

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

EDUARDO JOSÉ DE BORBA

**UM SISTEMA DE RECOMENDAÇÃO PARA AMBIENTES VIRTUAIS
DE APRENDIZAGEM QUE CONSIDERA A VARIAÇÃO TEMPORAL
DOS INTERESSES DOS ALUNOS: CONCEPÇÃO, EXPERIMENTO E
AVALIAÇÃO**

JOINVILLE

2018

EDUARDO JOSÉ DE BORBA

**UM SISTEMA DE RECOMENDAÇÃO PARA AMBIENTES VIRTUAIS
DE APRENDIZAGEM QUE CONSIDERA A VARIAÇÃO TEMPORAL
DOS INTERESSES DOS ALUNOS: CONCEPÇÃO, EXPERIMENTO E
AVALIAÇÃO**

Dissertação 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

2018

EDUARDO JOSÉ DE BORBA

**UM SISTEMA DE RECOMENDAÇÃO PARA AMBIENTES VIRTUAIS DE
APRENDIZAGEM QUE CONSIDERA A VARIAÇÃO TEMPORAL DOS
INTERESSES DOS ALUNOS: CONCEPÇÃO, EXPERIMENTO E AVALIAÇÃO**

Dissertação 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

Universidade do Estado de Santa Catarina
(UDESC)

Dra. Rebeca Schroeder Freitas

Universidade do Estado de Santa Catarina
(UDESC)

Dr. Tiago T. Primo

Universidade Federal de Pelotas (UFPel)

Dr. José Palazzo M. de Oliveira

Universidade Federal do Rio Grande do Sul (UFRGS)

Joinville, 28 de Junho de 2018

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 da situação do usuário, como considerar informações do seu contexto. O objetivo desse trabalho é avaliar se considerar a variação temporal dos interesses do usuário em itens acessados anteriormente em SRs voltados a AVAs influencia o desempenho da abordagem de recomendação e a percepção dos alunos sobre as recomendações recebidas. O algoritmo com decaimento 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 desses 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 à sequência de itens acessados e não ao tempo absoluto (em segundos) desde o acesso. A proposta desse trabalho foi incorporada ao ambiente AdaptWeb® e avaliada através de um experimento utilizando um Minicurso de Algoritmos e Linguagem de Programação ministrado no ambiente. O algoritmo proposto foi comparado à abordagem Baseada em Conteúdo Tradicional utilizando uma estratégia *Between Subjects*. Os resultados mostraram que existe diferença significativa em relação a abordagem Tradicional na Cobertura, F-measure e em uma das questões sobre a percepção do usuário e que a abordagem com Decaimento teve resultados melhores nessas métricas. Portanto, as duas hipóteses alternativas definidas para o experimento foram aceitas e indicam que considerar o Decaimento em um SR para AVAs influencia positivamente o desempenho do algoritmo e a percepção dos alunos sobre as recomendações.

Palavras-chaves: Sistema de Recomendação; Sensível ao Tempo; Sensível ao Contexto; Decaimento; 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 them. 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 accessed. 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 evaluate whether the use of Decay on user's interest for past items on RS for VLEs influences the algorithm performance and the user's perception of the recommendations. The proposed algorithm combines (1) similarity between user profile (represented by the 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 was 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 was incorporated to the AdaptWeb[®] environment and evaluated through an experiment using an Algorithms and Programming Language's course. The proposal was compared to the Content-Based traditional approach through using a Between Subjects strategy. The results show that there is significant difference on the Coverage, F-measure and in one of the questions about the user's perception and that the approach with Decay had better results in these metrics. Therefore, both alternative hypothesis defined for the experiment were accepted and it indicates that considering the Decay on a RS for VLEs influences the algorithm performance and the user's perception of the recommendations.

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

LISTA DE ILUSTRAÇÕES

Figura 1 – Construtos do <i>framework</i> de avaliação de SRs pela perspectiva do usuário	32
Figura 2 – Diagrama de metáfora de sistema	50
Figura 3 – Estrutura do AdaptWeb®	56
Figura 4 – Tela do Ambiente de Aula com as Áreas em Destaque	59
Figura 5 – Explicação da recomendação	60
Figura 6 – Fluxograma de uso das técnicas estatísticas	67
Figura 7 – Boxplot dos links acessados por aluno	101
Figura 8 – Boxplot dos links avaliados positivamente por aluno	103
Figura 9 – Boxplot dos links acessados por aluno	104
Figura 10 – Boxplot da precisão dos algoritmos de recomendação por aluno	106
Figura 11 – Boxplot da cobertura dos algoritmos de recomendação por aluno	107
Figura 12 – Boxplot do F-measure dos dois algoritmos	108
Figura 13 – Boxplot da questão 1	109
Figura 14 – Boxplot da questão 2	110
Figura 15 – Boxplot da questão 3	111
Figura 16 – Boxplot da questão 4	112
Figura 17 – Boxplot da questão 5	113
Figura 18 – Boxplot da questão 6	114
Figura 19 – Boxplot da questão 7	115
Figura 20 – Boxplot da questão 8	116
Figura 21 – Boxplot da questão 9	117
Figura 22 – Boxplot da questão 10	118
Figura 23 – Boxplot da questão 11	119
Figura 24 – Boxplot da questão 12	120
Figura 25 – Boxplot da questão 13	121
Figura 26 – Intervenção 1 (16/04/18)	128
Figura 27 – Intervenção 2 (18/04/18)	129
Figura 28 – Intervenção 3 (20/04/18)	129
Figura 29 – Intervenção 4 (22/04/18)	130
Figura 30 – Intervenção 5 (25/04/18)	130
Figura 31 – Intervenção 6 (29/04/18)	131
Figura 32 – Intervenção 7 (02/05/18)	131
Figura 33 – Intervenção 8 (06/05/18)	132
Figura 34 – Intervenção 9 (09/05/18)	132
Figura 35 – Intervenção 10 (10/05/18)	132

Figura 36 – Intervenção 11 (11/05/18) 133

Figura 37 – Intervenção 12 (13/05/18) 133

LISTA DE TABELAS

Tabela 1 – Comparação dos trabalhos relacionados	45
Tabela 2 – Cenário de Uso: comportamento do algoritmo após o acesso de itens relacionados a pilhas.	52
Tabela 3 – Cenário de Uso: comportamento do algoritmo após o acesso de itens relacionados a filas.	52
Tabela 4 – Cenário de Uso: comportamento do algoritmo após o aluno voltar a acessar itens relacionados a pilhas.	52
Tabela 5 – Dicionário de palavras-chave	58
Tabela 6 – Dados dos Participantes do Teste Piloto	63
Tabela 7 – Divisão dos alunos de acordo com os critérios	65
Tabela 8 – Uso do Minicurso de Algoritmos e do SR	66
Tabela 9 – Respostas ao Questionário	70
Tabela 10 – Categorização da questão discursiva do questionário	73
Tabela 11 – Palavras-chave associadas aos Conceitos	88
Tabela 12 – Palavras-chave associadas aos Materiais Complementares	90
Tabela 13 – Palavras-chave associadas aos Links de Apoio	96

LISTA DE ABREVIATURAS E SIGLAS

AdaptWeb	Ambiente de Ensino-Aprendizagem Adaptativo na Web
AVA	Ambiente Virtual de Aprendizagem
CCT	Centro de Ciências Tecnológicas
CE	Critério de Exclusão
CI	Critério de Inclusão
CO	Critério Objetivo
MBA	Mecanismos de Busca Científica
PPGCA	Programa de Pós-Graduação em Computação Aplicada
ResQue	Recommender Systems' Quality of User Experience
SR	Sistema de Recomendação
SUMI	Software Usability Measurement Inventory
TAM	Technology Acceptance Model
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	14
1.1	Problema	15
1.2	Objetivos	15
1.2.1	Objetivo Geral	15
1.2.2	Objetivos Específicos	15
1.3	Escopo	16
1.4	Metodologia	16
1.5	Estrutura	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Sistemas de Recomendação	17
2.1.1	Baseada em Conteúdo	18
2.1.2	Filtragem Colaborativa	20
2.1.3	Baseada em Conhecimento	22
2.1.4	Híbrida	23
2.2	Sistemas de Recomendação Sensíveis ao Contexto	23
2.3	Sistemas de Recomendação Sensíveis ao Tempo	25
2.3.1	Restriction	27
2.3.2	Micro-profile	28
2.3.3	Bias	28
2.3.4	Decay	28
2.3.5	Time Rating	29
2.3.6	Novelty	29
2.3.7	Sequence	29
2.4	Projeto de Interface de Recomendações	29
2.5	Avaliação de Sistemas de Recomendação	30
2.6	Considerações sobre o capítulo	32
3	TRABALHOS RELACIONADOS	34
3.1	Fan et al. 2015	34
3.2	Luo et al. 2010	36
3.3	Benčič e Bieliková 2012	38
3.4	Hawalah e Fasli 2014	39
3.5	Qiao e Zhang 2012	41
3.6	Kushwaha et al. 2016	42

3.7	Wei, Khoury e Fong 2013	43
3.8	Considerações sobre o capítulo	44
4	SR SENSÍVEL AO TEMPO	47
4.1	Algoritmo Proposto	47
4.2	Cenário de Uso	51
4.3	Discussão Sobre o Algoritmo Proposto	53
5	EXPERIMENTO	55
5.1	Planejamento	55
5.1.1	Descrição do Ambiente AdaptWeb®	55
5.1.1.1	Estrutura do AdaptWeb®	55
5.1.1.2	Sistema de Recomendação no AdaptWeb®	57
5.1.1.3	Apresentação das Recomendações	58
5.1.2	Definição do experimento	60
5.1.3	Teste piloto	62
5.2	Execução	64
5.3	Análise dos Resultados	65
5.3.1	Técnicas de Análise Estatística	66
5.3.2	Análise do Desempenho da Abordagem de Recomendação	67
5.3.3	Análise da Percepção do Usuário	70
5.3.4	Questão Discursiva sobre o SR	72
5.3.5	Limitações e Ameaças à Validade do Experimento	74
5.4	Considerações sobre o capítulo	75
6	CONSIDERAÇÕES FINAIS	78
	REFERÊNCIAS	82
	APÊNDICES	85
	APÊNDICE A – PALAVRAS-CHAVE DOS MATERIAIS NO MINICURSO DE ALGORITMOS E LINGUAGEM DE PROGRAMAÇÃO	86
A.1	Conceitos	86
A.2	Materiais Complementares	88
A.3	Links de Apoio	90

	APÊNDICE B – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO	97
B.1	Descrição do Minicurso	97
B.2	Procedimento	97
B.3	Riscos e Desconfortos	97
B.4	Benefícios da sua Participação	97
B.5	Custos	98
B.6	Confidencialidade	98
B.7	Dúvidas	98
	APÊNDICE C – TESTE PILOTO	100
	APÊNDICE D – ANÁLISE ESTATÍSTICA DO DESEMPENHO DO SISTEMA DE RECOMENDAÇÃO	101
D.1	Links Acessados por Aluno	101
D.2	Links Avaliados Positivamente por Aluno	103
D.3	Links avaliados negativamente por aluno que acessou pelo menos uma recomendação	104
D.4	Precisão dos algoritmos de recomendação por aluno	106
D.5	Cobertura dos algoritmos de recomendação por aluno	107
D.6	F-measure do algoritmo de recomendação por aluno	108
	APÊNDICE E – ANÁLISE ESTATÍSTICA DAS QUESTÕES DO QUESTIONÁRIO	109
E.1	Questão 1: Os itens recomendados corresponderam com os meus interesses.	109
E.2	Questão 2: Os itens recomendados para mim são diversificados (o sistema se preocupa em trazer itens diferentes a cada recomendação).	110
E.3	Questão 3: Os itens recomendados corresponderam aos interesses e necessidades que eu tinha no momento.	111
E.4	Questão 4: As recomendações são feitas no momento adequado.	112
E.5	Questão 5: O sistema de recomendação explica porque os links são recomendados para mim.	113
E.6	Questão 6: A informação apresentada na interface para os itens recomendados é suficiente para mim.	114
E.7	Questão 7: O layout do sistema de recomendação é atrativo e adequado.	115
E.8	Questão 8: Eu encontrei facilmente o local onde os itens são recomendados.	116

E.9	Questão 9: Eu percebi que o sistema de recomendação aprendia sobre minhas necessidades/preferências conforme eu avançava na disciplina.	117
E.10	Questão 10: É fácil encontrar um item para estudar com a ajuda do sistema de recomendação.	118
E.11	Questão 11: Eu me senti apoiado para encontrar itens do meu interesse com a ajuda do sistema de recomendação.	119
E.12	Questão 12: Eu entendi porque os itens foram recomendados para mim.	120
E.13	Questão 13: No geral, estou satisfeito com o sistema de recomendação.	121

ANEXOS 122

ANEXO A – 60 QUESTÕES DO FRAMEWORK DE AVALIAÇÃO DE SISTEMAS DE RECOMENDAÇÃO RESQUE . . . 123

A.1	Quality of Recommended Items	123
A.1.1	Accuracy	123
A.1.2	Relative Accuracy	123
A.1.3	Familiarity	123
A.1.4	Attractiveness	123
A.1.5	Enjoyability	123
A.1.6	Novelty	123
A.1.7	Diversity	124
A.1.8	Context Compatibility	124
A.2	Interaction Adequacy	124
A.3	Interface Adequacy	124
A.4	Perceived Ease of Use	124
A.4.1	Ease of Initial Learning	124
A.4.2	Ease of Preference Elicitation	125
A.4.3	Ease of Preference Revision	125
A.4.4	Ease of Decision Making	125
A.5	Perceived Usefulness	125
A.6	Control/Transparency	126
A.7	Attitudes	126
A.8	Behavioral Intentions	126
A.8.1	Intention to Use the System	126
A.8.2	Continuance and Frequency	126
A.8.3	Recommendation to Friends	126
A.8.4	Purchase Intention	127

ANEXO B – INTERVENÇÕES	128
ANEXO C – DESAFIOS	134

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 têm aplicado técnicas de personalização para selecionar os itens mais adequados para cada estudante, considerando o seu conhecimento, objetivos, preferências e necessidades (BRUSILOVSKY, 1998). Os Sistemas de Recomendação (SR) 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 as interações do usuário com 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: **"Considerar a variação temporal dos interesses do aluno por conteúdos acessados anteriormente influencia o desempenho da abordagem de recomendação e a percepção dos alunos sobre as recomendações recebidas?"**.

1.2 OBJETIVOS

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

1.2.1 Objetivo Geral

Avaliar em um ambiente real de uso se considerar a variação temporal dos interesses dos alunos em sistemas de recomendação voltados a AVAs influencia o desempenho da abordagem de recomendação e a percepção dos alunos sobre as recomendações recebidas.

1.2.2 Objetivos Específicos

- Identificar as formas de utilizar os aspectos temporais do contexto do usuário em um algoritmo de recomendação;
- Definir como o aspecto temporal pode ser utilizado em sistemas de recomendação para AVAs;
- Conceber um algoritmo de recomendação considerando o decaimento do interesse dos alunos por itens acessados anteriormente no contexto educacional;
- Realizar um experimento com usuários reais.

1.3 ESCOPO

Esse trabalho não considera outras dimensões do contexto além do tempo na recomendação, e dentro do uso do contexto temporal apenas a categoria de Decaimento foi aplicada neste trabalho, que está relacionada à perda de interesse por itens acessados anteriormente. Além disso, a única abordagem de recomendação utilizada é 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 é feita apenas em um ambiente educacional, mesmo sendo possível aplicá-lo em outros domínios de aplicação. Não é avaliado, nesse trabalho, o impacto da proposta na aprendizagem dos alunos, pois esse tipo de avaliação exigiria 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 tanto qualitativa, através do questionário aplicado para extrair a percepção dos alunos, quanto quantitativa, através das métricas calculadas para medir o desempenho dos algoritmos. Os objetivos dessa pesquisa têm caráter Explicativo, visando identificar fatores que influenciam o desempenho de um algoritmo de recomendação e a percepção dos alunos sobre as recomendações. O procedimento utilizado para o desenvolvimento dessa pesquisa é Experimental, onde os objetos de estudos são os dados coletados da interação dos alunos com o SR e 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 Decaimento 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 e os resultados do experimento. Por último, o Capítulo 6 apresenta as considerações finais deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo são apresentados os principais conceitos relacionados à 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 proveem 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 tem 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 itens similares a este (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 automatização 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 item 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, torna-se 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 *offline* (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. 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, torna-se 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 Baseados 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: o contexto filtra os dados que representam o usuário e esses dados servem como entrada para um algoritmo tradicional de recomendação;
- Pós-Filtragem Contextual: uma abordagem tradicional de recomendação é utilizada para gerar uma lista de itens a serem recomendados e depois esses itens são filtrados de acordo com o contexto do usuário;

- Modelagem Contextual: 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.

Na área educacional é possível considerar o perfil do usuário unicamente com base nas suas ações dentro da disciplina em questão ou então com base no seu perfil dentro de um determinado curso composto de diversas disciplinas. Além disso, é possível gerar recomendações para o usuário que sejam apenas de itens disponíveis no sistema que geralmente são cadastrados por um professor (conceito de mundo fechado) ou então buscar materiais em outros repositórios ou sites externos através do uso de robôs (conceito de mundo aberto). Dessa forma, a proposta de um SR para a área educacional envolve mais variáveis e acaba se tornando mais complexa.

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 à questão temporal.

2.3 SISTEMAS DE RECOMENDAÇÃO SENSÍVEIS 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 têm 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 de tempo são hierárquicas, e.g., um dia tem 24 horas, uma hora tem 60 minutos e 1 minuto tem 60 segundos.

Em Borba, Gasparini e Lichtnow (2017) um mapeamento sistemático foi conduzido sobre os SR Sensíveis ao Tempo utilizando a metodologia de Petersen et al. (2008). Esse mapeamento sistemático considerou artigos de qualquer domínio de aplicação, e não apenas trabalhos na área educacional. A principal questão de pesquisa desse mapeamento é “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 “Como os algoritmos de recomendação utilizam o tempo?”; “Qual é a diferença entre o uso do tempo em diferentes domínios de aplicação?”; e “Que outras dimensões são utilizadas juntamente com o contexto temporal?”.

