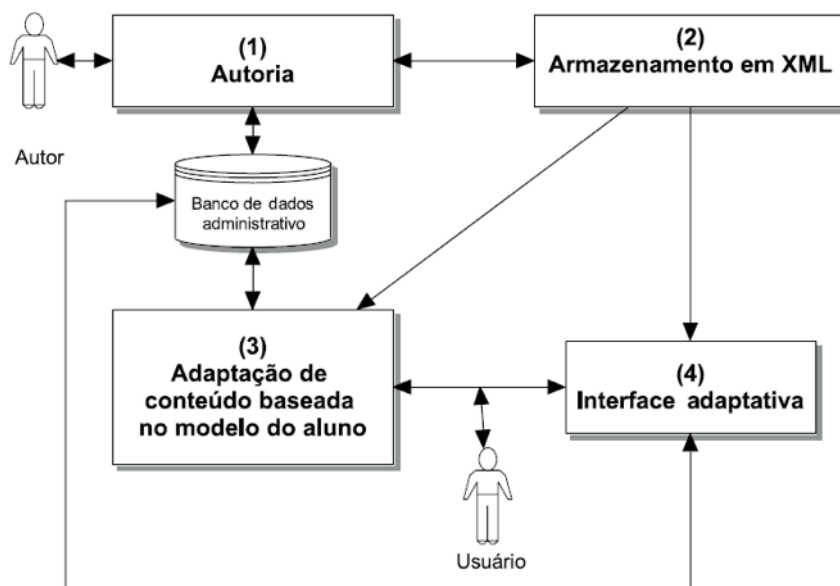
5.1.1 Estrutura do AdaptWeb[®]

A estrutura do AdaptWeb[®] é composta por quatro módulos: (1) o módulo de autoria; (2) o módulo de armazenamento em XML (Extensible Markup Language); (3) o módulo de adaptação do conteúdo baseado no modelo do usuário e (4) o módulo de interface adaptativa (GASPARINI, 2003), conforme pode ser visto na Figura 2.

O módulo de autoria (1) consiste na organização do conteúdo instrucional a ser disponibilizado para o aluno, sendo que este conteúdo pode ter arquivos classificados como conceito, exemplos, exercícios e materiais complementares (GASPARINI, 2003). Ao criar um conteúdo no sistema, o autor pode definir para quais cursos e disciplinas deseja que o conteúdo ou arquivo esteja disponível. Isto significa que um aluno de um Curso X e de outro Curso Y, matriculados em uma mesma disciplina, podem ter conteúdos distintos, conforme definido pelo professor. Por exemplo, a disciplina de Cálculo I pode ser oferecida para os cursos de Ciência da Computação e Engenharia Elétrica e sua abrangência e profundidade pode ser distinta para cada curso.

Em Borba (2015), foi proposto uma nova categoria para os conteúdos chamada Links de Apoio. Esses Links de Apoio são links externos ao ambiente AdaptWeb[®] que são cadastrados pelo professor como um material alternativo de estudo e não estão diretamente atrelados a nenhum conceito em específico. O objetivo foi criar uma nova

Figura 2 – Estrutura do AdaptWeb®



Fonte: GASPARINI (2003)

categoria de materiais que poderia ser recomendada para o usuário a qualquer momento de sua interação.

O módulo de armazenamento em XML (2) é responsável por organizar os conteúdos e arquivos disponibilizados pelo autor em um arquivo XML (GASPARINI, 2003). É utilizada a representação através de XML devido à sua alta flexibilidade, oferecendo a estruturação dos documentos de forma independente da apresentação.

O módulo de adaptação do conteúdo baseado no modelo do aluno (3) é responsável por adaptar o conteúdo da disciplina para cada curso. Por fim, o módulo de interface adaptativa (4) é responsável pela adaptação da navegação e da apresentação da interface do ambiente de acordo com o curso, preferências do modo de navegação (modo tutorial ou livre) e o conhecimento do usuário (GASPARINI, 2003).

5.1.2 Sistema de Recomendação no AdaptWeb®

Na Subseção 5.1.1 foi apresentada a estrutura do ambiente AdaptWeb®, que possui quatro categorias de materiais para cada conteúdo. Além disso, existe uma outra categoria chamada Links de Apoio com o propósito de ser um material auxiliar e que pode ser recomendado a qualquer momento para o usuário. Como no ambiente AdaptWeb® as palavras-chave para os itens já são cadastradas pelo professor da disciplina, não será utilizada nenhuma técnica para captura automática das palavras-chave.

Fazendo uma relação da estrutura do AdaptWeb® com o algoritmo proposto no

Capítulo 4, os itens de todas as cinco categorias de materiais serão consideradas para a composição do perfil do usuário. Todos os materiais acessados em cada uma das categorias é representado através das palavras-chave, e essas palavras-chave farão parte do perfil do aluno a partir do momento em que este acessar o material. Já para os itens recomendados, apenas os Links de Apoio serão considerados. O Sistema de Recomendação (SR) irá buscar, com base nos itens acessados pelo aluno, os Links de Apoio mais adequados para a recomendação e irá apresentar através de uma lista de itens. A forma de apresentação das Recomendações é discutida em mais detalhe na Subseção 5.1.3.