Com base nas questões de pesquisa, foi definido o seguinte argumento de busca (*time-aware OR context-aware*) AND (*“recommender system”*), que tem o objetivo de encontrar artigos sobre SRs Sensíveis ao Tempo ou SRs Sensíveis ao Contexto que utilizem o tempo como uma de suas dimensões. O argumento de busca foi aplicado em três Mecanismos de Busca Acadêmica (MBA): *IEEE Xplorer*, *Scopus*

e *Springer Link*. Esses MBAs foram os escolhidos por terem um grande acervo de artigos da área da Computação e possuírem mecanismos de busca e de filtro necessários (BORBA; GASPARINI; LICHTNOW, 2017). A busca foi realizada procurando pelos argumentos de busca no Título, Resumo ou Palavras-chave.

Três Critérios Objetivos (CO) foram definidos para filtrar artigos mais relevantes para a pesquisa. O primeiro CO é que os artigos tenham sido publicados nos 10 anos anteriores à realização do mapeamento, i.e., de 2006 à 2016. O segundo CO é que apenas artigos que estiverem disponíveis para o download completo foram utilizados e o último CO é que apenas artigos em inglês foram considerados. Após o processo de filtragem utilizando os COs, 556 foram baixados e analisados individualmente conforme os seguintes Critérios de Inclusão (CI) e Exclusão (CE):

- CI1: incluir apenas artigos que tenha como objetivo descrever uma estratégia (i.e., algoritmo, *framework*, método, modelo, etc.) para recomendar.
- CE1: excluir artigos que não utilizem o tempo para recomendar ou não expliquem detalhadamente como o tempo é utilizado.
- CE2: excluir artigos duplicados ou artigos diferentes relativos ao mesmo trabalho.

Após a última filtragem, 88 trabalhos fizeram parte do estudo e foram considerados para responder as questões de pesquisa. Entre os resultados do mapeamento sistemático desenvolvido, 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: *Restriction*, *Micro-profiles*, *Bias*, *Decay*, *Time Rating*, *Novelty* e *Sequence*. Essas categorias têm seus nomes originalmente criadas em inglês, porém para o *Decay* iremos utilizar Decaimento ou Recência no restante do texto. Essas categorias são descritas em detalhes nas Subseções 2.3.1-2.3.7.

Além disso, os resultados mostraram que as aplicações do Tempo mais comuns são através de *Restriction*, *Micro-profiles* e *Bias*. O formato do tempo mais utilizado é o contínuo e a dimensão do contexto mais utilizada em conjunto com o tempo é a localização. Foi observado também que em cada domínio de aplicação o Tempo costuma ser aplicado de forma diferente, e.g., na recomendação de Pontos de Interesse o uso de *Restriction* é mais comum, enquanto que na recomendação de Multimídia o Tempo é mais aplicado através dos *Micro-profiles*. Dentre os 88 artigos analisados, apenas quatro são da área educacional e utilizam o Decaimento (dois artigos) e *Restriction* (dois artigos).

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. Duas formas identificadas durante o mapeamento para se aplicar a restrição na recomendação são: (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. 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 assistir 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 os 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* (em português, Decaimento), o tempo é utilizado como um fator de decaimento na importância das interações do usuário, i.e., interações

(itens consumidos, avaliações, etc.) mais antigas têm 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 Decaimento, é apenas dado um peso menor para essas interações.

Nessa categoria a velocidade com que a importância de uma interação decai, também chamada de fator de decaimento, é uma variável do algoritmo que varia de acordo com o domínio de aplicação (de acordo com a necessidade de dar uma importância maior para os itens recentes para determinado sistema) ou até mesmo com o perfil do usuário (considerando que determinados usuários podem perder o interesse por itens antigos mais rapidamente que outros).

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 ao itens, sendo 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. Na área educacional, essa categoria pode ser aplicada em disciplinas de áreas tecnológicas de evolução rápida, onde itens mais atuais tendem a ser mais adequados e/ou corretos que itens antigos.

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 considera 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 o utilizando ou que os itens serão “convertidos” (nesse sentido, os autores se referem à conversão como a aceitar a recomendação dada e consumir o item recomendado). Os autores afirmam que percepção do usuário sobre a qualidade do SR é afetada tanto pela qualidade das recomendações em si, 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 do 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 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-se 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 usuá-

rio, 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

Para avaliar o desempenho dos algoritmos de recomendação, as métricas (quantitativas) tradicionalmente utilizadas são:

- Erro Absoluto Médio (do inglês *Mean Absolute Error*) e *Root Mean Square Error*: utilizadas para calcular o quão próximas as previsões do algoritmo de recomendação estão da realidade;
- Precisão (do inglês *Precision*): definida pela divisão do número de itens relevantes recomendados pelo número total de itens recomendados;
- *Recall*: definida pela divisão do número de itens relevantes recomendados pelo número total de itens relevantes existentes;
- Cobertura (do inglês *Coverage*): definida pela divisão da união de todas as listas de recomendação geradas (i.e., todos os itens distintos recomendados) pela quantidade de itens disponíveis no sistema. Também chamada de Cobertura do Catálogo (GE; DELGADO-BATTENFELD; JANNACH, 2010);
- *F-measure*: definida pela média harmônica entre Precisão e *Recall*.

A avaliação dos SRs é comumente realizada através de experimentos, comparando dois ou mais algoritmos de recomendação. A avaliação pode ter por objetivo medir as métricas quantitativas citadas anteriormente ou fazer uma avaliação pela perspectiva do usuário onde informações qualitativas são extraídas e analisadas. Os métodos de avaliação de SRs podem ser divididos 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 terá as notas dadas pelo usuário que serão repassadas ao algoritmo de recomendação como forma de construir o perfil dos usuários. A base de teste contém os itens para os quais o algoritmo de recomendação irá buscar prever o interesse

do usuário. As métricas apresentadas anteriormente são utilizadas para medir a eficiência do algoritmo, comparando a recomendação e/ou predições realizadas pelo algoritmo de recomendação com o resultado real presente na base de teste;

- Estudos com os usuários: um pequeno grupo de usuários é convidado a participar de um experimento e realiza tarefas específicas relacionadas ao SR. As métricas tradicionais de SR podem ser utilizadas em conjunto com medidas qualitativas para mensurar a satisfação dos usuários, por exemplo através da observação, questionários, entrevistas, etc.;
- Uso real do sistema: 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 através de ferramentas de *Web Analytics*, registros de *Logs*, notas dadas pelos usuários, etc.

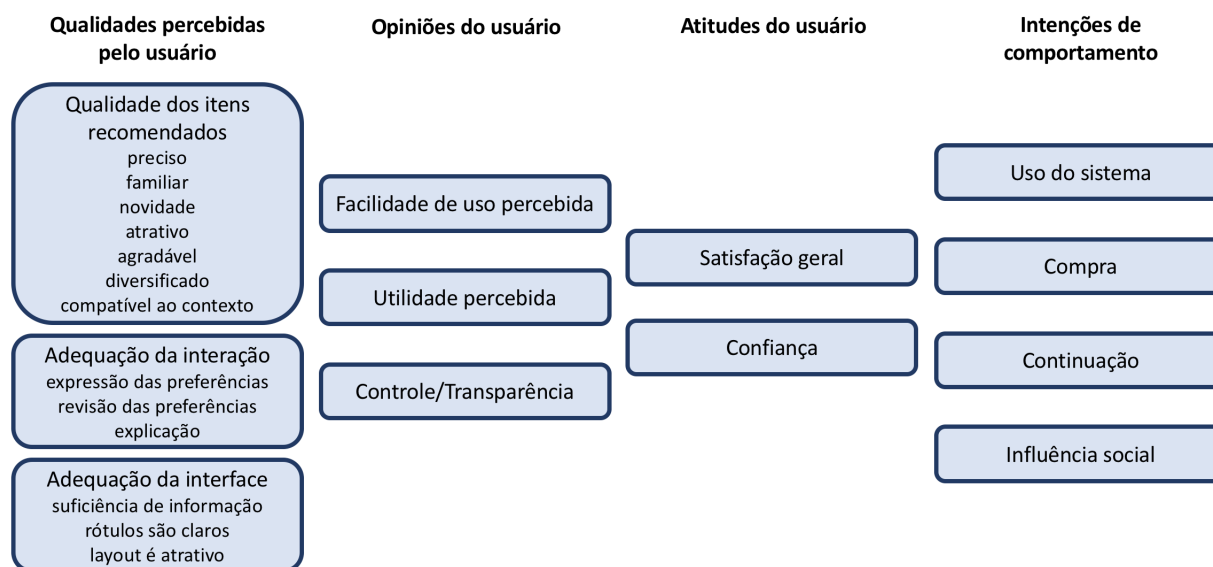
Pu, Chen e Hu (2011) propõem um *framework* para a avaliação de SRs utilizando **Estudos com os usuários**, com o objetivo de realizar uma avaliação do SR através da perspectiva do usuário. Esse *framework* se chama *Recommender Systems' Quality of User Experience* (ResQue) e foi proposto com base em 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 pelo Usuário, (2) Opiniões do Usuário, (3) Atitudes do Usuário, (4) Intenções de comportamento. 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 um 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).

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 paradig-

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



Fonte: Tradução de Pu, Chen e Hu (2011)

mas até as categorias de SRs Sensíveis ao Tempo identificadas 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 Decaimento. 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 têm 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, foi utilizado um Estudo com Usuários para a avaliação, utilizando um ambiente real de uso e medindo tanto as interações dos alunos com os SRs e a percepção do usuário sobre as recomendações recebidas. 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 quatro construtos do *framework*. Para ser utilizado, é 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.

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 que estão enquadrados dentro da categoria de Decaimento 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 a similaridade entre os serviços invocados pelos usuários com uma função de decaimento que considera que a QoS diminui com o passar do tempo em um algoritmo de Filtragem Colaborativa. 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: $t_{i,k}$ e $t_{j,k}$ são os tempos no qual os usuários u_i e u_j invocaram o serviço k ; 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.

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

Onde r_{u_i, s_k} é o QoS do serviço k na sua última invocação e $r_{u_i, s_k, t}$ representa o QoS considerando o decaimento desde a sua última invocação. 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 \bar{r}_{u_i} representa a média do QoS dos serviços invocados pelo usuário u_i , \bar{r}_{u_j} representa a média do QoS dos serviços invocados pelo usuários u_j e w_{u_i, u_j} é o 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.

Os autores realizaram 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 os autores realizaram 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 pelos quais 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 Resource 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 adequado para a recomendação se o contexto do usuário (dispositivo, ambiente) não for adequado para utilizar o recurso. Para isso, no contexto dos recursos são descritos quais os dispositivos nos quais 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 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 uma Precisão 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 con-

texto 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 seus interesses. 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.

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

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 , $r_{v,i}$ o grau de interesse do usuário u pelo item i , \bar{r}_v é a média das notas dadas pelo usuário v e $t_{v,i}$ é o tempo passado desde que o usuário v utilizou o item i . A função $f(t_{v,i})$ é uma 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.

$$f(t_{v,i}) = e^{-t_{v,i}} \quad (3.9)$$

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 $s_{i,j}(t)$ é a similaridade entre os itens i e j , U_i^t é o conjunto de usuários que avaliaram o item i no tempo t , U_j^t é o conjunto de usuários que avaliaram o item j no tempo t , a função f_{ui}^α é definida pelos autores como relevância temporal do item e pode ser vista na Equação 3.14 e r_{ui} e r_{uj} são as avaliações do usuário u para os

itens i e j .

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

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 Decaimento. 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 sozinha em três trabalhos e em mais dois outros trabalhos combinada com outras abordagens (i.e., 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 havendo 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 utiliza uma Constante de decaimento, pois é um fator que ainda requer estudos para verificar qual o valor é ideal e esse não é o objetivo desse trabalho. Na proposta deste trabalho é utilizada uma função de decaimento linear, por ser a função mais simples que poderia ser utilizada.

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
Hawalah 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.

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 *Offline*. Bencic e Bielikova (2012) realiza uma simulação para demonstrar o algoritmo de recomendação, que não se encaixa em nenhuma das formas de avaliação previstas por Shani e Gunawardana (2011). 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 Movielens¹. Com essa avaliação, apesar de ser realizada de forma correta e apresentar bons resultados, não é possível afirmar que o SR desenvolvido por Luo et al. (2010) teria um bom resultado em um ambiente educacional, porque os dados utilizados não são da área educacional. No presente trabalho foi realizada uma avaliação com participação de usuários em um ambiente educacional durante uma situação real de uso.

Na avaliação realizada no presente trabalho, o objetivo é comparar o algoritmo proposto 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 Base de dados com notas dadas por usuários a 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 Decaimento - 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 Decaimento (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 a posição 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, porque o algoritmo considera que os itens não acessados pelo usuário tendem a ser mais relevantes para ele do que os itens já acessados. O valor da relevância varia entre zero (menor relevância) e dois (maior relevância), e os itens com maior valor de F serão recomendados.

A função S de similaridade entre o usuário u e o item i pode ser calculada utilizando funções de similaridade como o Jaccard ou Cosseno (ver Seção 2.1.1). No experimento realizado nesse trabalho, é utilizado o Cosseno. A função de recência R ,

responsável pelo Decaimento, 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 pela ordem de 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 A, B e C com as seguintes posições: Item A - posição 2, Item B - posição 1 e Item C - posição 3. Supondo que o item candidato i a ser recomendado seja similar aos itens B e C, a Recência desse item, calculada utilizando a equação apresentada anteriormente, seria 1.0 como pode ser vista na Equação 4.3.

$$R(i) = \max\left(\frac{1}{3}, \frac{3}{3}\right) = 1.0 \quad (4.3)$$

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 à recomendação. Os itens candidatos à 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 realizada a busca da 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. Por fim, a função *nãoAcessado?* retorna 0 se o item já foi acessado e 1 caso contrário e esses valores são utilizados para o cálculo da Relevância para o item candidato, que é realizado utilizando a Equação 4.1.

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 a relevância dos itens candidatos a recomendação
/* Inicialização */
itens = buscarItensCandidatos();
itensAcessados = buscarItensAcessados(Usuário);
recênciaItensAcessados = calcularRecência(itensAcessados);
relevância = {};
/* Computação das recomendações */
foreach item ∈ itens do
    similaridade = calcularSimilaridade(itensAcessados, item);
    itensSimilares = buscarItensSimilares(itensAcessados, item);
    recência = calcularMaxRecência(itensSimilares,
        recênciaItensAcessados);
    relevância[item] = similaridade * recência + nãoAcessado?(item);
end
return relevância;

```

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.
recência = {};
quantidadeItensAcessados = contar(itensAcessados);
foreach itemAcessado ∈ itensAcessados do
    recência[itemAcessado] = itemAcessado.posição /
        quantidadeItensAcessados;
end
return recência;

```

Algoritmo 2: Pseudocódigo do cálculo da recência para cada item presente no perfil do usuário.

Na Figura 2 está um diagrama que mostra o funcionamento do SR proposto. As duas primeiras etapas envolvem a interação do usuário com a Aplicação, no caso um Ambiente Virtual de Aprendizagem (AVA), com o objetivo de permitir ao SR montar

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.

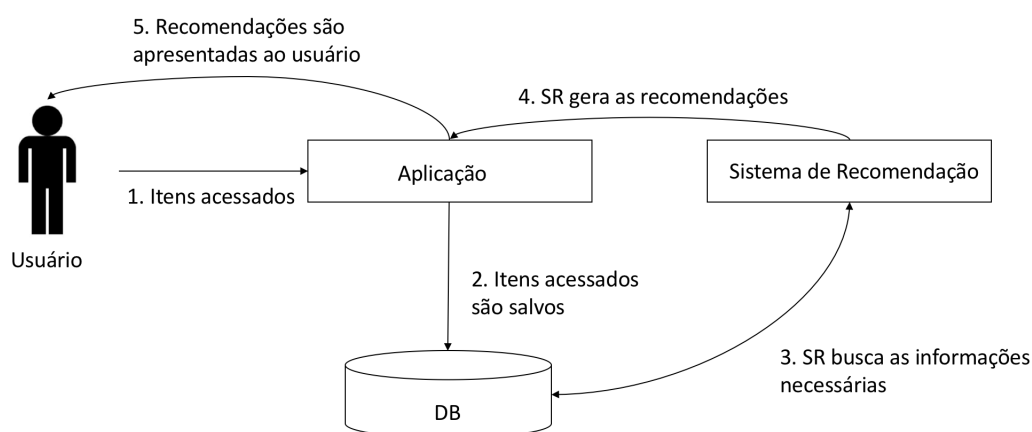
$\text{recênciaItensSimilares} = \text{recênciaItensAcessados.filtrar}(\text{itensSimilares});$

return $\max(\text{recênciaItensSimilares});$

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

o perfil para o usuário, identificando seus interesses e necessidades. A primeira etapa é a ação do usuário de acessar algum item dentro da aplicação. Na segunda etapa, a aplicação salva os itens acessados pelo usuário no Banco de Dados, de forma que essa informação possa ser utilizada pelo SR. Durante essa etapa, é calculada a posição do item acessado, a partir da quantidade de itens que já foram acessados pelo usuário até o momento, e salva juntamente com o restante das informações.

Figura 2 – Diagrama de metáfora de sistema



Fonte: O autor.

Na terceira etapa, o SR busca do Banco de Dados as informações necessárias para a recomendação, tanto sobre os itens acessados pelo usuário quanto sobre o conjunto de itens que podem ser recomendados. Sobre os itens acessados, o SR precisa das palavras-chave e a posição dos itens. Sobre os itens a serem recomendados, o SR precisa das palavras-chave e se aqueles itens já foram acessados pelo usuário ou não. Após buscar essas informações, o SR aplica o algoritmo mostrado anteriormente para gerar as recomendações, salva as recomendações geradas no Banco de Dados e retorna para a Aplicação. A última etapa é responsabilidade da Aplicação, que prepara as recomendações para serem visualizadas pelo usuário e as apresenta para ele.

4.2 CENÁRIO DE USO

Um cenário de uso do Decaimento 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 podem ser acessados 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 **Decaimento**, 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 é apresentado em mais detalhe o comportamento do algoritmo proposto no cenário de uso apresentado. Ao estudar o primeiro assunto da disciplina, Pedro acessa conteúdos relacionados a Pilhas e no seu perfil o SR formará o seu perfil com vários itens representados pela palavra-chave “Pilha”. O SR, ao gerar uma recomendação, irá encontrar um valor de similaridade alto entre o perfil do usuário e itens 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 todos os itens similares ao perfil do usuário terão a mesma recência. Na Tabela 2 é possível observar de forma resumida o comportamento do algoritmo (estamos considerando que nenhum item candidato a recomendação tenha sido acessado).

Como é possível ver na Tabela 2, o item relacionado a pilhas terá uma maior Relevância e será o item recomendado. 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 devido à recência desse tópico para o perfil do usuário. Na Tabela 3 é possível observar o comportamento do algoritmo nesse

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

Funções	Item relacionado a pilhas	Item relacionado a filas	Item relacionado a listas	Item relacionado 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.

caso.

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

Funções	Item relacionado a pilhas	Item relacionado a filas	Item relacionado a listas	Item relacionado 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 ao item com a palavra-chave “Pilha”, o impacto da Recência no algoritmo fez com a Relevância do item relacionado a filas fosse maior e, dessa forma, esse item seja recomendado. 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.

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

Funções	Item relacionado a pilhas	Item relacionado a filas	Item relacionado a listas	Item relacionado 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.

Como pode ser observado na Tabela 4, o 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 acessados primeiramente, ao voltar a acessar esses itens a relevância do item relacionado a pilhas voltará a ser maior que o restante.

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 têm uma vantagem em relação a outros SRs Sensíveis ao Contexto porque a informação temporal é mais simples de capturar, manipular e processar 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 Decaimento 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. Por outro lado, ao aplicar o Decaimento, 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, quando o usuário 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 o 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 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. Vale ressaltar que não foram encontrados nos trabalhos relacionados referências que mostrem como 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.

O algoritmo proposto utiliza a abordagem Baseada em Conteúdo em conjunto com o uso do tempo através da categoria Decaimento, ou seja, é dado um peso menor na recomendação para os itens acessados há 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 itens recomendados já foram ou não acessados. O Capítulo 5 apresenta o experimento no qual o algoritmo proposto foi avaliado.

5 EXPERIMENTO

Neste capítulo é apresentado o experimento realizado para a avaliação da proposta apresentada no Capítulo 4. Para avaliar a proposta deste trabalho foi utilizada 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 foi incorporado ao ambiente AdaptWeb[®] e avaliado em uma situação real de uso em um Minicurso de Algoritmos desenvolvido por Santos (2017). Na Seção 5.1 é descrito o ambiente AdaptWeb[®] no qual a proposta foi incorporada, as mudanças realizadas no ambiente para o experimento, o objetivo do experimento e o teste piloto realizado. A Seção 5.2 descreve o experimento que foi realizado e a Seção 5.3 apresenta as análises realizadas no uso do Sistema de Recomendação (SR) e no questionário aplicado.

5.1 PLANEJAMENTO

Essa seção explica em mais detalhes o ambiente AdaptWeb[®], o Minicurso de Algoritmos e Linguagem de Programação utilizado para o experimento, as mudanças propostas para o ambiente para a realização do experimento e os detalhes do experimento realizado, tais como as hipóteses do experimento e os critérios para divisão dos alunos em dois grupos.

5.1.1 Descrição do Ambiente AdaptWeb[®]

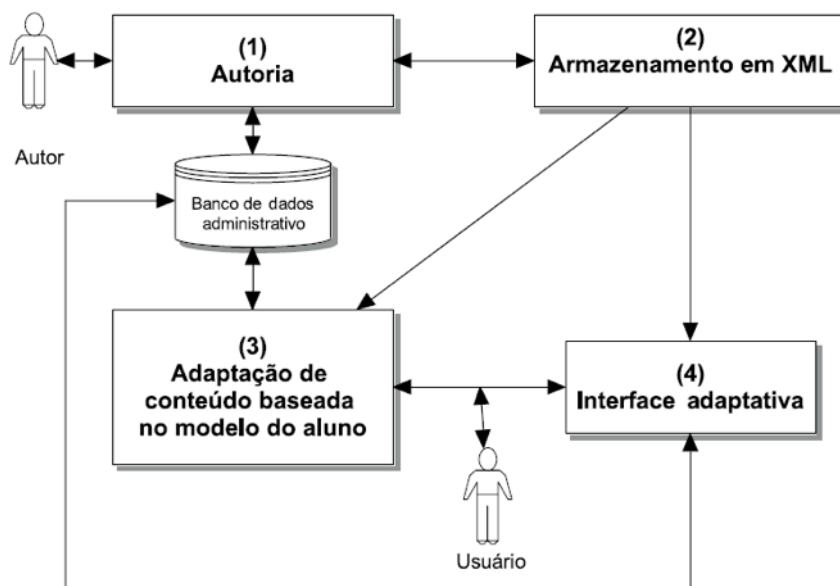
O AdaptWeb[®] (Ambiente de Ensino-Aprendizagem Adaptativo na Web) é um sistema *open source* que consiste em um AVA capaz de adaptar o conteúdo, a apresentação e a navegação em determinado curso às características e preferências do aluno (GASPARINI et al., 2009). A Seção 5.1.1.1 apresenta a Estrutura Geral do AdaptWeb[®].

5.1.1.1 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 3.

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,

Figura 3 – Estrutura do AdaptWeb®



Fonte: GASPARINI (2003)

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 proposta 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 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).

5.1.1.2 Sistema de Recomendação no AdaptWeb®

Na Seção 5.1.1.1 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.

Fazendo uma relação da estrutura do AdaptWeb® com o algoritmo proposto no Capítulo 4, os itens das categorias Conceito, Materiais Complementares e os próprios Links de Apoio são considerados 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 são utilizados. Isto porque os itens das outras categorias são estruturados pelo professor para serem acessados em determinada ordem e estão fortemente relacionados à conceitos específicos, enquanto os Links de Apoio são conteúdos extras da disciplina e podem ser acessados a qualquer momento.

Como no ambiente AdaptWeb® as palavras-chave para o itens podem ser cadastradas pelo professor da disciplina, não será utilizada nenhuma técnica para captura automática das palavras-chave. Para a representação dos materiais envolvidos no processo de recomendação (i.e., Conceitos, Materiais Complementares e Links de Apoio) foi criado um Dicionário de Palavras-Chave com as possíveis palavras a ser utilizadas. Esse Dicionário foi essencial para o funcionamento do SR, pois garante que as palavras-chave presentes nos materiais que são similares também serão similares. Isso também evita a necessidade de lidar com palavras-chave sinônimas ou a variação de singular e plural, já que as palavras-chave que podem ser utilizadas para representar os materiais são restritas às presentes no dicionário.

O Dicionário completo pode ser visto na Tabela 5. A criação do Dicionário foi realizada em conjunto com um aluno do Mestrado em Ensino de Ciências, Matemática e Tecnologias que é professor da disciplina de Algoritmos no SENAC-SC. O professor teve acesso ao Minicurso de Algoritmos utilizado no experimento e teve o papel de avaliar a pertinência das palavras-chave identificadas na primeira versão do Dicionário e identificar mais palavras para agregar o conjunto de palavras-chave.

Depois de finalizado o processo de criação do Dicionário, foi realizada a associação de forma manual entre cada material dos Conceitos, Materiais Complementares e Links de Apoio com as palavras-chave do Dicionário. No total, 51 Conceitos, 28 Materiais Complementares e 108 Links de Apoio foram analisados. O resultado dessa associação pode ser visto no Apêndice A.

O SR irá buscar, com base nos itens acessados pelo aluno, os Links de Apoio

Tabela 5 – Dicionário de palavras-chave

#	Palavras-chave	#	Palavras-chave
1	Algoritmo	27	Narrativa
2	Aplicacao de algoritmo	28	Operadores
3	Atribuicao	29	Operadores aritmeticos
4	Blocos	30	Operadores logicos
5	Comentario	31	Operadores relacionais
6	Compilador	32	Para
7	Constantes	33	Portugol
8	Decisao composta	34	Precedencia de operadores
9	Decisao encadeada	35	Problema
10	Decisao simples	36	Processamento
11	Diagrama de Chapin	37	Programa de computador
12	Enquanto	38	Pseudocodigo
13	Entrada	39	Repita
14	Escolha	40	Saida
15	Estrutura condicional	41	Se
16	Estrutura de repeticao	42	Semantica
17	Estrutura sequencial	43	Sintaxe
18	Fluxograma	44	Tipo caracter
19	Instrucao	45	Tipo inteiro
20	Interpretador	46	Tipo logico
21	Linguagem C	47	Tipo real
22	Linguagem de maquina	48	Tipo texto
23	Linguagem de programacao	49	Tipos de algoritmo
24	Linguagem natural	50	Tipos de dados
25	Logica de programacao	51	Variaveis
26	Matriz	52	Vetor

Fonte: O autor.

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 Seção 5.1.1.3.

5.1.1.3 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 estiver na tela de Conceito, Exercícios, Exemplos ou Materiais Complementares. Na Figura 4 pode-se observar a tela inicial do ambiente do aluno, onde estão destacadas as seguintes áreas: (1) Menu de navegação pelos tópicos; (2) Categorias dos materiais dentro do ambiente; (3) Interface das recomendações; (4) Mapa da disciplina; (5) Ajuda. As 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.

Figura 4 – Tela do Ambiente de Aula com as Áreas em Destaque

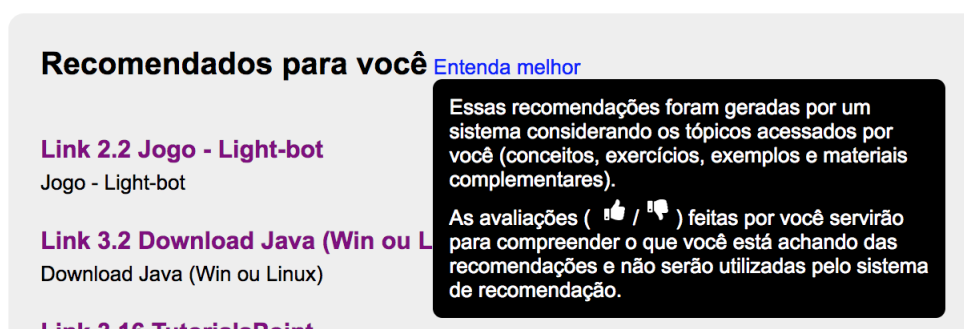


Fonte: O autor.

Como visto na imagem, as principais informações dos Links de Apoio apresentadas para o aluno são: Link, Nome do Link, Descrição e a possibilidade de avaliar o item positivamente ou negativamente. A avaliação feita pelo usuário não é considerada pelo algoritmo de recomendação, sendo que os itens acessados pelo usuário são considerados como do seu interesse. Como trabalho futuro é possível incorporar a avaliação na recomendação, e.g., não tornar a recomendar itens que foram avaliados com notas baixas ou apenas considerar para o algoritmo Baseado em Conteúdo os itens avaliados positivamente. 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), foi adicionado o 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 5 está a explicação das recomendações mostrada para o aluno.

Figura 5 – Explicação da recomendação



Fonte: O autor.

5.1.2 Definição do experimento

Para a execução do experimento, o SR proposto foi comparado a abordagem Baseada em Conteúdo tradicional utilizando uma estratégia *Between Subjects*, i.e., os alunos foram divididos em dois grupos e cada grupo utilizou apenas um dos sistemas. Para garantir que a única variável seja o SR utilizado, ambos os grupos utilizaram a mesma interface proposta para as recomendações.

O experimento proposto neste trabalho visa avaliar o desempenho e a percepção do usuário do SR proposto quando comparado à abordagem Baseada em Conteúdo Tradicional. Para avaliar o desempenho do algoritmo de recomendação em relação à variável independente *Algoritmo de Recomendação* foram adotadas as seguintes hipóteses:

- **H₀**: Não há diferenças entre o desempenho da abordagem Baseada em Conteúdo Tradicional e a proposta desse trabalho.
- **H₁**: Há diferenças entre o desempenho da abordagem Baseada em Conteúdo Tradicional e a proposta desse trabalho.

Para avaliar a percepção do usuário sobre a qualidades das recomendações recebidas em relação à variável independente *Algoritmo de Recomendação* foram adotadas as seguintes hipóteses:

- **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.

O experimento foi realizado através do Minicurso de Algoritmos e Linguagem de Programação, o qual teve seu design instrucional realizado por Santos (2017). Foram convidados a participar alunos dos cursos do Centro de Ciências Tecnológicas (CCT) da Universidade do Estado de Santa Catarina (UDESC) que possuem essa disciplina na grade curricular, i.e.: Bacharelado em Ciência da Computação, Tecnólogo em Análise e Desenvolvimento de Sistemas, Engenharia Elétrica, Engenharia Civil, Engenharia Mecânica, Engenharia da Produção, Matemática e Física. Os convites foram realizados diretamente nas salas das disciplinas de Algoritmos, Algoritmos e Linguagem de Programação, Linguagem de Programação e Iniciação a Ciência da Computação. Além disso, foi enviado um convite para todos os alunos do campus por *e-mail* através da Assessoria de Comunicação e foi divulgado na página do Facebook da UDESC Joinville.

Os usuários que se matricularam no Minicurso foram aleatoriamente divididos em dois grupos, da forma mais igualitária possível pelos seguintes critérios: Professor, Curso, Sexo e Idade. Isso foi possível porque durante o processo de matrícula os alunos responderam um questionário para montar o seu perfil. Também durante a matrícula os alunos tiveram acesso ao Termo de Consentimento Livre e Esclarecido, presente no Apêndice B, que explica o objetivo do experimento e no qual eles consentiram em participar e em permitir o uso dos resultados para essa pesquisa, garantindo a anonimidade dos participantes.

Ao final do minicurso, os alunos puderam acessar a avaliação do Minicurso, composta de 10 questões, e o questionário sobre a percepção do usuário sobre o SR. As questões do questionário foram selecionadas de um conjunto de questões definidas por Pu, Chen e Hu (2011) (presente no Anexo A), considerando um dos objetivos desse experimento que é medir a percepção do usuário sobre as recomendações recebidas. As questões selecionadas foram traduzidas para o Português e podem ser vistas a seguir:

1. Os itens recomendados corresponderam com os meus interesses;
2. Os itens recomendados para mim são diversificados (o sistema se preocupa em trazer itens diferentes a cada recomendação);

3. Os itens recomendados corresponderam aos interesses e necessidades que eu tinha no momento;
4. As recomendações são feitas no momento adequado;
5. O sistema de recomendação explica porque os links são recomendados para mim;
6. A informação apresentada na interface para os itens recomendados é suficiente para mim;
7. O layout do sistema de recomendação é atrativo e adequado;
8. Eu encontrei facilmente o local onde os itens são recomendados;
9. Eu percebi que o sistema de recomendação aprendia sobre minhas necessidades/preferências conforme eu avançava na disciplina;
10. É fácil encontrar um item para estudar com a ajuda do sistema de recomendação;
11. Eu me senti apoiado para encontrar itens do meu interesse com a ajuda do sistema de recomendação;
12. Eu entendi porque os itens foram recomendados para mim;
13. No geral, estou satisfeito com o sistema de recomendação.

Além dessas questões, foi adicionada uma questão discursiva perguntando sobre os pontos positivos e negativos do SR utilizado. Durante o desenvolvimento do Minicurso foram utilizadas as Intervenções definidas por Santos (2017), para fazer com que os alunos fiquem engajados no curso. As Intervenções são *e-mails* combinados com postagens no Fórum de Discussão que guiam os alunos no seu estudo e também propõem desafios para os estudantes. As Intervenções propostas por Santos (2017) consideram o Minicurso com uma duração de dois meses, por isso foi necessário uma adaptação dessas Intervenções para o período mais reduzido no qual foi realizado esse experimento. As Intervenções de Santos (2017), com as adaptações necessárias como data de início do experimento, podem ser vistas no Anexo B e os Desafios postados no Fórum de Discussão estão presentes no Anexo C.

5.1.3 Teste piloto

Antes do experimento ser realizado com os alunos no Minicurso de Algoritmos, foi realizado um teste piloto com quatro alunos que já realizaram essa disciplina. O objetivo do teste piloto foi avaliar os instrumentos do experimento, além de permitir

encontrar problemas na experiência do usuário para serem corrigidos antes da execução do minicurso. O teste piloto foi realizado no dia 06 de abril de 2018.

Durante o teste piloto os alunos foram divididos em dois grupos aleatoriamente, sendo que dois alunos utilizaram o SR Baseado em Conteúdo Tradicional e os outros dois utilizaram o SR com o Decaimento. Os alunos receberam um protocolo de atividades para realizar, presente no Apêndice C. As tarefas envolvem realizar a matrícula na disciplina, na qual eles leram e aceitaram o Termo de Consentimento Livre e Esclarecido, realizar o acesso a alguns conceitos e materiais complementares, utilizar o SR, realizar a avaliação da disciplina e responder ao questionário sobre a experiência com o SR. Durante o teste, os comentários e observações feitas pelos alunos foram anotadas para posterior análise.

Os quatro participantes do teste piloto foram identificados como Participante 1, Participante 2, Participante 3 e Participante 4. Os Participantes 1 e 2 utilizaram o algoritmo tradicional de recomendação, enquanto os participantes 3 e 4 utilizaram a proposta desse trabalho, porém eles não sabiam disso durante a realização do Teste. Os participantes foram livres na escolha do Sistema Operacional e Navegador utilizados, bem como no Modo de Navegação escolhido (Livre ou Tutorial). A Tabela 6 apresenta o Modo de Navegação e o Algoritmo de Recomendação utilizado por cada participante.

Tabela 6 – Dados dos Participantes do Teste Piloto

Participante	Modo de Navegação	Algoritmo de Recomendação
1	Tutorial	Baseado Em Conteúdo Tradicional
2	Livre	Baseado Em Conteúdo Tradicional
3	Tutorial	Baseado Em Conteúdo com Decaimento
4	Livre	Baseado Em Conteúdo com Decaimento

Fonte: O autor.

Durante a execução do Teste Piloto, os participantes encontraram erros de digitação no Termo de Consentimento Livre e Esclarecido (TCLE) e alguns problemas na prova da disciplina, como perguntas de Verdadeiro e Falso com questões duplicadas e uma pergunta que não possuía nenhuma resposta certa. Esses problemas foram todos resolvidos antes do início do experimento.