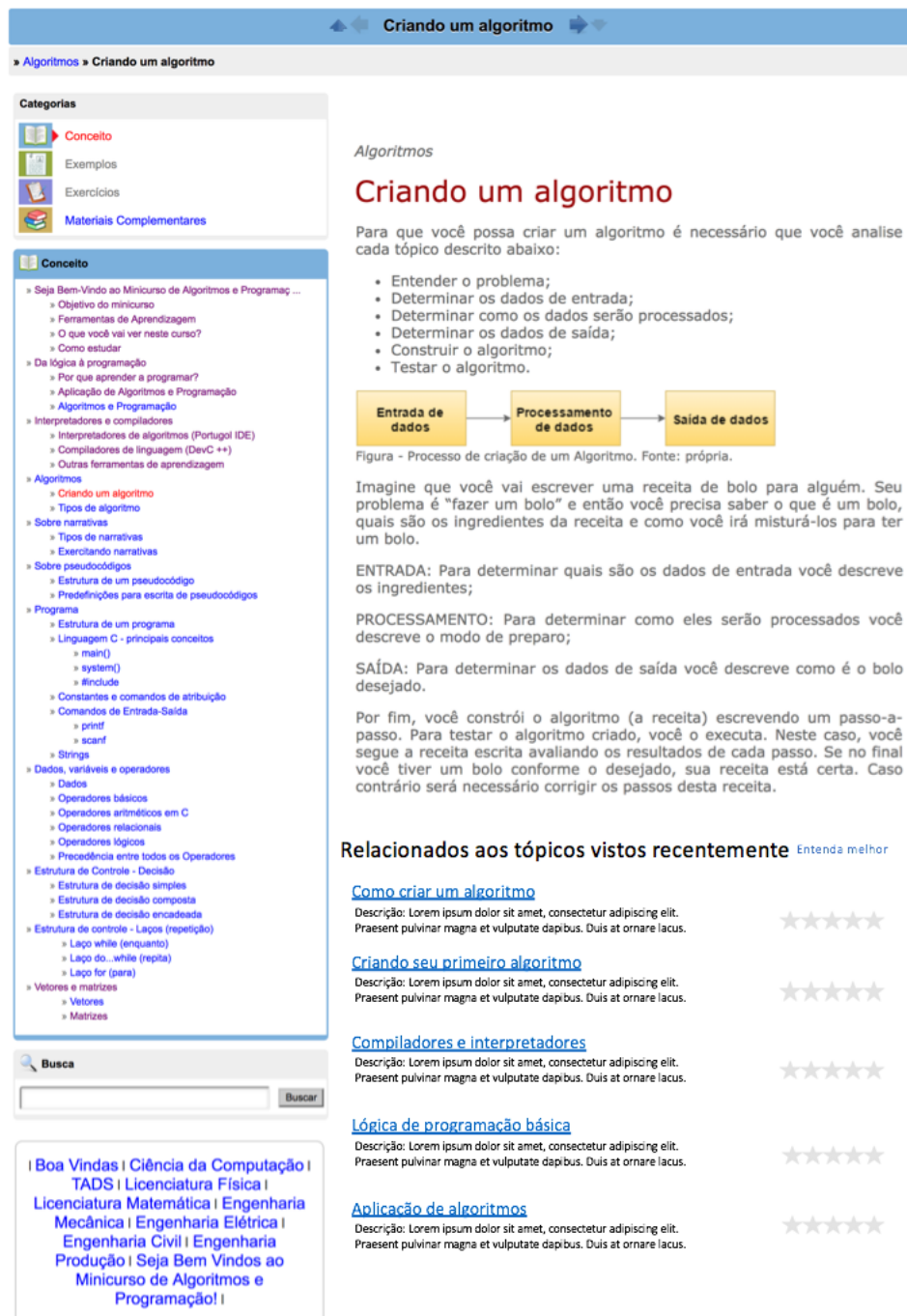
5.1.3 Apresentação das Recomendações

A lista de recomendações neste trabalho será apresentada ao aluno na tela principal do ambiente do aluno. Dessa forma, quando o SR possuir itens para recomendar para o usuário esses itens aparecem em uma lista logo abaixo do conteúdo que ele estiver visualizando no momento, independente se o aluno esteja na tela de Conceito, Exercícios, Exemplos ou Materiais Complementares. Na Figura 3 pode-se observar um protótipo da apresentação das recomendações. Essas recomendações podem ser apresentadas ao usuário no momento em que este estiver acessando quaisquer itens que sejam da categoria Conceito e Materiais Complementares, assim que o SR possuir itens relevantes para recomendar.