Sobre as recomendações, os Participantes 1 e 2 comentaram que as recomendações recebidas eram quase sempre as mesmas, mesmo tendo acessado conceitos diferentes - mudando apenas de ordem. Já os Participantes 3 e 4, que utilizaram a proposta desse trabalho, comentaram que ao acessar um novo conceito pelo menos 3 novos Links eram recomendados. Essa observação dos participantes demonstra o problema da Superespecialização presente na Abordagem Baseada em Conteúdo Tradicional, e mostra que a proposta desse trabalho utilizando o Decaimento ajuda a diminuir esse problema.

Os participantes também observaram que desde o primeiro acesso a disciplina eles receberam cinco links com recomendação, i.e., o número máximo possível. Isso mostra que, como não foi definido um limiar mínimo para a similaridade entre o perfil do usuário e os Links de Apoio, mesmo que a similaridade seja muito pequena o algoritmo sempre irá recomendar algo para o aluno. Por outro lado, não seria interessante adicionar um limiar para o experimento deste trabalho pois estaria adicionando mais uma variável independente ao experimento. Como trabalho futuro é possível analisar como o limiar mínimo para a similaridade pode afetar a qualidade percebida das recomendações.

Ao final do Teste Piloto foi realizada uma pequena entrevista com os participantes onde eles afirmaram ter entendido a interface do SR. Quando revelado a existência de dois SRs e que duas pessoas estavam usando a abordagem Tradicional e os outros dois estavam usando a proposta do trabalho, os participantes associaram essa informação com os comentários feitos sobre a Superespecialização, pelos alunos da abordagem Tradicional, e a novidade nas recomendações, pelos alunos da Proposta.

5.2 EXECUÇÃO

O período de matrícula para o minicurso foi de 09 de abril de 2018 até 13 de abril de 2018. Durante esse período, todas as turmas das disciplinas mencionadas na Seção 5.1.2 foram visitadas, convidando os alunos a se matricular no minicurso. O apoio dos professores das disciplinas foi essencial nessa etapa.

No total, 208 alunos se matricularam no Minicurso. Esses alunos foram homogeneamente divididos em dois grupos (AlgoritmoTradicional e AlgoritmoProposta) utilizando os seguintes critérios: Professor, Curso, Sexo e Idade. A Tabela 7 mostra o resultado da divisão dos alunos.

O período de execução do Minicurso foi de 16 de abril de 2018 até 10 de maio de 2018. Nesse período, dos 208 alunos matriculados no Minicurso, 145 acessaram a disciplina pelo menos uma vez, sendo 76 do grupo AlgoritmoTradicional e 69 do grupo AlgoritmoProposta.

O período para realizar a prova da disciplina e responder ao questionário foi de 11 de maio de 2018 até 14 de maio de 2018. Nesse período, dos 145 alunos que acessaram a disciplina pelo menos uma vez, 85 realizaram a prova final e responderam ao questionário, sendo 48 do grupo AlgoritmoTradicional e 37 do grupo AlgoritmoProposta. Dos 85 alunos que finalizaram a disciplina (i.e., realizaram a prova e responderam ao questionário), 47 acessaram pelo menos uma recomendação que recebeu, sendo 25 do grupo AlgoritmoTradicional e 22 do grupo AlgoritmoProposta. A Tabela 8

Tabela 7 – Divisão dos alunos de acordo com os critérios

			Algoritmo de Recomendação	
Critério		Total	Algoritmo Tradicional	Algoritmo Proposta
Professor	Professor A	12	6	6
	Professor B	18	9	9
	Professor C	7	4	3
	Professor D	5	3	2
	Professor E	38	19	19
	Professor F	4	2	2
	Professor G	10	5	5
	Professor H	6	3	3
	Professor I	6	3	3
	Professor J	26	13	13
	Professor K	14	7	7
	Outro	52	30	32
Curso	Computação	55	23	22
	TADS	36	18	18
	Elétrica	36	18	18
	Física	10	5	5
	Mecânica	31	15	16
	Química	3	2	1
	Produção	20	10	10
	Civil	15	7	8
	Matemática	12	6	6
Sexo	Masculino	135	67	68
	Feminino	72	36	36
	Não informado	1	1	0
Idade	Até 17 anos	21	10	11
	18 ou 19 anos	75	37	38
	20 ou 21 anos	35	18	17
	22 ou 23 anos	11	6	5
	24 ou 25 anos	26	13	13
	26 anos ou mais	40	20	20
	Total	208	104	104

apresenta mais informações sobre as ações dos alunos dentro do AdaptWeb®.

Pela Tabela 8 podemos perceber que 38 alunos que responderam ao questionário não acessaram nenhuma recomendação, sendo 23 do grupo Algoritmo Tradicional e 15 do grupo Algoritmo Proposta. Na Seção 5.3.2 os dados de acesso e avaliação das recomendações são analisadas mais profundamente, comparando os resultados dos dois algoritmos utilizando técnicas estatísticas descritas na Seção 5.3.1.

5.3 ANÁLISE DOS RESULTADOS

Nessa seção são descritas as análises estatísticas realizadas sobre os dados coletados durante a execução do experimento. As técnicas estatísticas utilizadas para fazer a análise são apresentadas na Seção 5.3.1 e nas Subseções subsequentes são apresentadas as análises realizadas sobre o desempenho dos algoritmos de recomendação e sobre a percepção dos alunos sobre a qualidade do SR utilizado.

Tabela 8 – Uso do Minicurso de Algoritmos e do SR

Quantidade de:	Tradicional	Proposta	Total
Alunos que se matricularam	104	104	208
Alunos que entraram pelo menos uma vez no curso	76	69	145
Alunos que acessaram pelo menos uma recomendação	46	39	85
Alunos que avaliaram pelo menos uma recomendação	30	22	52
Alunos que realizaram a prova	48	38	86
Alunos que responderam o questionário	48	37	85
Alunos que responderam o questionário e acessaram pelo menos uma recomendação	25	22	47
Alunos que responderam o questionário e avaliaram pelo menos uma recomendação	16	13	29
Acessos aos itens recomendados	227	396	623
Avaliações positivas aos itens recomendados	141	234	375
Avaliações negativas aos itens recomendados	5	4	9

5.3.1 Técnicas de Análise Estatística

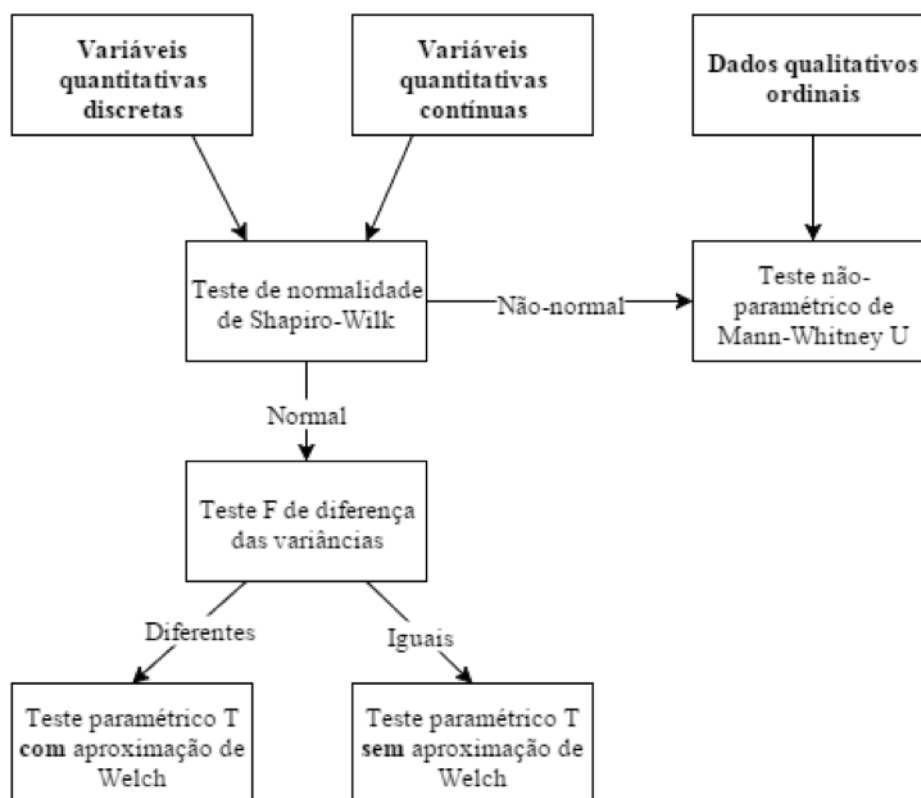
Para realizar as análises estatísticas, primeiro foi necessário entender os tipos de variáveis que podem ser analisadas. As variáveis são as informações coletadas durante o experimento e podem ser quantitativas ou qualitativas (BUSSAB; MORETTIN, 2012). As variáveis quantitativas são resultado de uma contagem e/ou mensuração (BUSSAB; MORETTIN, 2012) e podem ser discretas (conjunto finito ou enumerável de números) ou contínuas (pertencem a um intervalo de números reais). Exemplos de variáveis quantitativas são número de filhos, número de cômodos em uma casa, número de pessoas presentes em uma sala, altura e peso. As variáveis qualitativas são aquelas que descrevem um aspecto relacionado ao objeto estudado (BUSSAB; MORETTIN, 2012) e podem ser nominais (quando não possuem ordenação) ou ordinais (quando há ordem entre os valores). Exemplos de variáveis qualitativas são sexo, estado civil e grau de instrução.

A variável independente desse trabalho é o algoritmo de recomendação utilizado pelos alunos, que pode assumir os valores AlgoritmoTradicional ou Algoritmo-Proposta. As variáveis dependentes analisadas são o desempenho do algoritmo de recomendação e a percepção do usuário sobre as recomendações. O desempenho do SR foi medido através da quantidade de recomendações acessadas, quantidade de recomendações avaliadas, Precisão e Cobertura do algoritmo. A quantidade de recomendações acessadas e/ou avaliadas são variáveis quantitativas discretas, enquanto a Precisão e a Cobertura do algoritmo de recomendação são variáveis quantitativas contínuas. A percepção do usuário sobre as recomendações recebidas, que foi medida através do questionário, pode ser classificada como uma variável qualitativa ordinal, pois as respostas da escala de Likert podem ser ordenadas de Discordo Totalmente até Concordo Totalmente.

De acordo com o tipo de variável analisado e a distribuição dos valores dessa

variável é possível definir qual técnica estatística será utilizada. O fluxograma da Figura 6 produzido por Moissa (2016) sintetiza como decidir as técnicas estatísticas utilizadas para as análises. Todos os testes mencionados na Figura 6 estão disponíveis na ferramenta estatística R (R Core Team, 2018), que foi a ferramenta utilizada para todas as análises realizadas nas Subseções a seguir.

Figura 6 – Fluxograma de uso das técnicas estatísticas



Fonte: Moissa (2016)

5.3.2 Análise do Desempenho da Abordagem de Recomendação

Nessa seção é realizada uma análise estatística sobre os acessos e avaliações dos usuários aos Links de Apoio recomendados. O objetivo é extrair informações dos dados capturados durante a interação para avaliar o desempenho do algoritmo em relação a diversas métricas. As análises feitas foram da (1) Quantidade de recomendações acessadas, (2) Quantidade de avaliações positivas para as recomendações, (3) Quantidade de avaliações negativas para as recomendações, (4) Precisão do algoritmo de recomendação, (5) Cobertura do algoritmo de recomendação e (6) F-measure do algoritmo de recomendação, comparando os alunos que utilizaram a abordagem Baseada em Conteúdo Tradicional com os alunos que utilizaram a Proposta desse trabalho. Os resultados em detalhes das análises estatísticas realizadas nessa Seção estão presentes no Apêndice D.

Para a análise da quantidade de recomendações acessadas, foram considerados apenas os alunos que acessaram pelo menos uma recomendação no ambiente (i.e., 85 alunos). O grupo AlgoritmoTradicional teve média de 4,935 itens acessados por aluno e mediana de 4 itens, enquanto no grupo AlgoritmoProposta a média foi de 10,15 itens e a mediana de 5 itens. A análise utilizando o Teste de normalidade de Shapiro-Wilk mostrou que a distribuição dos dados não é normal (com $p - value < 2.2 \cdot 10^{-16}$) e portanto o Teste não-paramétrico de Mann-Whitney U foi utilizado para comparar os dois fatores. Esse Teste mostrou que não existe diferença significativa entre as amostras (com $p - value = 0.4957$) e portanto não podemos afirmar que existe diferença na quantidade de recomendações acessadas nos dois grupos analisados.

A análise realizada sobre as avaliações positivas para as recomendações utilizou apenas os alunos que avaliaram positivamente alguma recomendação (i.e., 52 alunos). O grupo AlgoritmoTradicional teve média de 4,7 avaliações positivas por aluno e mediana de 4 avaliações, enquanto o grupo AlgoritmoProposta teve média de 10,64 avaliações por aluno e mediana de 2,5 avaliações. Essa análise teve um resultado similar ao anterior, onde o Teste de normalidade de Shapiro-Wilk também mostrou que a distribuição é não-normal (com $p - value = 1.654 \cdot 10^{-13}$) e o Teste de Mann-Whitney U mostrou que não existem diferença significitiva entre as amostras dos dois grupos (com $p - value = 0.8275$).

A análise da quantidade de avaliações negativas para as recomendações utilizou apenas os alunos que avaliaram negativamente os itens (i.e., 7 alunos). O grupo AlgoritmoTradicional teve média e mediana de 1 avaliação negativa por aluno, enquanto o grupo AlgoritmoProposta teve média e mediana de 2 avaliações negativas por aluno. Ao realizar a análise estatística, o Teste de normalidade de Shapiro-Wilk mostrou que a distribuição dos dados é não-normal (com $p - value = 4.136 \cdot 10^{-06}$). O Teste não-paramétrico de Mann-Whitney U foi aplicado e mostrou que não existe diferença significativa para esse critério entre os dois grupos com ($p - value = 0.2059$).

A Precisão dos algoritmos de recomendação foi calculada para cada aluno que acessou pelo menos uma recomendação dentro do ambiente (i.e., 85 alunos) dividindo a quantidade de recomendações distintas acessadas pelo usuário pela quantidade de itens distintos que foram recomendados para o usuário durante todo o minicurso. Essa métrica resulta em um valor entre 0 e 1, onde um valor próximo de 0 indica que poucos dos itens recomendados foram acessados e um valor próximo de 1 mostra que a maioria dos itens recomendados foram acessados, independente da quantidade de vezes que um item foi acessado e da quantidade de vezes que o item foi recomendado. O grupo AlgoritmoTradicional teve média de 0,49679 e mediana de 0,42857, enquanto o AlgoritmoProposta teve média de 0,39859 e mediana de 0,31818. O Teste

de normalidade de Shapiro-Wilk mostra que a distribuição dos dados da Precisão é não-normal (com $p - value = 4.234 \cdot 10^{-06}$) e o Teste de Mann-Whitney U mostrou que não existem diferença significativa entre a precisão calculada para cada algoritmo (com $p - value = 0.2603$).

A Cobertura dos algoritmos da recomendação foi calculada considerando os alunos que acessaram pelo menos uma recomendação dentro do Minicurso (i.e., 85 alunos). Essa métrica foi calculada dividindo a quantidade de itens distintos recebidos como recomendação pelo usuário pela quantidade de itens disponíveis para recomendação na disciplina (i.e., 108 itens) e mostra a porcentagem dos itens disponíveis para recomendação que foram efetivamente recomendados. O grupo Algoritmo Tradicional teve média de 0,10064 e mediana de 0,09259, enquanto o Algoritmo Proposta teve média de 0,17450 e mediana de 0,14815 na cobertura. Ao realizar o Teste de normalidade de Shapiro-Wilk o resultado mostrou que a distribuição dos dados é não-normal (com $p - value = 1.421 \cdot 10^{-14}$). Quando aplicado o Teste de Mann-Whitney U o resultado mostrou que existe diferença significativa entre os resultados dos dois grupos com relação à Cobertura (com $p - value = 0.001031$) e, nesse caso, podemos afirmar que o algoritmo Com Decaimento tem uma Cobertura melhor que a algoritmo Tradicional. Essa diferença significativa mostra que a Proposta desse trabalho ajuda a diminuir o problema da Superespecialização presente na abordagem Tradicional.

Como visto na Seção 2.5, o *F-measure* é calculado como a média harmônica entre a Precisão e o *Recall* do algoritmo. Essa métrica é interessante para garantir que o algoritmo não diminua o *Recall* para aumentar a Precisão, e vice-versa. Como no experimento realizado o *Recall* não pode ser determinado, pois não é possível saber todos os itens relevantes existentes para cada usuário (essa métrica é mais comum em experimentos *offline*), nesse trabalho foi utilizado a Cobertura no cálculo do *F-measure* ao invés do *Recall*. Com isso podemos mostrar se o algoritmo Proposto teve uma melhor Cobertura que a abordagem Tradicional sem diminuir a sua Precisão. A análise foi realizada utilizando os 85 alunos que acessaram pelo menos uma recomendação. O grupo Algoritmo Tradicional teve média de 0,13852 nessa métrica e mediana de 0,12952, enquanto para o grupo Algoritmo Proposta a média foi de 0,21719 e mediana de 0,16393. O Teste de Normalidade de Shapiro-Wilk mostrou que a distribuição dos dados é não-normal (com $p - value = 2.931 \cdot 10^{-13}$) e portanto o Teste de Mann-Whitney U foi utilizado para verificar a diferença significativa entre os dois grupos. Esse teste mostrou que existe diferença significativa entre os dois grupos (com $p - value = 0.02082$) e portanto podemos afirmar que o *F-measure*, calculado com a Cobertura ao invés do *Recall*, teve um resultado melhor para o algoritmo proposto do que na abordagem tradicional. Isso mostra que o uso do Decaimento aumentou a Cobertura do algoritmo sem influenciar negativamente outros aspectos, como a Precisão.

Como a Cobertura e F-measure deram diferença significativa, podemos aceitar a hipótese alternativa definida que diz que “Há diferenças entre o desempenho da abordagem Baseada em Conteúdo Tradicional e a proposta desse trabalho” e a diferença significativa nos dois casos foi favorável a proposta desse trabalho.

5.3.3 Análise da Percepção do Usuário

A percepção dos usuários da qualidade das recomendações recebidas foi medida por meio do questionário presente na Seção 5.1.2. O questionário possui 13 questões que foram selecionadas do questionário definido por Pu, Chen e Hu (2011). Essas questões são afirmações nas quais os alunos devem se posicionar em uma escala de Likert de cinco pontos, de “Discordo Totalmente” até “Concordo Totalmente”. Também foi adicionada uma opção de “Não utilizei” para os alunos que não utilizaram o SR (ou usaram sem perceber). A Tabela 9 apresenta as respostas agrupadas dos alunos para cada uma das perguntas.

Tabela 9 – Respostas ao Questionário

Questão	Algoritmo	Discordo tot.	-	-	-	Concordo tot.	Não utilizei
1	Tradicional	2	1	2	15	11	17
	Proposta	0	0	6	11	7	13
2	Tradicional	1	0	5	18	6	18
	Proposta	0	1	7	9	9	11
3	Tradicional	2	0	8	13	8	17
	Proposta	0	2	4	12	8	11
4	Tradicional	1	0	5	17	6	19
	Proposta	0	1	8	8	10	10
5	Tradicional	3	0	4	10	10	21
	Proposta	0	2	9	8	6	12
6	Tradicional	3	3	3	12	8	19
	Proposta	0	0	5	11	10	11
7	Tradicional	2	5	4	14	9	14
	Proposta	1	1	7	9	7	12
8	Tradicional	0	6	3	12	9	18
	Proposta	0	2	8	8	6	13
9	Tradicional	2	2	7	10	7	20
	Proposta	0	0	11	5	8	13
10	Tradicional	1	0	10	14	4	19
	Proposta	0	2	4	12	8	11
11	Tradicional	2	2	6	13	5	20
	Proposta	0	1	6	10	8	12
12	Tradicional	4	1	6	17	2	18
	Proposta	0	1	5	9	10	12
13	Tradicional	2	0	6	13	12	15
	Proposta	0	1	4	7	14	11

Através das respostas dos alunos presentes na Tabela 9, podemos perceber que o grupo Algoritmo Tradicional tem mais respostas de “Discordo Totalmente” do que o grupo Algoritmo Proposta, com 25 para o Algoritmo Tradicional e uma para o Algoritmo Proposta. Além disso, pode-se perceber que 12 questões tiveram pelo menos um

“Discordo Totalmente” para o grupo AlgoritmoTradicional e uma questão para o grupo AlgoritmoProposta. Com relação a resposta “Concordo Totalmente”, 97 respostas foram dadas pelos alunos do grupo AlgoritmoTradicional e 111 pelos alunos do grupo AlgoritmoProposta. Por outro lado, tiveram mais respostas “Discordo Parcialmente”, “Não Concordo e Nem Discordo” e “Concordo Parcialmente” para o grupo AlgoritmoTradicional do que para o grupo AlgoritmoProposta, com 267 para o primeiro e 217 para o segundo.

O resultado das respostas “Não utilizei” mostram que os alunos responderam que não utilizaram para apenas algumas perguntas e não para todas, sendo que para o grupo AlgoritmoTradicional o valor varia entre 17 e 21 alunos e para o grupo AlgoritmoProposta varia entre 10 e 13 alunos. O fato desse valor ter variado tanto não foi esperado inicialmente, mas pode-se afirmar que pelo menos 17 alunos do primeiro grupo e 10 alunos do segundo grupo não utilizaram o SR. Ao comparar essa informação com os dados de acesso dos alunos, que mostra que 23 alunos do grupo AlgoritmoTradicional e 15 alunos do grupo AlgoritmoProposta não acessaram nenhuma recomendação, é provável que apesar de não terem acessado nenhuma recomendação os alunos perceberam a existência do SR, analisaram as recomendações recebidas e com base nisso responderam às questões do questionário.

As respostas dadas ao questionário foram analisadas utilizando o Teste não-paramétrico de Mann-Whitney U para verificar se existe diferença estatisticamente significativa entre as respostas dos dois grupos (i.e., $p < 0.05$). As respostas de “Não utilizei” não foram consideradas nessa análise. O resultado das análises está presente no Apêndice E.

A análise mostrou que a questão 12 (i.e., “Eu entendi porque os itens foram recomendados para mim.”) apresentou diferença significativa entre as respostas dos dois grupos (com $p - value = 0.01513$), e a diferença foi favorável ao grupo AlgoritmoProposta. Para as outras questões não foi encontrada diferença significativa entre as respostas dos alunos dos dois grupos. A diferença existente na questão 12 mostra que os alunos do grupo AlgoritmoProposta perceberam melhor a relação entre os conteúdos que acessaram com os links que foram recomendados para eles do que o grupo AlgoritmoTradicional. Essa informação mostra que o uso do Decaimento na Abordagem Baseada em Conteúdo fez com que as recomendações geradas fossem relacionadas aos conteúdos acessados recentemente, enquanto a Abordagem Tradicional recomenda considerando todos os conteúdos acessados pelo aluno durante todo o Minicurso igualmente.

Como pelo menos uma questão do questionário deu diferença significativa, podemos aceitar a hipótese alternativa definida que diz que “Há diferenças na percepção do usuário da qualidade das recomendações recebidas utilizando a abordagem

Baseada em Conteúdo tradicional e a proposta desse trabalho” e a diferença significativa foi favorável a proposta desse trabalho.

5.3.4 Questão Discursiva sobre o SR

Juntamente com o questionário de satisfação, foi colocada um questão discursiva para alunos com a seguinte pergunta: “Você utilizou o Sistema de Recomendação? Se sim, cite os pontos positivos e negativos desse sistema.” Dos 85 alunos que responderam à essa questão, 60 disseram não ter utilizado ou ter utilizado muito pouco o SR e não citaram nenhum ponto positivo ou negativo, sendo 36 do grupo AlgoritmoTradicional e 24 do grupo AlgoritmoProposta. Dois alunos do grupo AbordagemTradicional disseram não ter utilizado o SR pois não perceberam a sua existência e um aluno desse mesmo grupo disse que não utilizou com tanta frequência pois não entendeu como este funcionava. Um aluno do grupo AlgoritmoProposta disse não ter utilizado o SR pois achou que o conteúdo do Minicurso de Algoritmos era bem completo e não viu necessidade de acessar as recomendações.

Os pontos positivos e negativos apontados pelos alunos que utilizaram o SR foram analisados utilizando a técnica de análise e interpretação de dados qualitativos como descrito por Creswell e Creswell (2014). O processo de análise e interpretação é composto pelos seguintes passos: (1) Organizar e preparar os dados para análise; (2) Ler todos os dados; (3) Codificar os dados; (4) Usar o processo de codificação para gerar descrição ou categorização dos dados; (5) Decidir como a descrição ou categorização será representada para a narrativa qualitativa; (6) Interpretar as descobertas ou resultados. No processo de interpretação, o autor ainda destaca que é possível citar passagens da narrativa para transmitir os resultados da análise.

O primeiro passo, de organização e preparação dos dados, foi realizado capturando as respostas dadas pelos alunos que disseram ter utilizado o SR e organizando em uma planilha de dados, separando as respostas pelo grupo ao qual o aluno fazia parte (AlgoritmoTradicional ou AlgoritmoProposta). Na sequência, todos dados foram lidos e vários códigos (categorias) foram propostos com base nas respostas dos alunos (Passos dois e três). A partir dessas categorias, o quarto passo foi ler as respostas dos alunos e contar quantas vezes cada uma das categorias propostas apareceram nas respostas dos alunos de cada grupo, também chamado de etiquetagem. A representação da categorização foi realizada utilizando a Tabela 10 (quinto passo).

A partir da Tabela 10 é possível realizar o último passo que é a análise e interpretação dos resultados. Pode-se perceber pontos positivos em comum nos dois grupos, como “Guiar os estudos” e “Recomendações adequadas ao conteúdo estudado”. Com relação a esses códigos, algumas respostas do grupo AlgoritmoTradicional foram “ajudou bastante pois foi me passando um conhecimento sequencial”, “atende às ne-

Tabela 10 – Categorização da questão discursiva do questionário

Algoritmo Tradicional		Frequência
Pontos positivos	Conteúdo das recomendações	1
	Layout simples e intuitivo	1
	Guiar os estudos	2
	Funciona corretamente	1
	Ajudou a aprofundar alguns conteúdos	2
	Recomendações adequadas ao conteúdo estudado	2
Pontos negativos	Recomendações repetidas	2
Algoritmo Proposta		Frequência
Pontos positivos	Encontrar materiais para estudar	2
	Recomendações adequadas ao conteúdo estudado	3
	Ajudou a engajar mais nos estudos	2
	Funciona corretamente	1
	Recomendações são boas	2
	Guiar os estudos	1
Pontos negativos	Recomendações repetidas	2
	Difícil de encontrar recomendações do meu interesse	1
	Possível encontrar materiais similares com uma busca no Google	1

cessidades, recomendando sempre o que convém” e “lhe coloca conteúdo condizente ao seu nível”. Já dos alunos do grupo AlgoritmoProposta foram “achei adequado ao andamento da disciplina”, “ajuda em uma melhor e mais eficiente construção do conhecimento” e “sempre trouxe sugestões apropriadas”.

Também é possível perceber que o ponto negativo “Recomendações repetidas” aparecem nos dois grupos. Uma resposta do grupo Algoritmo Tradicional com esse código foi “Se repete demais em muitos conceitos”, enquanto do grupo AlgoritmoProposta foi “talvez eles pudessem se renovar mais a cada conteúdo”. Por essas respostas podemos afirmar que, mesmo as respostas com esse código tendo aparecido nos dois grupos, o aluno do grupo que utilizou a abordagem Baseada em Conteúdo tradicional aparentou estar mais decepcionado com o SR do que o aluno do grupo que utilizou a abordagem com Decaimento. Sendo que a segunda resposta demonstrou mais uma sugestão de renovar mais o conteúdo do que uma crítica ao SR.

Dois pontos positivos que apareceram nas respostas dos alunos do grupo Algoritmo Tradicional não se referem diretamente ao SR, que são “Conteúdo das recomendações” e “Layout simples e intuitivo”. Essas respostas se referem ao conteúdo dos links recomendados, que o aluno achou interessante por serem artigos e que ele podia baixar esse conteúdo para estudar *offline*, e à interface do SR, que era a mesma para os dois grupos.

Os principais destaques do Algoritmo Tradicional foram nas respostas de “Guiar os estudos”, “Ajudou a aprofundar alguns conteúdos” e “Recomendações adequadas ao conteúdo estudado”, com um total de seis respostas. E o único ponto negativo citado pelo alunos desse grupo foi o ponto já mencionado anteriormente de muitas recomendações serem repetidas.

Os principais destaques do AlgoritmoProposta foram nas respostas de “Encontrar materiais para estudar”, “Recomendações adequadas ao conteúdo estudado”, “Recomendações são boas” e “Ajudou a engajar mais nos estudos”, com um total de nove respostas. Por outro lado, alguns alunos desse grupo também citaram alguns pontos negativos para o SR, que são a dificuldade de encontrar recomendações interessantes e que seria mais fácil procurar pela resposta de suas dúvidas no Google que tentar encontrar uma recomendação do SR para isso, além do ponto negativo em comum com o AlgoritmoTradicional de recomendações repetidas.

Em geral, pode-se observar que o AlgoritmoProposta teve um maior número de respostas com pontos positivos (11 para o AlgoritmoProposta e 9 para AlgoritmoTradicional). Além disso, foi notado uma diferença na forma como os alunos dos dois grupos relataram as recomendações repetidas que receberam, que demonstraram uma maior decepção pelos alunos do grupo AlgoritmoTradicional. Outro ponto relevante é o fato de dois alunos do grupo AlgoritmoTradicional não lembrarem, no momento de responder ao questionário, da existência do SR. Isso demonstra que, mesmo as recomendações estando nas páginas principais do ambiente do aluno (i.e., Conceitos e Materiais Complementares), eles provavelmente não ficaram interessados em nenhum momento pelos Links recomendados, já que a interface para os dois grupos era a mesma e o mesmo não apareceu nas respostas do grupo AlgoritmoProposta.

5.3.5 Limitações e Ameaças à Validade do Experimento

Nesta Seção são apresentadas as limitações do experimento realizado e as ameaças que podem ter influenciado o resultado do experimento e poderiam invalidá-lo. O tempo de execução do experimento (i.e., pouco menos de um mês) foi uma limitação do experimento, que não permitiu que os alunos tivessem mais tempo para interagir com o Minicurso e com o SR. Outra limitação foi o tema do Minicurso sobre Algoritmos e Linguagem de Programação, que é um tema fundamental dos cursos de exatas e com uma ementa rigorosamente definida, enquanto um curso sobre outro tema mais exploratório ou progressivo como Desenvolvimento *Web* ou Computação Quântica em que os alunos poderiam tirar maior proveito da ferramenta de recomendação. Apesar da quantidade de alunos que se matriculou no Minicurso ter sido considerável, o tamanho da amostra que efetivamente utilizou o Minicurso (145 alunos) e principalmente que acessou alguma recomendação gerada (85 alunos) é outra limitação do experimento, que poderia ter resultados mais conclusivos se mais alunos tivessem utilizado a ferramenta de recomendação.

A qualidade dos Links de Apoio cadastrados no ambiente AdaptWeb® é uma ameaça à validade do experimento, pois podem influenciar a percepção dos alunos

sobre o SR positiva ou negativamente. Além disso, os Links de Apoio foram encontrados por pessoas que não são professores da disciplina de Algoritmos e podem não ser totalmente adequados para alunos do Minicurso. A representação dos Links de Apoio e dos demais itens do Minicurso através do Dicionário de Palavras criado também podem ser uma ameaça à validade do resultado desse experimento, pois influenciaram diretamente nas recomendações geradas para os alunos.

O questionário sobre a percepção dos alunos sobre a qualidade das recomendações recebidas ser aplicado apenas ao final do experimento é outra ameaça à validade do experimento, pois muitos alunos podem não lembrar das recomendações recebidas ou do SR em geral após um longo período do seu uso. Esse pode ser um dos motivos de não ter sido capturado através do questionário o mesmo que foi observado no teste piloto, de que a proposta desse trabalho ajudaria a diminuir o problema da Superespecialização presente na abordagem Tradicional.

O interesse e engajamento dos alunos no Minicurso pode ter influenciado também o resultado do experimento, sendo que no momento da divisão dos alunos nos dois grupos não era possível prever essa variável interveniente e pode acontecer de alunos em um dos grupos terem um interesse e engajamento muito maior que os alunos do outro grupo.

Como último fator que pode ter influenciado o experimento e ser uma ameaça à validade do mesmo está a grande variância existente entre os valores das variáveis dependentes do experimento, e.g., a quantidade de recomendações acessadas que variou de 1 à 107 recomendações. Esse é um dos fatores pelo qual, apesar de a proposta desse trabalho ter uma média maior em muitas das variáveis medidas, a maioria delas não apresentou diferença significativa.

5.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

Neste capítulo foi descrito o ambiente no qual o Sistema de Recomendação (SR) proposto foi 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 categorias “Conceito”, “Materiais Complementares” e “Links de Apoio”. Os itens a serem recomendados são os Links de Apoio, visto que os itens das outras categorias são estruturados pelo professor para serem acessados em determinada ordem e estão fortemente relacionados à conceitos específicos.

Utilizando como base as diretrizes propostas por Pu, Chen e Hu (2012) foi pro-

posta 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 porquê daqueles itens serem recomendados. Essa interface foi utilizada tanto pelo grupo AlgoritmoTradicional quanto pelo AlgoritmoProposta.

O experimento realizado tem por objetivo avaliar a experiência do usuário com o SR proposto no Capítulo 4 (AlgoritmoProposta) em comparação à abordagem Baseada em Conteúdo tradicional (AlgoritmoTradicional). O experimento foi realizado no ambiente AdaptWeb[®] através de um Minicurso de Algoritmos e Linguagem de Programação desenvolvido no ambiente nos meses de abril e maio de 2018. Ao final do minicurso os alunos receberam um questionário para responder sobre a sua experiência ao interagir com o SR, que foi adaptado do conjunto de questões definidas por Pu, Chen e Hu (2011). No total, 208 alunos se matricularam no Minicurso, sendo que 85 alunos chegaram até o final e responderam ao questionário.

Foram analisados os dados de acesso e avaliações das recomendações pelos alunos e da geração de recomendações pelos dois algoritmos. Na análise da Cobertura e do F-measure o AlgoritmoProposta teve resultado estatisticamente melhor que o AlgoritmoTradicional. Isso mostra que a proposta desse trabalho ajudou a diminuir o problema da Superespecialização presente na Abordagem Baseada em Conteúdo Tradicional, aumentando a Cobertura do método sem impactar negativamente na Precisão e na Percepção do Usuário. Para as outras métricas, como quantidade de links acessados e precisão do algoritmo, não existe diferença significativa entre os resultados dos dois grupos.

Ao analisar o questionário utilizando-se de técnicas estatísticas descritas na Seção 5.3.1, o resultado foi que proposta desse trabalho apresentou melhor resultado na questão 12, na qual dizia “Eu entendi porque os itens foram recomendados para mim”. Isso mostra que os alunos puderam perceber uma melhor relação entre os conteúdos acessados com os itens recebidos como recomendação pelos alunos que utilizaram a proposta desse trabalho do que pelos alunos que utilizaram a abordagem Tradicional de recomendação. Sobre as outras 12 questões do questionário não é possível realizar nenhum tipo de afirmação pois não existe diferença significativa entre os resultados alcançados pelos dois grupos.

Na análise da questão discursiva do questionário, foi aplicado uma técnica de análise e interpretação de dados qualitativos para identificar os principais pontos positivos e negativos. Ao final do processo podemos afirmar que os dois algoritmos de recomendação tiveram algumas respostas em comum, tanto em pontos positivos (e.g., “Guiar o aluno”) como em pontos negativos (e.g., “Recomendações repetidas”). Porém, a proposta desse trabalho teve mais pontos positivos apontados, com pontos

que indicaram a importância do SR para esses alunos, como “Ajudou a engajar mais nos estudos” e “Recomendações são boas”.

Como resultado final do experimento podemos aceitar as duas hipóteses alternativas, confirmando que existe diferença significativa tanto no desempenho do algoritmo de recomendação quanto na percepção do usuário sobre as recomendações recebidas entre o Algoritmo Tradicional e o Algoritmo Proposta. Parece essas duas hipóteses, o resultado foi favorável a proposta desse trabalho e portanto pode-se afirmar que a aplicação do Decaimento teve um melhor resultado nos dois aspectos medidos em relação a Abordagem Baseada em Conteúdo Tradicional.