As principais informações dos Links de Apoio recomendados apresentados para o aluno são: Link, Nome do Link, Descrição e a possibilidade de avaliar o item com notas de 1 a 5. A nota dada pelo usuário não será considerada pelo algoritmo em um primeiro momento, mas como trabalho futuro pode vir a incorporar na recomendação, e.g., não tornar a recomendar itens que foram avaliados com notas baixas. Pu, Chen e Hu (2012) afirmam que enquanto recomendar um item apenas é pouco, recomendar mais do que cinco itens aumenta a dificuldade de escolher do usuário. Por isso, a quantidade máxima de itens recomendadas para o usuário em cada recomendação é de cinco itens.

Para cumprir o requisito de Explicação das recomendações citada por Pu, Chen e Hu (2012), será adicionado um botão de "Entenda melhor" que tem por objetivo explicar ao usuário como a lista de itens foi gerada. Ao entender o funcionamento do algoritmo de recomendação o usuário tem a possibilidade aprimorar o seu perfil para personalizar as recomendações recebidas. Na Figura 4 está um protótipo da explicação da recomendação mostrada para o aluno.

Figura 3 – Proposta de Interface de Recomendação



Criando um algoritmo

» Algoritmos » Criando um algoritmo

Categorias

- Conceito
- Exemplos
- Exercícios
- Materiais Complementares

Conceito

- » Seja Bem-Vindo ao Minicurso de Algoritmos e Programação ...
 - » Objetivo do minicurso
 - » Ferramentas de Aprendizagem
 - » O que você vai ver neste curso?
 - » Como estudar
- » Da lógica à programação
 - » Por que aprender a programar?
 - » Aplicação de Algoritmos e Programação
 - » Algoritmos e Programação
- » Interpretadores e compiladores
 - » Interpretadores de algoritmos (Portugol IDE)
 - » Compiladores de linguagem (DevC++)
 - » Outras ferramentas de aprendizagem
- » Algoritmos
 - » Criando um algoritmo
 - » Tipos de algoritmo
- » Sobre narrativas
 - » Tipos de narrativas
 - » Exercitando narrativas
- » Sobre pseudocódigos
 - » Estrutura de um pseudocódigo
 - » Predefinições para escrita de pseudocódigos
- » Programa
 - » Estrutura de um programa
 - » Linguagem C - principais conceitos
 - » main()
 - » system()
 - » #include
 - » Constantes e comandos de atribuição
 - » Comandos de Entrada-Saída
 - » printf
 - » scanf
 - » Strings
- » Dados, variáveis e operadores
 - » Dados
 - » Operadores básicos
 - » Operadores aritméticos em C
 - » Operadores relacionais
 - » Operadores lógicos
 - » Precedência entre todos os Operadores
- » Estrutura de Controle - Decisão
 - » Estrutura de decisão simples
 - » Estrutura de decisão composta
 - » Estrutura de decisão encadeada
- » Estrutura de controle - Laços (repetição)
 - » Laço while (enquanto)
 - » Laço do...while (repita)
 - » Laço for (para)
- » Vetores e matrizes
 - » Vetores
 - » Matrizes

Algoritmos

Criando um algoritmo

Para que você possa criar um algoritmo é necessário que você analise cada tópico descrito abaixo:

- Entender o problema;
- Determinar os dados de entrada;
- Determinar como os dados serão processados;
- Determinar os dados de saída;
- Construir o algoritmo;
- Testar o algoritmo.

Entrada de dados → **Processamento de dados** → **Saída de dados**

Figura - Processo de criação de um Algoritmo. Fonte: própria.

Imagine que você vai escrever uma receita de bolo para alguém. Seu problema é "fazer um bolo" e então você precisa saber o que é um bolo, quais são os ingredientes da receita e como você irá misturá-los para ter um bolo.

ENTRADA: Para determinar quais são os dados de entrada você descreve os ingredientes;

PROCESSAMENTO: Para determinar como eles serão processados você descreve o modo de preparo;

SAÍDA: Para determinar os dados de saída você descreve como é o bolo desejado.

Por fim, você constrói o algoritmo (a receita) escrevendo um passo-a-passo. Para testar o algoritmo criado, você o executa. Neste caso, você segue a receita escrita avaliando os resultados de cada passo. Se no final você tiver um bolo conforme o desejado, sua receita está certa. Caso contrário será necessário corrigir os passos desta receita.

Relacionados aos tópicos vistos recentemente [Entenda melhor](#)

[Como criar um algoritmo](#)
 Descrição: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent pulvinar magna et vulputate dapibus. Duis at ornare lacus. ★★★★★

[Criando seu primeiro algoritmo](#)
 Descrição: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent pulvinar magna et vulputate dapibus. Duis at ornare lacus. ★★★★★

[Compiladores e interpretadores](#)
 Descrição: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent pulvinar magna et vulputate dapibus. Duis at ornare lacus. ★★★★★

[Lógica de programação básica](#)
 Descrição: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent pulvinar magna et vulputate dapibus. Duis at ornare lacus. ★★★★★

[Aplicação de algoritmos](#)
 Descrição: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent pulvinar magna et vulputate dapibus. Duis at ornare lacus. ★★★★★

Boas Vindas | Ciência da Computação |
 TADS | Licenciatura Física |
 Licenciatura Matemática | Engenharia
 Mecânica | Engenharia Elétrica |
 Engenharia Civil | Engenharia
 Produção | Seja Bem Vindos ao
 Minicurso de Algoritmos e
 Programação! |

Fonte: O autor.

5.2 DEFINIÇÃO DO EXPERIMENTO

O experimento proposto neste trabalho visa avaliar a experiência dos alunos ao interagir com o SR proposto, se comparado a um SR utilizando a abordagem Baseada em Conteúdo tradicional. Este experimento foi baseado nas seguintes hipóteses:

Figura 4 – Explicação da recomendação



Fonte: O autor.

- **H₀**: Não há diferenças na percepção do usuário da qualidade das recomendações recebidas utilizando a abordagem Baseada em Conteúdo tradicional e a proposta desse trabalho.
- **H₁**: Há diferenças na percepção do usuário da qualidade das recomendações recebidas utilizando a abordagem Baseada em Conteúdo tradicional e a proposta desse trabalho.

Para a execução do experimento, o SR proposto será comparado a abordagem Baseada em Conteúdo tradicional utilizando uma estratégia *Between Subjects*, i.e., os alunos serão divididos em dois grupos e cada grupo irá testar apenas um dos sistemas. Porém, para garantir que a única variável será qual o SR utilizado, ambos os grupos irão utilizar a mesma interface proposta para as recomendações.

O experimento será realizado em um minicurso de algoritmos do qual serão convidados a participar alunos de todos os cursos do Centro de Ciências Tecnológicas (CCT) da Universidade do Estado de Santa Catarina (UDESC). Isso porque em todos os cursos do CCT a disciplina de algoritmos está presente. O processo de design instrucional desse minicurso foi realizado por Santos (2017).

Nesse minicurso costumam se matricular em média 100 alunos por semestre. Sendo que desses, espera-se que pelo menos 60 alunos chegam até o final do curso e respondam ao questionário. Estima-se que o minicurso terá a duração de aproximadamente 45 dias e irá ocorrer no primeiro semestre de 2018, entre os meses de Abril

e Maio.

Os usuários ao se matricularem no minicurso serão aleatoriamente divididos nos dois grupos de usuários, de forma que ambos os grupos tenham uma quantidade de usuários similar. Além disso, os alunos irão receber um Termo de Consentimento Livre e Esclarecido (TCLE) que explica o objetivo do experimento e no qual eles irão consentir em participar dos experimentos e em permitir o uso dos resultados do experimento para essa pesquisa, sempre garantindo a anonimidade dos participants.

Ao final dos 45 dias do minicurso, os alunos irão receber um questionário para responder sobre a sua experiência com o SR. Esse questionário será produzindo utilizando um subconjunto das perguntas definidas por Pu, Chen e Hu (2011) (presente no Anexo A), que deverão ser traduzidas para o português para garantir o entendimento de todos os alunos. Os resultados dos questionários serão analisados através de métodos estatísticos para definir se existe diferença significativa entre a percepção do usuário da qualidade das recomendações nos dois SRs (o proposto e o Baseado em Conteúdo tradicional) e, se existir, qual deles teve o melhor desempenho.

5.3 TESTE PILOTO

Antes do experimento ser realizado com os alunos das disciplinas de algoritmos do CCT, será realizado um teste piloto com outros alunos da universidade que já realizaram essa disciplina. O objetivo desse teste piloto será avaliar os instrumentos do experimento, além de permitir encontrar problemas na experiência do usuário para serem corrigidos antes da execução do minicurso.

Para o teste piloto, os alunos receberão o TCLE ao qual eles também serão pedidos a assinar e serão pedidos a realizar algumas tarefas dentro do minicurso de algoritmos enquanto são observados por um pesquisador. As tarefas envolverão o acesso a alguns conceitos, exemplos, exercícios e materiais complementares, como também a ferramenta de recomendação. Ao final, os participantes do teste piloto irão responder ao questionário desenvolvido (o mesmo que será aplicado no experimento) e serão entrevistados pelo pesquisador sobre problemas encontrados nos instrumentos.

Com o resultado da observação do pesquisador e do questionário respondido pelos participantes do teste piloto será possível identificar possíveis problemas no sistema desenvolvido para serem corrigidos antes da execução do experimento. Já o objetivo da entrevista será exclusivamente para aprimorar os instrumentos (i.e., TCLE e o questionário desenvolvido).

5.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

Neste capítulo foi descrito o ambiente no qual o Sistema de Recomendação (SR) proposto será incorporado (AdaptWeb[®]) e definido o experimento para a avaliação da proposta desse trabalho. Nos SRs desenvolvidos no AdaptWeb[®], tanto para a proposta deste trabalho como para a abordagem Baseada em Conteúdo tradicional que será utilizada como parâmetro, o perfil do usuário é composto pelo conjunto de palavras-chave de todos os itens acessados pelo usuário, das cinco categorias apresentadas. Os itens a serem recomendados são os Links de Apoio, visto que os itens das outras categorias são estruturados pelo professor e estão fortemente relacionados à conceitos específicos.