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 da situação do usuário, como considerar as informações contextuais do aluno. Com o propósito de investigar sobre esse tema, 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, devido a quantidade reduzida de trabalhos nesse tema aplicados à essa área.

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 analisar se a aplicação do contexto temporal, através do decaimento, em um sistema de recomendação voltado a AVAs influencia o desempenho do algoritmo de recomendação e a percepção do aluno sobre as recomendações recebidas. Dentre as categorias de SR Sensíveis ao Tempo presentes na Seção 2.3, a proposta desse trabalho se encaixa no Decaimento.

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 à sequência de itens aces-

sados 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.

Para alcançar o objetivo do trabalho e responder a questão de pesquisa, um experimento controlado foi realizado em um ambiente real de uso comparando a proposta do trabalho com a abordagem Baseada em Conteúdo Tradicional. O experimento aconteceu dentro do ambiente AdaptWeb® durante um Minicurso de Algoritmos e Linguagem de Programação no período de 16 de Abril de 2018 à 10 de Maio de 2018. Durante a execução do experimento, dados de acesso e avaliação das recomendações pelos alunos foram capturadas para analisar o desempenho do algoritmo proposto. Ao final do experimento os alunos responderam ao questionário presente na Seção 5.1.2, com questões selecionadas do questionário proposto por Pu, Chen e Hu (2011) para avaliar a percepção do usuário sobre as recomendações recebidas.

Os dados coletados durante o experimento foram analisados utilizando técnicas estatísticas para verificar se existe diferença significativa entre o grupo que utilizou a Abordagem Baseada em Conteúdo Tradicional (AlgoritmoTradicional) e o grupo que utilizou a Proposta desse trabalho que considera o Decaimento no interesse nos alunos pelos itens acessados anteriormente (AlgoritmoProposta). Os resultados do experimento presentes na Seção 5.3 mostraram que o grupo AlgoritmoProposta apresentou resultados estatisticamente melhores que o AlgoritmoTradicional em relação à Cobertura e ao F-measure do algoritmo (métricas do desempenho do algoritmo) e em relação à uma das perguntas do questionário aplicado (que visava medir a percepção do usuário). Em relação às outras métricas de desempenho e às outras questões do questionário, não foi encontrado diferença significativa entre o resultado para os dois grupos. Mesmo assim, foi possível aceitar as duas hipóteses alternativas definidas que afirmam que existe diferença significativa no desempenho do algoritmo e na percepção do usuário entre os dois algoritmos comparados, com resultado favorável a proposta desse trabalho. Com esses resultados, pode-se responder à questão de pesquisa desse trabalho afirmando que considerar o Decaimento do interesse do aluno em relação aos itens acessados anteriormente influencia positivamente tanto o desempenho do algoritmo quanto a percepção do usuário sobre isso.

Os principais resultados desse trabalho foram:

- A categorização dos SRs Sensíveis ao Tempo em sete formas distintas de utilizar o tempo em um SR e que representam os estudos na área, independente do domínio de aplicação.

- A concepção de um algoritmo que utiliza a categoria do Decaimento para o Contexto Educacional.
- A avaliação da proposta do trabalho utilizando Decaimento em um experimento com usuários reais em um Ambiente Educacional, comparando a proposta com uma das Abordagens Tradicionais de Recomendação através do desempenho do algoritmo e da percepção dos usuários sobre as recomendações. Esse tipo de avaliação não é comumente realizada na área, como pode ser visto nos trabalhos relacionados.

Como outros resultados 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 Conference 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.
- BORBA, E. J.; GASPARINI, I.; LICHTNOW, D. Describing Scenarios and Architectures for Time-Aware Recommender Systems for Learning. In: Enterprise Information Systems, Springer International Publishing AG, part of Springer Nature, no. 17, 2018. no prelo.

Como trabalhos futuros, propõe-se o estudo mais aprofundado com relação a um limiar mínimo de similaridade entre os itens acessados pelo alunos e os itens candidatos a recomendação para que a recomendação efetivamente aconteça. Ao conseguir definir esse limiar, através de experimentos em um AVA, será possível garantir que apenas recomendações realmente relevantes para o aluno sejam recomendadas para ele e que itens que possam desmotivar ou tirar a confiança do aluno no SR não sejam recomendados.

Outro aspecto a se considerar é estudar a influência de um fator de decaimento no algoritmo de recomendação. O fator de decaimento é uma constante, geralmente no intervalo de 0 a 1, que foi evitado nesse trabalho através do algoritmo proposto porém muito utilizado nos trabalhos relacionados presentes no Capítulo 3.

Esse fator de decaimento pode ser personalizado para cada aluno ou igual para todos e vai influenciar a velocidade com que o algoritmo considera que o interesse do aluno por um determinado item acessado diminui.

Aplicar o conceito de decaimento na criação do perfil de um usuário em outros sistemas, de recomendação ou não, é outro possível trabalho futuro. Um exemplo seria na seleção de especialidades de um pesquisador de acordo com o seu currículo Lattes. Geralmente durante a vida de um pesquisador este acaba passando por diversas áreas de interesse e os assuntos pesquisados no início de sua carreira não necessariamente representam sua área de interesse atual. Ou seja, apesar de ainda ser um assunto do seu domínio, é provável que o pesquisador não esteja mais interessado em pesquisas desse tema e muitas vezes nem esteja mais atualizado com as pesquisas recentes da área. Esse tipo de perfil poderia ser aplicado para auxiliar na seleção de pessoas para revisar um artigo (e.g., em revistas ou eventos) ou projeto e na recomendação de palestras para comparecer em um evento científico.

Além disso, analisar as outras categorias de SRs Sensíveis ao Tempo e propor algoritmos para aplicação na área educacional, combinando com o Decaimento ou não, é outra proposta de trabalho futuro.

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.
- BUSSAB, W. O.; MORETTIN, P. A. **Estatística básica**. São Paulo: Saraiva, 2012. v. 7.
- 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.
- CRESWELL, J. W.; CRESWELL, J. D. **Research design: Qualitative, quantitative, and mixed methods approaches**. [S.l.]: Sage publications, 2014.
- 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.

GE, M.; DELGADO-BATTENFELD, C.; JANNACH, D. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In: ACM. **Proceedings of the fourth ACM conference on Recommender systems**. [S.l.], 2010. p. 257–260.

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.

MOISSA, B. **A INFLUÊNCIA DE FERRAMENTAS DE LEARNING ANALYTICS NA INTERAÇÃO, DESEMPENHO E SATISFAÇÃO DOS ALUNOS**. Dissertação (Mestrado) — Mestrado em Computação Aplicada, Centro de Ciências Tecnológicas, UDESC, Joinville., 2016.

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.

R Core Team. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2018. Disponível em: <<https://www.R-project.org/>>.

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.

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.; OPPERMANN, R. An operational definition of context. **Context**, Springer, v. 7, p. 558–571, 2007.

Apêndices

APÊNDICE A – PALAVRAS-CHAVE DOS MATERIAIS NO MINICURSO DE ALGORITMOS E LINGUAGEM DE PROGRAMAÇÃO

A.1 CONCEITOS

Conceito	Nome	Palavras-chave
1	Seja bem-vindo ao minicurso de Algoritmos e Programação!	Algoritmo
1.1	Objetivo do minicurso	Algoritmo
1.2	Ferramentas de aprendizagem	Algoritmo
1.3	O que você vai ver nesse curso	Algoritmo
1.4	Como estudar	Algoritmo
2	Da lógica à programação	Logica de programacao, Instrucao, Algoritmo, Programa de computador
2.1	Por que aprender a programar?	Logica de programacao, Problema
2.2	Aplicação de Algoritmos	Logica de programacao, Aplicacao de algoritmo
2.3	Algoritmos e Programação	Programa de computador, Instrucao, Linguagem de programacao
3	Interpretadores e compiladores	Interpretador, Compilador, Linguagem de maquina
3.1	Interpretadores de algoritmos (Portugol IDE)	Interpretador, Portugol
3.2	Compiladores de linguagem	Compilador, Linguagem C
3.3	Outras ferramentas de aprendizagem	Sintaxe, Semantica
4	Algoritmos	Algoritmo, Instrucao, Entrada, Processamento, Saida
4.1	Criando um algoritmo	Problema, Algoritmo, Entrada, Processamento, Saida
4.2	Tipos de algoritmo	Tipos de algoritmo, Narrativa, Fluxograma, Pseudocodigo, Diagrama de Chapin
5	Sobre narrativas	Narrativa, Logica de programacao, Linguagem natural

5.1	Tipos de narrativa	Narrativa, Estrutura sequencial, Estrutura condicional, Estrutura de repeticao
5.2	Exercitando as narrativas	Narrativa, Algoritmo, Estrutura condicional
6	Sobre pseudocódigos	Pseudocodigo, Linguagem natural, Linguagem de programacao
6.1	Estrutura de um pseudocódigo	Pseudocodigo, Comentario, Variaveis, Semantica
6.2	Predefinições para escrita de pseudocódigos	Pseudocodigo, Sintaxe, Variaveis, Constantes, Tipos de dados, Tipo inteiro, Tipo real, Tipo caracter, Tipo texto, Tipo logico
7	O que é um programa?	Programa de computador, Linguagem de programacao, Linguagem de maquina
7.1	Estrutura de um programa	Programa de computador, Sintaxe, Semantica, Instrução, Estrutura sequencial, Estrutura condicional, Estrutura de repeticao
7.2	Linguagem C - Principais conceitos	Linguagem C, Compilador
7.2.1	main()	Linguagem C
7.2.2	system()	Linguagem C
7.2.3	#include	Linguagem C, Compilador
7.3	Constantes e comandos de atribuição	Constantes, Variaveis, Atribuicao, Tipo inteiro, Tipo caracter, Tipo texto
7.4	Comandos de Entrada-Saída	Saida, Entrada, Tipo texto
7.4.1	printf	Linguagem C, Saida
7.4.2	scanf	Linguagem C, Entrada, Tipo inteiro, Tipo real, Tipo texto, Tipo caracter
7.5	Strings	Linguagem C, Tipo alfanumerico, Tipo caracter, Operadores, Atribuicao
8	Dados, variáveis e operadores	Variaveis, Operadores, Precedencia de operadores

8.1	Dados	Tipos de dados, Tipo inteiro, Tipo real, Tipo caracter, Tipo texto, Tipo logico
8.2	Operadores básicos	Operadores, Operadores aritmeticos
8.3	Operadores aritméticos em C	Operadores, Operadores aritmeticos, Linguagem C, Precedencia de operadores
8.4	Operadores relacionais	Operadores, Operadores relacionais, Precedencia de operadores
8.5	Operadores lógicos	Operadores, Operadores logicos, Precedencia de operadores
8.6	Precedência entre todos os Operadores	Operadores, Precedencia de operadores
9	Estrutura de Controle - Decisão	Estrutura condicional
9.1	Estrutura de decisão simples	Estrutura condicional, Decisao simples, Se
9.2	Estrutura de decisão composta	Estrutura condicional, Decisao composta, Se
9.3	Estrutura de decisão encadeada	Estrutura condicional, Decisao encadeada, Escolha
10	Estrutura de controle – Laços (repetição)	Estrutura de repeticao
10.1	Laço while (enquanto)	Estrutura de repeticao, Enquanto
10.2	Laço do...while (repita)	Estrutura de repeticao, Repita
10.3	Laço para (for)	Estrutura de repeticao, Para
11	Vetores e matrizes	Variaveis, Matriz, Vetor
11.1	Vetores	Variaveis, Vetor
11.2	Matrizes	Variaveis, Matriz

Tabela 11 – Palavras-chave associadas aos Conceitos

A.2 MATERIAIS COMPLEMENTARES

Conceito relacionado	Nome	Palavras-chave
1	Seja Bem-Vindo	Algoritmo
1	Créditos	Algoritmo

2	Podcast do Papo BJPnet	Programa de computador, Linguagem de programacao, Linguagem de maquina
2	Sugestão - Jogo Light-bot	Logica de programacao, Problema
3.3	Sugestão - Outros compiladores de linguagem C	Linguagem C, Compilador
3.3	Outros interpretadores de pseudo-linguagem	Portugol, Interpretador
4.1	Passos para criação de um algoritmo	Problema, Algoritmo, Entrada, Processamento, Saida
5.1	Narrativa Sequencial	Narrativa, Estrutura sequencial
5.1	Narrativa de Seleção	Narrativa, Estrutura condicional
5.1	Narrativa de repetição	Narrativa, Estrutura de repeticao
7.1	Pseudocódigo: Alô mundo	Portugol
7.1	Linguagem C: Alô mundo	Linguagem C
7.1	Comparativo das estruturas básicas de pseudocódigo e C	Portugol, Linguagem C
7.3	Comando de atribuição em pseudocódigos	Portugol, Atribuicao
7.4	Comparativo Pseudocódigo x Linguagem C - Comandos de Saída	Saida, Portugol, Linguagem C
7.4	Tabela de códigos especiais em C	Linguagem C, Entrada, Saida, Tipo texto
7.4	Tabela códigos de formatação em C	Linguagem C, Entrada, Saida, Tipo texto
7.4.2	Pseudocódigo - Função de entrada	Portugol, Entrada
7.4.2	Função de entrada	Linguagem C, Entrada
7.4.2	Pseudocódigo - Entrada de dados	Portugol, Entrada
7.4.2	Entrada de dados	Linguagem C, Entrada
7.5	Tabela das funções de String	Linguagem C, Tipo texto
8.1	Tabela - Pseudo x C - Tipos de dados	Portugol, Linguagem C, Tipos de dados, Tipo inteiro, Tipo real, Tipo caracter, Tipo logico, Tipo texto
8.1	Pseudocódigo - Atribuições com variáveis	Portugol, Variaveis, Atribuicao, Matriz, Tipo inteiro, Tipo texto, Tipo logico, Tipo real

8.1	Atribuições com variáveis	Linguagem C, Variaveis, Atribuicao, Matriz, Tipo inteiro, Tipo texto, Tipo logico, Tipo real
8.4	Pseudo x C - Tabela de operadores relacionais	Linguagem C, Operadores relacionais
11.1	Declaracao de vetores	Linguagem C, Vetor, Atribuicao, Entrada
11.2	Declaração de matrizes	Linguagem C, Matriz, Atribuicao

Tabela 12 – Palavras-chave associadas aos Materiais Complementares

A.3 LINKS DE APOIO

#	Link de Apoio	Palavras-chave associadas
1	Lógica de Programação - Fluxograma e Portugol	Logica de programacao, Fluxograma, Portugol, Pseudocodigo, Operadores, Blocos, Tipos de dados, Variaveis,
2	Maçãs e laranjas: diferenças entre compilador e interpretador	Compilador, Interpretador
3	Por que aprender a programar?	Aplicacao de algoritmo, Programa de computador
4	Algoritmo: Definição e introdução	Algoritmo, Linguagem de maquina, Compilador, Linguagem de programacao
5	Os Benefícios e o Porquê de Aprender a Programar.	Aplicacao de algoritmo, Algoritmo, Instrucao, Programa de computador
6	Vetores e Matrizes	Vetor, Matriz, Contantes, Entrada, Saida
7	VisuALG - estrutura de repetição	Estrutura de repetição, Enquanto, Para, Repita
8	Introdução ao VisuALG	Portugol, Entrada, Saida, Operadores aritmeticos, Precedencia de operadores, Operadores logicos, Estrutura condicional
9	VisuALG – Arrays e Strings	Portugol, Vetor, Tipo texto
10	A Linguagem de Programação do VisuAlg	Portugol, Pseudocodigo, Tipos de dados, Variaveis, Constantes

11	Algoritmo: Estrutura de vetores e matrizes.	Portugol, Entrada, Saida, Vetor, Matriz
12	Programação em C	Linguagem C, Variaveis, Constantes, Operadores, Vetor, Matriz
13	Linguagem máquina	Linguagem de maquina
14	Estruturas de decisão encadeadas – if – else – if – else	Estrutura condicional, Decisao Encadeada, Se
15	Loop infinito em C	Linguagem C, Estrutura de repeticao, Para, Repita
16	Começando a programar	Linguagem C, Sintaxe, Semantica
17	Lógica de Programação - Vetores e Matrizes	Vetor, Matriz, Sintaxe
18	Introdução à Ciência da Computação - Estruturas de Controle – Parte I	Portugol, Estrutura condicional, Decisao simples, Decisao composta, Decisao encadeada
19	Conceito e formas de representação de algoritmos	Algoritmo, Tipos de algoritmo, Narrativa, Fluxograma, Pseudocodigo
20	Lógica de Programação – Desvio Condicional Aninhado (SE... ENTÃO... SENÃO... SE)	Portugol, Estrutura condicional, Decisao composta, Se
21	O que é e para que serve alocação dinâmica	Linguagem C, Vetor
22	Arquivos em c - tutorial completo	Linguagem C
23	Linguagem Natural	Linguagem natural
24	Linguagem de Máquina e Assembler	Linguagem de maquina
25	Expressões Lógicas	Operadores logicos
26	Operadores Relacionais	Operadores relacionais
27	Aspectos básicos de linguagem C	Linguagem C, Tipos de dados, Variaveis, Constantes, Entrada, Saida, Operadores
28	Tipos de dados >>Constantes	Constantes, Portugol
29	Entrada/Saída >>Ler	Entrada, Portugol
30	Entrada/Saída >>Escrever	Saida, Portugol
31	Operadores Aritméticos	Operadores aritmeticos, Portugol
32	Operadores Lógicos	Operadores logicos, Portugol