Utilizando como base as diretrizes propostas por Pu, Chen e Hu (2012) foi proposta uma interface para a apresentação das recomendações no AdaptWeb[®], com o objetivo de que os usuários tenham acesso direto as recomendações na tela principal do ambiente do aluno e entendam melhor o porque aqueles itens foram recomendados. Como visto, essa interface será utilizada tanto pelo algoritmo proposto quanto pela abordagem tradicional.

O experimento visa avaliar a experiência do usuário com o SR proposto no Capítulo 4 em comparação à abordagem Baseada em Conteúdo tradicional. O experimento acontecerá no ambiente AdaptWeb[®] através de um minicurso de algoritmos desenvolvido no ambiente nos meses de Abril e Maio de 2018. Ao final do minicurso os alunos receberão um questionário para responder sobre a sua experiência ao interagir com o SR, que será adaptado do conjunto de questões definidas por Pu, Chen e Hu (2011). Ao final do experimento, espera-se chegar a uma conclusão se existe diferença na percepção do usuário sobre a qualidade das recomendações recebidas ao utilizar um SR Sensível ao Tempo, que se adapta as variações de interesse do aluno, em relação a uma abordagem tradicional de recomendação.

6 CONSIDERAÇÕES FINAIS

Sistemas de Recomendação (SR) são ferramentas de software que sugerem itens para os usuários de forma automatizada e personalizada, sem a necessidade do usuário formular uma consulta para encontrar os itens do seu interesse. Esses sistemas são explorados em Ambientes Virtuais de Aprendizagem (AVA) com o objetivo de reduzir alguns problemas existentes nesses ambientes quando a quantidade de materiais disponíveis é grande, tais como: sobrecarga cognitiva, dificuldade de encontrar os materiais do seu interesse e muitos materiais nunca serem utilizados.

Pesquisadores da área argumentam que os algoritmos de SRs tradicionais não são suficientes para os AVAs (VERBERT et al., 2012; DRACHSLER et al., 2015), sendo necessário um nível maior de personalização a situação do usuário, como considerar dimensões do contexto. Para isso, em Borba, Gasparini e Lichtnow (2017) foi realizado um mapeamento sistemático da literatura com o objetivo de identificar como os SRs Sensíveis ao Contexto Temporal (também chamados de SR Sensíveis ao Tempo) são utilizados. Nesse estudo, foram considerados todos os domínios de aplicação e não apenas o domínio educacional.

Foi observado que, dos 88 artigos que utilizam esse tipo de SR, apenas quatro são aplicados no domínio educacional e esses trabalhos carecem em avaliações em ambientes reais de uso ou que utilizem bases de dados educacionais. Analisando esses 88 artigos, também foi possível categorizar os SRs propostos nesses trabalhos pela forma que eles utilizam o tempo para a recomendação. As sete categorias criadas como resultado do mapeamento são apresentadas na Seção 2.3.

O objetivo desse trabalho é a criação de perfis de usuário que levem em conta a mudança dos interesses destes usuários ao longo do tempo. Esses perfis considerando o contexto temporal serão aplicados no algoritmo de recomendação proposto nesse trabalho. Dentre as categorias de SR Sensíveis ao Tempo presentes na Seção 2.3, a proposta desse trabalho se encaixa no *Decay*.

O algoritmo proposto no Capítulo 4 combina a (1) similaridade do perfil do usuário (representado pelos materiais acessados pelo usuário) com os itens disponíveis para a recomendação com a (2) recência dos materiais acessados pelo usuários, além da (3) informação se aquele item disponível para a recomendação já foi acessado ou não. A proposta leva em conta que o ritmo de estudo dos alunos pode ser diferente, portanto a recência é considerada em relação a sequência de itens acessados e não ao tempo absoluto (em segundos) desde o acesso. Dessa forma, para cada aluno o decaimento acaba sendo personalizado ao seu ritmo de estudo. Também

é considerado que itens já acessados podem ser recomendados novamente, porém esses itens tem um probabilidade menor de ser recomendados do que itens ainda não acessados.

Como continuação desse trabalho está a etapa de implementação da proposta e a experimentação utilizando um ambiente real de uso. A proposta desse trabalho será incorporada ao ambiente AdaptWeb® e será avaliado através de um minicurso de algoritmos que é ministrado no ambiente todo semestre, no qual participam alunos da primeira fase dos cursos do Centro de Ciências Tecnológicas (CCT) da Universidade do Estado de Santa Catarina (UDESC). O algoritmo proposto será comparado a abordagem Baseada em Conteúdo tradicional através de um experimento utilizando um estratégia *Between Subjects*.

O objetivo do experimento é verificar se existe diferença na percepção do usuário sobre a qualidade das recomendações do algoritmo proposto em relação a abordagem Baseada em Conteúdo tradicional. A percepção do usuário será capturada utilizando o questionário proposto por Pu, Chen e Hu (2011) para identificar a percepção do usuário sobre a qualidade das recomendações, presente no Anexo A.

6.1 CRONOGRAMA

O cronograma proposto para a execução do restante desse trabalho pode ser visto no Apêndice A. A etapa de implementação da interface e dos algoritmos de recomendação está previsto para começar no mês de Dezembro e ir até a metade do mês de Fevereiro. O período de preparação para o experimento irá acontecer da metade de Fevereiro até o final do mês de Março, incluindo a mobilização de participantes para o experimento. O experimento propriamente dito está previsto para acontecer durante o mês de Abril e até a metade do mês de Maio. Da metade do mês de Maio até o final do mês de Junho está previsto para ser realizadas as análises dos resultados e a finalização da dissertação.

6.2 RESULTADOS PARCIAIS

Como resultados parciais desse trabalho tem-se as seguintes publicações:

- BORBA, E. J.; Gasparini, I.; LICHTNOW, D. Time-Aware Recommender Systems: A Systematic Mapping. International Conference on Human-Computer Interaction (HCI), Vancouver, Part II, LNCS 10272, v. II, p. 464-479, 2017.
- BORBA, E. J.; GASPARINI, I.; LICHTNOW, D. The Use of Time Dimension in Recommender Systems for Learning. Proceedings of the 19th International Con-

ference on Enterprise Information Systems (ICEIS), Porto (Portugal) 2017. v. 2. p. 600-609.

- BORBA, E. J.; GASPARINI, I.; LICHTNOW, D. Sistema de Recomendação Sensível ao Tempo em Ambientes Educacionais. IV Workshop de Teses e Dissertações em IHC (WTD-IHC), Joinville, 2017.

REFERÊNCIAS

- 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**, IEEE, v. 17, n. 6, p. 734–749, 2005.
- ADOMAVICIUS, G.; TUZHILIN, A. Context-aware recommender systems. In: **Recommender systems handbook**. [S.l.]: Springer, 2011. p. 217–253.
- BEEL, J. et al. Towards reproducibility in recommender-systems research. **User modeling and user-adapted interaction**, Springer, v. 26, n. 1, p. 69–101, 2016.
- BENCIC, A.; BIELIKOVA, M. Action suggestion using situation rules. In: IEEE. **Semantic and Social Media Adaptation and Personalization (SMAP), 2012 Seventh International Workshop on**. [S.l.], 2012. p. 48–53.
- BOBADILLA, J. et al. Recommender systems survey. **Knowledge-based systems**, Elsevier, v. 46, p. 109–132, 2013.
- BORBA, E. J. de. **Sistema de Recomendação de Objetos de Aprendizagem no Ambiente AdaptWeb®**. 2015. TCC (Graduação)-Universidade do Estado de Santa Catarina, Curso de Ciência da Computação, Joinville.
- BORBA, E. J. de; GASPARINI, I.; LICHTNOW, D. Time-aware recommender systems: A systematic mapping. In: SPRINGER. **International Conference on Human-Computer Interaction**. [S.l.], 2017. p. 464–479.
- BRUSILOVSKY, P. Methods and techniques of adaptive hypermedia. In: **Adaptive hypertext and hypermedia**. [S.l.]: Springer, 1998. p. 1–43.
- BURKE, R. Hybrid recommender systems: Survey and experiments. **User modeling and user-adapted interaction**, Springer, v. 12, n. 4, p. 331–370, 2002.
- CAMPOS, P. G.; DÍEZ, F.; CANTADOR, I. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. **User Modeling and User-Adapted Interaction**, Springer, v. 24, n. 1-2, p. 67–119, 2014.
- CHEN, G.; KOTZ, D. et al. **A survey of context-aware mobile computing research**. [S.l.], 2000.
- CHRISTOPHER, D. M.; PRABHAKAR, R.; HINRICH, S. Introduction to information retrieval. **An Introduction To Information Retrieval**, v. 151, p. 177, 2008.
- DEY, A. K. Understanding and using context. **Personal and ubiquitous computing**, Springer-Verlag, v. 5, n. 1, p. 4–7, 2001.
- DRACHSLER, H. et al. Panorama of recommender systems to support learning. In: **Recommender systems handbook**. [S.l.]: Springer, 2015. p. 421–451.
- FAN, X. et al. Modeling temporal effectiveness for context-aware web services recommendation. In: IEEE. **Web Services (ICWS), 2015 IEEE International Conference on**. [S.l.], 2015. p. 225–232.

GASPARINI, I. **Interface Adaptativa no ambiente AdaptWeb**. Dissertação (Mestrado) — Mestrado em Ciência da Computação—Instituto de Informática, UFRGS, Porto Alegre, 2003.

GASPARINI, I. et al. AdaptWeb®- evolução e desafios. **Cadernos de Informática**, Universidade Federal do Rio Grande do Sul, v. 4, n. 2, p. 47–56, 2009.