33	Operadores Relacionais	Operadores relacionais, Portugal
34	Decisão >>Se	Se, Estrutura condicional, Portugal
35	Decisão >>Escolhe	Escolha, Estrutura condicional, Portugal
36	Repetição >>Enquanto	Enquanto, Estrutura de repeticao, Portugal
37	Repetição >>Para	Para, Estrutura de repeticao, Portugal
38	Repetição >>Repete	Repita, Estrutura de repeticao, Portugal
39	Repetição >>Faz	Repita, Estrutura de repeticao, Portugal
40	Linguagem Algorítmica	Sintaxe, Semantica, Portugal
41	Tipos de dados >>Básicos	Tipos de dados, Tipo inteiro, Tipo real, Tipo logico, Tipo caracter, Tipo texto
42	Tipos de dados >>Variáveis	Variaveis, Portugal
43	Fluxogramas, diagrama de blocos e de Chapin no desenvolvimento de algoritmos	Tipos de algoritmo, Fluxograma
44	Estrutura de seleção múltipla ESCOLHA-CASO	Estrutura condicional, Decisao encadeada, Escolha
45	O que é Algoritmo?	Algoritmo, Problema
46	Algoritmos – Estruturas de Controle	Estrutura condicional, Decisao simples, Decisao composta, Decisao encadeada, Se, Escolha
47	O uso cotidiano do algoritmo	Aplicacao de algoritmo
48	Comandos de entrada e saída	Linguagem C, Entrada, Saida, Atribuicao, Operadores
49	Estruturas Condicionais	Linguagem C, Estrutura condicional, Operadores relacionais
50	Programação para Engenharia I	Linguagem C, Tipos de dados, Tipo texto
51	Comandos de decisão Comandos de seleção	Linguagem C, Estrutura condicional, Estrutura de repeticao

52	Introdução à Linguagem C	Variaveis, Tipos de dados, Tipo texto, Entrada, Saida, Tipo real, Operadores aritmeticos
53	Matrizes e Vetores	Linguagem C, Vetor, Matriz, Entrada, Saida, Atribuicao
54	Arrays multidimensionais	Vetor, Matriz
55	Operadores Lógicos	Linguagem C, Operadores logicos
56	Manual do Visualg	Portugol, Variaveis, Tipos de dados, Sintaxe, Constantes, Atribuiçao, Entrada, Saida, Operadores, Estrutura condicional, Estrutura de repeticao
57	Noção e Representação de Algoritmos	Algoritmo, Tipos de algoritmo, Problema, Narrativa, Fluxograma
58	Representação de Algoritmos	Algoritmo, Tipos de algoritmo, Narrativa, Fluxograma, Diagrama de Chapin, Pseudocodigo
59	Estrutura de seleção	Pseudocodigo, Estrutura condicional
60	Introdução à linguagem C - Parte 1	Linguagem C, Tipos de dados, Operadores relacionais, Tipo texto
61	Introdução à lógica de programação	Algoritmo, Logica de programacao, Portugol, Pseudolinguagem
62	Laços de repetição	Pseudocodigo, Estrutura de repeticao
63	Vetores e Matrizes	Pseudocodigo, Vetor, Matriz
64	O que é algoritmo?	Algoritmo, Tipos de algoritmo
65	O que é um algoritmo?	Algoritmo
66	Vetores e Matrizes	Vetor, Matriz
67	Algoritmo e Programação	Algoritmo, Tipos de algoritmo, Fluxograma, Narrativa, Portugol, Variaveis, Tipos de dados
68	Estruturas de Controle	Estrutura sequencial, Estrutura condicional, Decisao simples, Decisao composta, Decisao encadeada, Estrutura de repeticao, Para, Repita, Enquanto

69	As 15 principais linguagens de programação do mundo!	Linguagem de programação
70	O que são Vetores e Matrizes (arrays)	Vetor, Matriz, Portugol, Estrutura de repetição
71	Conheça os Operadores Relacionais!	Portugol, Operadores relacionais
72	Aprendendo a interpretar exercícios de algoritmos	Algoritmo, Problema, Entrada, Processamento, Saida
73	Operadores de atribuição C	Linguagem C, Atribuição
74	Operadores relacionais e de igualdade C	Linguagem C, Operadores relacionais
75	Operador de expressão condicional	Linguagem C, Operadores lógicos, Operadores relacionais, Estrutura condicional
76	Operadores lógicos C	Linguagem C, Operadores lógicos
77	Algoritmo	Algoritmo, Tipos de algoritmo, Programa de computador, Interpretador, Compilador
78	Estrutura de controle	Estrutura sequencial, Estrutura condicional, Estrutura de repetição
79	Estrutura de repetição	Estrutura de repetição, Enquanto, Para, Repita
80	Operadores em C e C++	Linguagem C, Operadores
81	Comandos de Repetição (Laços ou Loops)	Linguagem C, Estrutura de repetição, Enquanto, Para, Repita
82	Outros conceitos sobre lógica de programação: Tipos de dados, variáveis e expressões	Tipos de dados, Variáveis, Operações, Operações aritméticas, Operações lógicas
83	Linguagens de Programação	Linguagem de programação, Linguagem de máquina, Compilador
84	Estrutura de Decisão em C/C++	Linguagem C, Estrutura condicional, Se, Escolha
85	Fundamentos	Linguagem C, Entrada, Saida, Atribuição, Operadores, Variáveis, Precedência de operadores
86	Alocação dinâmica de memória	Vetor, Matriz, Linguagem C

87	Strings	Tipo texto, Entrada, Saida, Constantes
88	Programação C/C++ - Matrizes e Vetores	Linguagem C, Vetor, Matriz
89	The if-else Statement in c programming language	Se, Linguagem C
90	While Statement in c programming language	Enquanto, Linguagem C
91	C Programming Multidimensional Arrays	Linguagem C, Vetor, Matriz
92	C- arrays	Linguagem C, Vetor
93	Multi-dimensional Arrays in C	Linguagem C, Matriz
94	C - Strings	Linguagem C, Tipo texto
95	Programar em C - Revisão Vetores/Matrizes	Vetor, Matriz, Problema
96	Lógica de programação - Aula 03 - Legibilidade do código	Algoritmo, Comentario, Blocos
97	Lógica de programação - Aula 01 - Introdução	Logica de programacao, Aplicacao de algoritmo
98	Portugol - VisuALG - Aula 01 (Princípios Básicos)	Tipos de dados, Tipo inteiro, Tipo real, Tipo logico, Tipo caracter, Entrada, Saida, Atribuicao, Operadores aritmeticos
99	Libraries	Linguagem C
100	Lógica de programação - Aula 05 - Expressões, operadores e comandos	Entrada, Saida, Operadores, Operadores relacionais, Operadores lógicos, Operadores aritmeticos, Atribuicao
101	Lógica de programação - Aula 02 - Tipos de algoritmo	Tipos de algoritmo, Fluxograma, Pseudocodigo, Narrativa
102	Programming Basics: Statements & Functions: Crash Course Computer Science	Semantica, Sintaxe, Blocos
103	Portugol Studio - Teste de Mesa	Algoritmo, Processamento, Logica de programacao
104	Por que todos deveriam aprender a programar?	Aplicacao de algoritmo

105	Vetores e Matrizes	Vetor, Matriz, Fluxograma, Pseudocódigo
106	Lógica de Programação - Estruturas de Repetição (Enquanto, Para, FaçaEnquanto)	Logica de programacao, Estrutura de repeticao, Para, Enquanto, Repita
107	Lógica de programação - Aula 04 - Variáveis e constantes	Variaveis, Constantes, Tipos de dados, Tipo inteiro, Tipo real, Tipo texto, Tipo logico, Vetor, Matriz
108	As principais bibliotecas em linguagem C	Linguagem C

Tabela 13 – Palavras-chave associadas aos Links de Apoio

APÊNDICE B – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

B.1 DESCRIÇÃO DO MINICURSO

Você está sendo convidado a participar do Minicurso de Algoritmos vinculado a um projeto de mestrado. Este projeto intitula-se “Sistema de Recomendação Sensível ao Tempo em Ambientes Virtuais de Aprendizagem” e visa descobrir se o uso da informação temporal em um Sistema de Recomendação influencia na qualidade dos itens recomendados. Durante o minicurso, seus dados de utilização serão coletados e posteriormente analisados pelos pesquisadores envolvidos no projeto.

B.2 PROCEDIMENTO

Após o período de matrícula (de 09/04/2018 a 13/04/2018) no minicurso, todos os alunos matriculados terão acesso ao conteúdo do minicurso a partir do dia 16/04/2018. Ao final do minicurso, os alunos realizarão uma avaliação final e responderão a um questionário de satisfação referente ao minicurso. Durante o minicurso, os dados de navegação/interação dos alunos com o AdaptWeb® serão coletados para análise posterior com o objetivo de descobrir aspectos positivos e negativos das ferramentas existentes no sistema.

B.3 RISCOS E DESCONFORTOS

A participação neste minicurso não apresenta riscos diretos a seus participantes. Caso você não se sinta confortável em ter suas informações coletadas; não goste do assunto abordado, da metodologia utilizada ou do material utilizado; ou ainda por quaisquer outros motivos não deseje continuar a participar do minicurso, você está livre pra desistir a qualquer momento.

B.4 BENEFÍCIOS DA SUA PARTICIPAÇÃO

Esperamos que os resultados obtidos auxiliem a identificar os benefícios práticos do uso da informação temporal em um Sistema de Recomendação através das recomendações realizadas para os alunos durante sua interação, suas características positivas e negativas. Desta forma, esperamos contribuir com uma melhor experiência do usuário em ambientes de Educação a Distância.

B.5 CUSTOS

Sua participação no minicurso não acarretará em nenhum custo. Você também não será pago(a) para participar.

B.6 CONFIDENCIALIDADE

Sua identidade será preservada, pois você será referenciado por um identificador numérico, de forma que seu nome nunca será citado. As únicas pessoas que terão acesso aos dados brutos serão as pesquisadoras envolvidas no projeto: Eduardo José de Borba, profa. Dra. Isabela Gasparini e prof. Dr. Daniel Lichtnow. Os resultados, sem identificações, poderão ser veiculados em artigos técnicos e científicos.

B.7 DÚVIDAS

Caso haja qualquer dúvida a respeito do minicurso, sinta-se à vontade para entrar em contato.

Eduardo José de Borba (Aluno do Programa de Pós-Graduação em Computação Aplicada da Universidade do Estado de Santa Catarina)
E-mail: eduardojoseborba@gmail.com

Dra. Isabela Gasparini (Orientadora)
E-mail: isabela.gasparini@udesc.br

Dr. Daniel Lichtnow (Coorientador)
E-mail: dlichtnow@politecnico.ufsm.br

Endereço para contato:

Departamento de Ciência da Computação (DCC)
Centro de Ciências Tecnológicas (CCT)
Rua Paulo Malschitzki, 200 - Campus Universitário Prof. Avelino Marcante
- Bairro Zona Industrial Norte
Joinville - SC - Brasil

Termo de Consentimento Livre e Esclarecido

Solicitamos a sua permissão para utilizarmos os dados coletados, bem como para divulgar os resultados em artigos técnicos e científicos. Lembramos que iremos

garantir sua privacidade. Destacamos que este estudo visa avaliar a ferramenta e não os participantes. Nós queremos saber a sua opinião!

- ☐ Declaro que fui informado sobre todos os procedimentos da pesquisa e, que recebi de forma clara e objetiva todas as explicações pertinentes ao projeto e, que todos os dados coletados serão sigilosos. Eu compreendo que neste estudo, as medições dos experimentos/procedimentos de tratamento serão feitas sobre minhas ações no sistema.
- ☐ Declaro que meu responsável está ciente que estou participando deste minicurso, que dados sobre mim estão sendo coletados e que minha identidade será preservada.

Nome do responsável:

E-mail para contato:

APÊNDICE C – TESTE PILOTO

Protocolo:

- Primeira etapa:

1. Acessar a página do AdaptWeb: <http://200.19.107.172/adaptweb/>
2. Logar com o usuário testepiloto1@udesc.br e senha testepiloto1
3. Acessar o link <http://200.19.107.172/adaptweb/index.php?opcao=Minicurso-Algoritmos> e realizar a matrícula na disciplina
4. Acessar o ambiente de aula da disciplina pelo modo de navegação dado:
5. Livre ou Tutorial
6. Acessar 5 conceitos da disciplina
7. Encontrar as recomendações
8. Quantos recomendações você recebeu?
9. Olhando pelo título e descrição dos links, eles parecem estar de acordo com os conceitos que você acessou?
10. Acessar as recomendações
11. Avaliar as recomendações
12. Acessar mais 3 conceitos da disciplina e um material complementar (dica: use o mapa de navegação presente na parte superior da tela)
13. Analisar as recomendações para perceber se algo mudou
14. Pedir para o aplicador do teste piloto habilitar a avaliação da disciplina
15. Realizar a avaliação da disciplina
16. Ao final da avaliação, responder ao questionário de satisfação

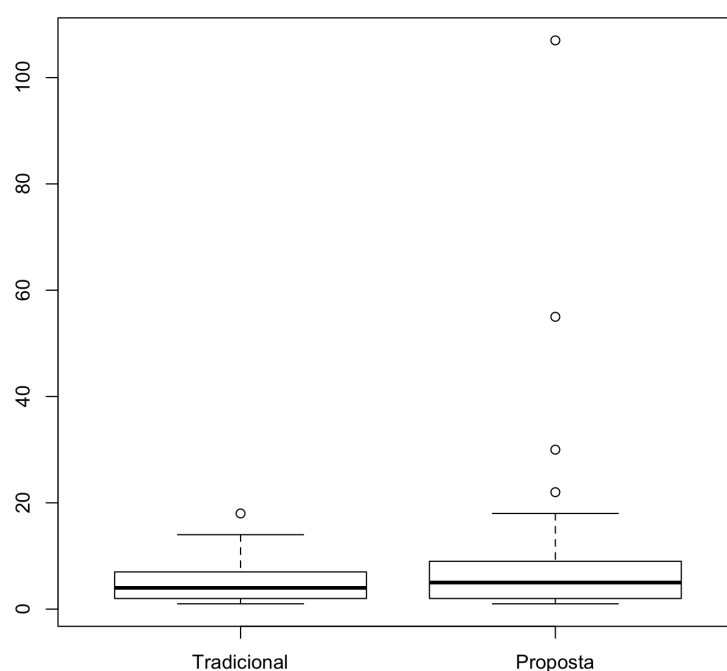
- Segunda etapa (avise o aplicador do teste antes de iniciar essa etapa):

1. Deslogar do ambiente se ainda estiver logado
2. Acessar o link <http://200.19.107.172/adaptweb/index.php?opcao=Minicurso-Algoritmos> para criar um usuário no ambiente e depois se matricular na disciplina
3. Verificar como está o acesso à disciplina

APÊNDICE D – ANÁLISE ESTATÍSTICA DO DESEMPENHO DO SISTEMA DE RECOMENDAÇÃO

D.1 LINKS ACESSADOS POR ALUNO

Figura 7 – Boxplot dos links acessados por aluno



Fonte: O autor.

Total de alunos comparados: 85

Tradicional

Min = 1.000
 1° Quad = 2.000
 Mediana = 4.000
 Média = 4.935
 3° Quad = 6.500
 Max = 18.000

Proposta

Min = 1.00
 1° Quad = 2.00
 Mediana = 5.00
 Média = 10.15
 3° Quad = 9.00
 Max = 107.00

Shapiro-Wilk normality test

data: data[["quantidade"]]

W = 0.42461, p-value < 2.2e-16

Resultado: Aceita a hipótese alternativa - Distribuição não normal

Wilcoxon rank sum test with continuity correction

data: data[["quantidade"]] by data[["algoritmo_recomendacao"]]

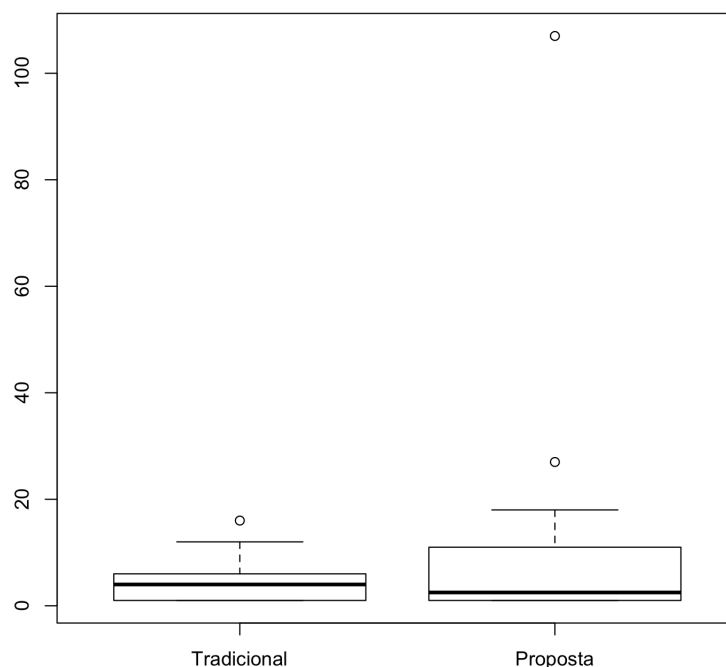
W = 820, p-value = 0.4957

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese nula - Sem diferença significativa

D.2 LINKS AVALIADOS POSITIVAMENTE POR ALUNO

Figura 8 – Boxplot dos links avaliados positivamente por aluno



Fonte: O autor.

Total de alunos comparados: 52

Tradicional

Min = 1.0
 1° Quad = 1.0
 Mediana = 4.0
 Média = 4.7
 3° Quad = 6.0
 Max = 16.0

Proposta

Min = 1.00
 1° Quad = 1.00
 Mediana = 2.50
 Média = 10.64
 3° Quad = 9.75
 Max = 107.00

Shapiro-Wilk normality test

data: data[["quantidade"]]

W = 0.37836, p-value = 1.654e-13

Resultado: Aceita a hipótese alternativa - Distribuição não normal

Wilcoxon rank sum test with continuity correction

data: data[["quantidade"]] by data[["algoritmo_recomendacao"]]

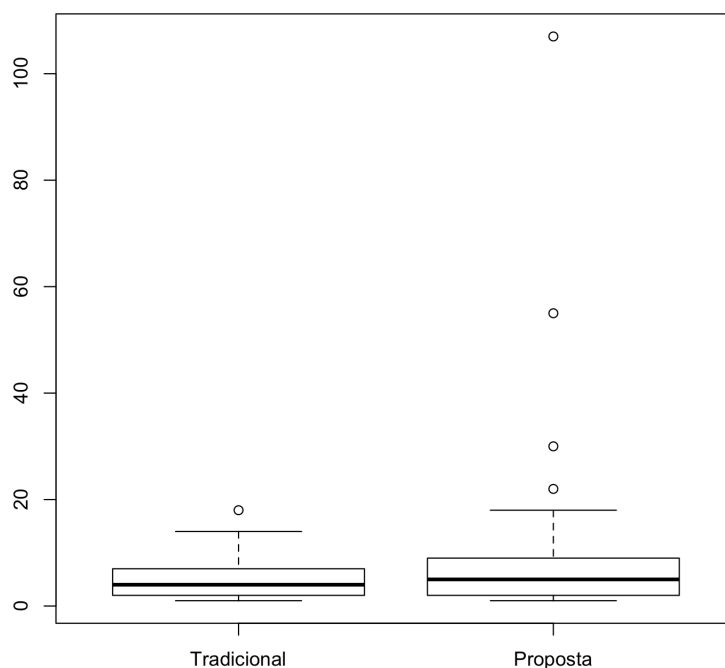
W = 318, p-value = 0.8275

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese nula - Sem diferença significativa

D.3 LINKS AVALIADOS NEGATIVAMENTE POR ALUNO QUE ACESSOU PELO MENOS UMA RECOMENDAÇÃO

Figura 9 – Boxplot dos links acessados por aluno



Fonte: O autor.