HAWALAH, A.; FASLI, M. Utilizing contextual ontological user profiles for personalized recommendations. **Expert Systems with Applications**, Elsevier, v. 41, n. 10, p. 4777–4797, 2014.

IOFCIU, T.; DEMARTINI, G. Time based tag recommendation using direct and extended users sets. **ECML PKDD discovery Challenge**, p. 99–107, 2009.

JANNACH, D. et al. **Recommender systems: an introduction**. [S.l.]: Cambridge University Press, 2010.

KELLY, D. et al. Methods for evaluating interactive information retrieval systems with users. **Foundations and Trends® in Information Retrieval**, Now Publishers, Inc., v. 3, n. 1–2, p. 1–224, 2009.

KUSHWAHA, N. et al. Inclusion of semantic and time-variant information using matrix factorization approach for implicit rating of last. fm dataset. **Arabian Journal for Science and Engineering**, Springer, v. 41, n. 12, p. 5077–5092, 2016.

LOPS, P.; GEMMIS, M. D.; SEMERARO, G. Content-based recommender systems: State of the art and trends. In: **Recommender systems handbook**. [S.l.]: Springer, 2011. p. 73–105.

LUCKESI, C. C. **Avaliação da aprendizagem escolar: estudos e proposições**. [S.l.]: Cortez editora, 2014.

LUO, J. et al. A context-aware personalized resource recommendation for pervasive learning. **Cluster Computing**, Springer, v. 13, n. 2, p. 213–239, 2010.

MICHAELIS, D. Disponível em:< <http://michaelis.uol.com.br>>. **Acesso em**, v. 17, 2011.

PETERSEN, K. et al. Systematic mapping studies in software engineering. In: **EASE**. [S.l.: s.n.], 2008. v. 8, p. 68–77.

PU, P.; CHEN, L.; HU, R. A user-centric evaluation framework for recommender systems. In: ACM. **Proceedings of the fifth ACM conference on Recommender systems**. [S.l.], 2011. p. 157–164.

PU, P.; CHEN, L.; HU, R. Evaluating recommender systems from the user's perspective: survey of the state of the art. **User Modeling and User-Adapted Interaction**, Springer, v. 22, n. 4, p. 317–355, 2012.

QIAO, L.; ZHANG, R. Personalized recommendation algorithm based on situation awareness. In: IEEE. **Logistics, Informatics and Service Sciences (LISS), 2015 International Conference on**. [S.l.], 2015. p. 1–4.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Introduction to recommender systems handbook. In: **Recommender systems handbook**. [S.l.]: Springer, 2011. p. 1–35.

SALTON, G. A document retrieval system for man-machine interaction. In: ACM. **Proceedings of the 1964 19th ACM national conference**. [S.l.], 1964. p. 122–301.

SANTOS, L. V. dos. **O Uso da Metodologia Addie no Design Instrucional de um Minicurso Online de Algoritmos e Programação para o Ensino Superior**. Dissertação (Mestrado) — Mestrado em Ensino de Ciências, Matemática e Tecnologias, Centro de Ciências Tecnológicas, UDESC, Joinville., 2017. A ser publicado.

SCHILIT, B.; ADAMS, N.; WANT, R. Context-aware computing applications. In: IEEE. **Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on**. [S.l.], 1994. p. 85–90.

SCHMIDT, A.; BEIGL, M.; GELLERSEN, H.-W. There is more to context than location. **Computers & Graphics**, Elsevier, v. 23, n. 6, p. 893–901, 1999.

SHANI, G.; GUNAWARDANA, A. Evaluating recommendation systems. **Recommender systems handbook**, Springer, p. 257–297, 2011.

TORRES, R. Personalização na internet: como descobrir os hábitos de consumo de seus clientes, fidelizá-los e aumentar o lucro de seu negócio. **São Paulo: Novatec**, 2004.

VERBERT, K. et al. Context-aware recommender systems for learning: a survey and future challenges. **IEEE Transactions on Learning Technologies**, IEEE, v. 5, n. 4, p. 318–335, 2012.

WEI, C.; KHOURY, R.; FONG, S. Web 2.0 recommendation service by multi-collaborative filtering trust network algorithm. **Information Systems Frontiers**, Springer, v. 15, n. 4, p. 533–551, 2013.

ZIMMERMANN, A.; LORENZ, A.; OPPERMAN, R. An operational definition of context. **Context**, Springer, v. 7, p. 558–571, 2007.

Apêndices

APÊNDICE A – CRONOGRAMA DAS ATIVIDADES

Este apêndice apresenta o cronograma das atividades durante todo o desenvolvimento dessa pesquisa e das etapas que ainda estão por vir. Cada atividade está numerada no cronograma e explicada em detalhe abaixo.

Atividade	Dez-17				Jan-18				Fev-18				Mar-18				Abril-18				Mai-18				Jun-18			
1																												
2																												
3																												
4																												
5																												
6																												
7																												
8																												
9																												
10																												
11																												
12																												

1. Implementar a interface das recomendações
2. Incorporar ferramentas para facilitar o cálculo da similaridade do perfil do usuário com os itens
3. Implementar os algoritmos de recomendação (tradicional e a proposta desse trabalho)
4. Melhorar a categorização (palavras-chave) dos materiais e Links de Apoio presentes no minicurso de algoritmos
5. Realizar testes para garantir a funcionalidade dos algoritmos de recomendação desenvolvidos
6. Produzir os instrumentos necessários para o experimento
7. Convidar alunos a participar do experimento
8. Realizar o experimento
9. Estudar e definir as técnicas de análise dos dados
10. Analisar os dados (resultados do experimento)
11. Escrever a dissertação

Anexos

ANEXO A – 60 QUESTÕES DO FRAMEWORK DE AVALIAÇÃO DE SISTEMAS DE RECOMENDAÇÃO RESQUE

A.1 QUALITY OF RECOMMENDED ITEMS

A.1.1 Accuracy

- The items recommended to me matched my interests.*
- The recommender gave me good suggestions.
- I am not interested in the items recommended to me (reverse scale).

A.1.2 Relative Accuracy

- The recommendation I received better fits my interests than what I may receive from a friend.
- A recommendation from my friends better suits my interests than the recommendation from this system (reverse scale).

A.1.3 Familiarity

- Some of the recommended items are familiar to me.
- I am not familiar with the items that were recommended to me (reverse scale).

A.1.4 Attractiveness

- The items recommended to me are attractive.

A.1.5 Enjoyability

- I enjoyed the items recommended to me.

A.1.6 Novelty

- The items recommended to me are novel and interesting.*
- The recommender system is educational.
- The recommender system helps me discover new products.
- I could not find new items through the recommender (reverse scale).

A.1.7 Diversity

- The items recommended to me are diverse.*
- The items recommended to me are similar to each other (reverse scale).*

A.1.8 Context Compatibility

- I was only provided with general recommendations.
- The items recommended to me took my personal context requirements into consideration.
- The recommendations are timely.

A.2 INTERACTION ADEQUACY

- The recommender provides an adequate way for me to express my preferences.
- The recommender provides an adequate way for me to revise my preferences.
- The recommender explains why the products are recommended to me.*

A.3 INTERFACE ADEQUACY

- The recommender's interface provides sufficient information.
- The information provided for the recommended items is sufficient for me.
- The labels of the recommender interface are clear and adequate.
- The layout of the recommender interface is attractive and adequate.*

A.4 PERCEIVED EASE OF USE

A.4.1 Ease of Initial Learning

- I became familiar with the recommender system very quickly.
- I easily found the recommended items.
- Looking for a recommended item required too much effort (reverse scale).

A.4.2 Ease of Preference Elicitation

- I found it easy to tell the system about my preferences.
- It is easy to learn to tell the system what I like.
- It required too much effort to tell the system what I like (reversed scale).

A.4.3 Ease of Preference Revision

- I found it easy to make the system recommend different things to me.
- It is easy to train the system to update my preferences.
- I found it easy to alter the outcome of the recommended items due to my preference changes.
- It is easy for me to inform the system if I dislike/like the recommended item.
- It is easy for me to get a new set of recommendations.

A.4.4 Ease of Decision Making

- Using the recommender to find what I like is easy.
- I was able to take advantage of the recommender very quickly.
- I quickly became productive with the recommender.
- Finding an item to buy with the help of the recommender is easy.*
- Finding an item to buy, even with the help of the recommender, consumes too much time.

A.5 PERCEIVED USEFULNESS

- The recommended items effectively helped me find the ideal product.*
- The recommended items influence my selection of products.
- I feel supported to find what I like with the help of the recommender.*
- I feel supported in selecting the items to buy with the help of the recommender.

A.6 CONTROL/TRANSPARENCY

- I feel in control of telling the recommender what I want.
- I don't feel in control of telling the system what I want.
- I don't feel in control of specifying and changing my preferences (reverse scale).
- I understood why the items were recommended to me.
- The system helps me understand why the items were recommended to me.
- The system seems to control my decision process rather than me (reverse scale).

A.7 ATTITUDES

- Overall, I am satisfied with the recommender.*
- I am convinced of the products recommended to me.*
- I am confident I will like the items recommended to me. *
- The recommender made me more confident about my selection/decision.
- The recommended items made me confused about my choice (reverse scale).
- The recommender can be trusted.

A.8 BEHAVIORAL INTENTIONS

A.8.1 Intention to Use the System

- If a recommender such as this exists, I will use it to find products to buy.

A.8.2 Continuance and Frequency

- I will use this recommender again.*
- I will use this type of recommender frequently.
- I prefer to use this type of recommender in the future.

A.8.3 Recommendation to Friends

- I will tell my friends about this recommender.*

A.8.4 Purchase Intention

- I would buy the items recommended, given the opportunity.*