Total de alunos comparados: 7

Tradicional

Min = 1
 1º Quad = 1
 Mediana = 1
 Média = 1
 3º Quad = 1
 Max = 1

Proposta

Min = 1.0
 1º Quad = 1.5
 Mediana = 2.0
 Média = 2.0
 3º Quad = 2.5
 Max = 3.0

Shapiro-Wilk normality test

data: data[["quantidade"]]

W = 0.45297, p-value = 4.136e-06

Resultado: Aceita a hipótese alternativa - Distribuição não normal

Wilcoxon rank sum test with continuity correction

data: data[["quantidade"]] by data[["algoritmo_recomendacao"]]

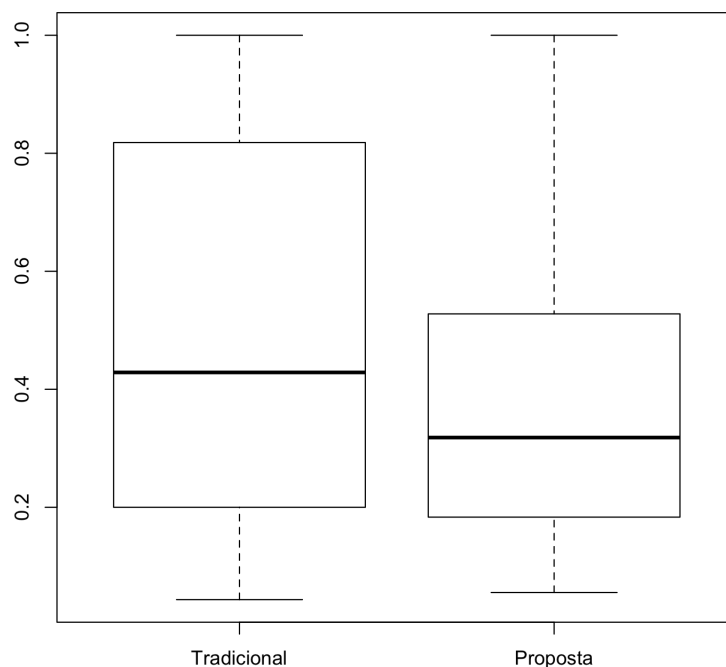
W = 2.5, p-value = 0.2059

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese nula - Sem diferença significativa

D.4 PRECISÃO DOS ALGORITMOS DE RECOMENDAÇÃO POR ALUNO

Figura 10 – Boxplot da precisão dos algoritmos de recomendação por aluno



Fonte: O autor.

Total de alunos comparados: 85

Tradicional

Min = 0.04348
 1º Quad = 0.20000
 Mediana = 0.42857
 Média = 0.49679
 3º Quad = 0.81364
 Max = 1.00000

Proposta

Min = 0.05556
 1º Quad = 0.18333
 Mediana = 0.31818
 Média = 0.39859
 3º Quad = 0.52778
 Max = 1.00000

Shapiro-Wilk normality test

data: data[["precisao"]]

W = 0.89495, p-value = 4.234e-06

Resultado: Aceita a hipótese alternativa - Distribuição não normal

Wilcoxon rank sum test with continuity correction

data: data[["precisao"]] by data[["algoritmo_recomendacao"]]

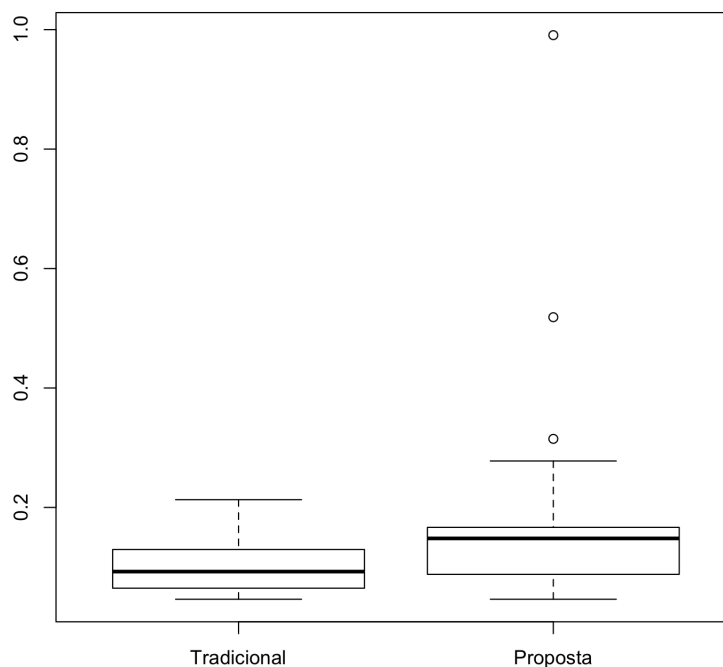
W = 1025, p-value = 0.2603

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese nula - Sem diferença significativa

D.5 COBERTURA DOS ALGORITMOS DE RECOMENDAÇÃO POR ALUNO

Figura 11 – Boxplot da cobertura dos algoritmos de recomendação por aluno



Fonte: O autor.

Total de alunos comparados: 85

Tradicional

Min = 0.04630
 1º Quad = 0.06481
 Mediana = 0.09259
 Média = 0.10064
 3º Quad = 0.12963
 Max = 0.21296

Proposta

Min = 0.04630
 1º Quad = 0.08796
 Mediana = 0.14815
 Média = 0.17450
 3º Quad = 0.16667
 Max = 0.99074

Shapiro-Wilk normality test

data: data[["coverage"]]

W = 0.55934, p-value = 1.421e-14

Resultado: Aceita a hipótese alternativa - Distribuição não normal

Wilcoxon rank sum test with continuity correction

data: data[["coverage"]] by data[["algoritmo_recomendacao"]]

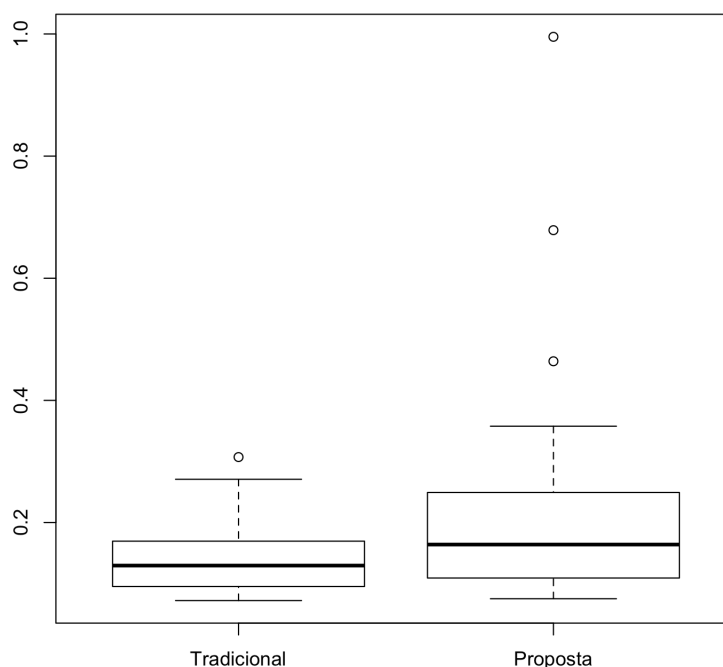
W = 525.5, p-value = 0.001031

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese alternativa - Com diferença significativa

D.6 F-MEASURE DO ALGORITMO DE RECOMENDAÇÃO POR ALUNO

Figura 12 – Boxplot do F-measure dos dois algoritmos



Fonte: O autor.

Total de alunos comparados: 85

Tradicional

Min = 0.07221
 1º Quad = 0.09545
 Mediana = 0.12952
 Média = 0.13852
 3º Quad = 0.16618
 Max = 0.30717

Proposta

Min = 0.07519
 1º Quad = 0.10913
 Mediana = 0.16393
 Média = 0.21719
 3º Quad = 0.24919
 Max = 0.99535

Shapiro-Wilk normality test

data: data[["media_harmonica"]]

W = 0.63292, p-value = 2.931e-13

Resultado: Aceita a hipótese alternativa - Distribuição não normal

Wilcoxon rank sum test with continuity correction

data: data[["media_harmonica"]] by data[["algoritmo_recomendacao"]]

W = 634.5, p-value = 0.02082

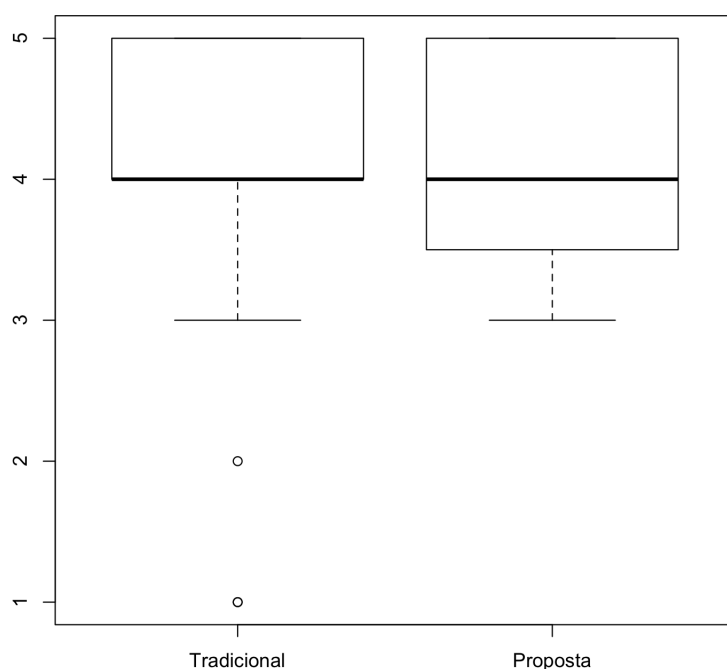
alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese alternativa - Com diferença significativa

APÊNDICE E – ANÁLISE ESTATÍSTICA DAS QUESTÕES DO QUESTIONÁRIO

E.1 QUESTÃO 1: OS ITENS RECOMENDADOS CORRESPONDERAM COM OS MEUS INTERESSES.

Figura 13 – Boxplot da questão 1



Fonte: O autor.

Tradicional

Min = 1.000
 1º Quad = 4.000
 Mediana = 4.000
 Média = 4.032
 3º Quad = 5.000
 Max = 5.000

Proposta

Min = 3.000
 1º Quad = 3.750
 Mediana = 4.000
 Média = 4.042
 3º Quad = 5.000
 Max = 5.000

Wilcoxon rank sum test with continuity correction

data: *data_1_tradicional* and *data_1_proposta*

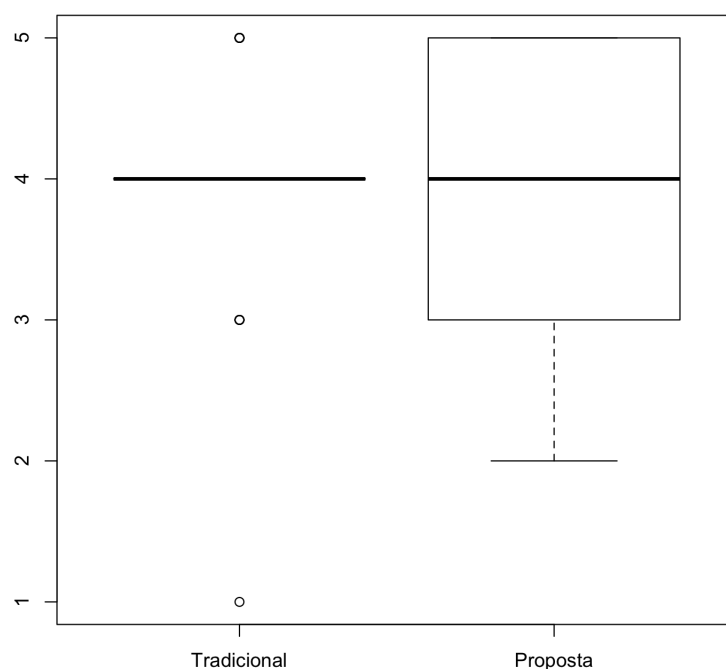
W = 404, p-value = 0.5635

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese nula - Sem diferença significativa

E.2 QUESTÃO 2: OS ITENS RECOMENDADOS PARA MIM SÃO DIVERSIFICADOS (O SISTEMA SE PREOCUPA EM TRAZER ITENS DIFERENTES A CADA RECOMENDAÇÃO).

Figura 14 – Boxplot da questão 2



Fonte: O autor.

Tradicional

Min = 1.000
 1° Quad = 4.000
 Mediana = 4.000
 Média = 3.933
 3° Quad = 4.000
 Max = 5.000

Proposta

Min = 2
 1° Quad = 3
 Mediana = 4
 Média = 4
 3° Quad = 5
 Max = 5

Wilcoxon rank sum test with continuity correction

data: *data_2_tradicional* and *data_2_proposta*

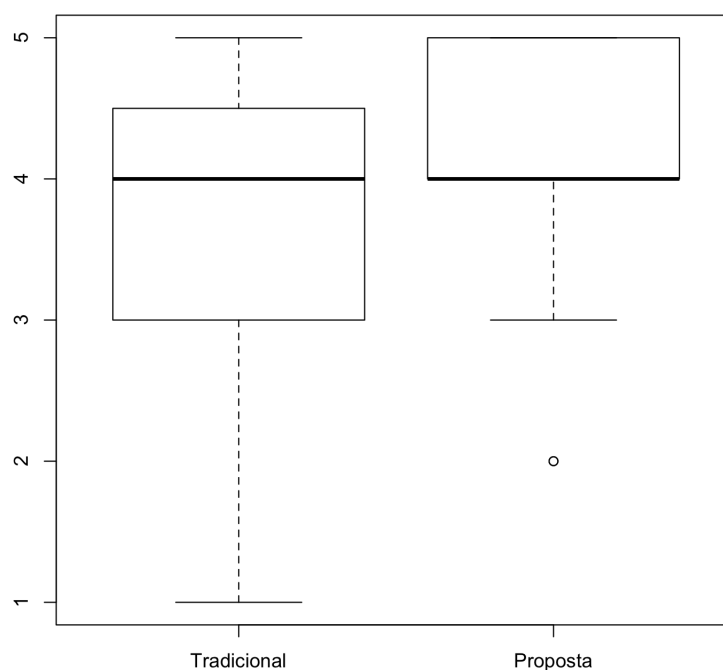
W = 376.5, p-value = 0.8178

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese nula - Sem diferença significativa

E.3 QUESTÃO 3: OS ITENS RECOMENDADOS CORRESPONDERAM AOS INTERESSES E NECESSIDADES QUE EU TINHA NO MOMENTO.

Figura 15 – Boxplot da questão 3



Fonte: O autor.

Tradicional

Min = 1.000
 1° Quad = 3.000
 Mediana = 4.000
 Média = 3.806
 3° Quad = 4.500
 Max = 5.000

Proposta

Min = 2
 1° Quad = 4
 Mediana = 4
 Média = 4
 3° Quad = 5
 Max = 5

Wilcoxon rank sum test with continuity correction

data: *data_3_tradicional* and *data_3_proposta*

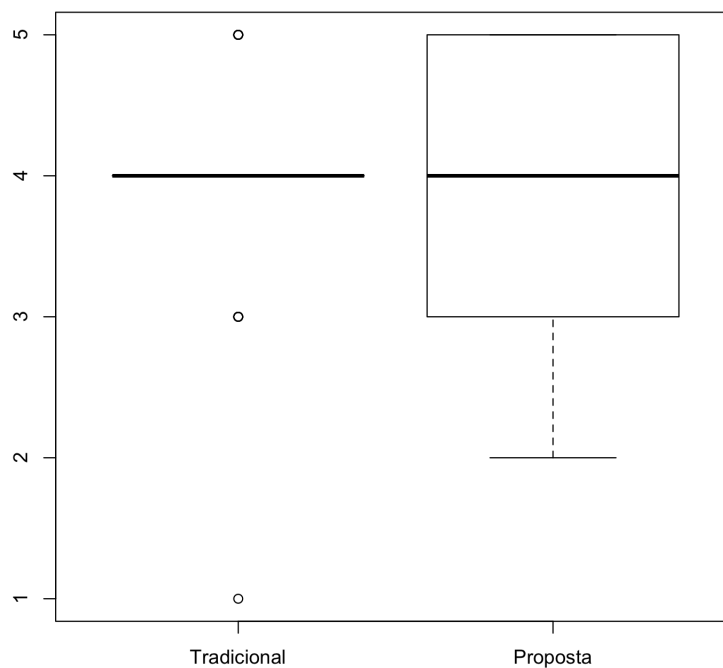
W = 364, p-value = 0.5119

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese nula - Sem diferença significativa

E.4 QUESTÃO 4: AS RECOMENDAÇÕES SÃO FEITAS NO MOMENTO ADEQUADO.

Figura 16 – Boxplot da questão 4



Fonte: O autor.

Tradicional

Min = 1.000
 1º Quad = 4.000
 Mediana = 4.000
 Média = 3.931
 3º Quad = 4.000
 Max = 5.000

Proposta

Min = 2
 1º Quad = 3
 Mediana = 4
 Média = 4
 3º Quad = 5
 Max = 5

Wilcoxon rank sum test with continuity correction

data: *data_4_tradicional* and *data_4_proposta*

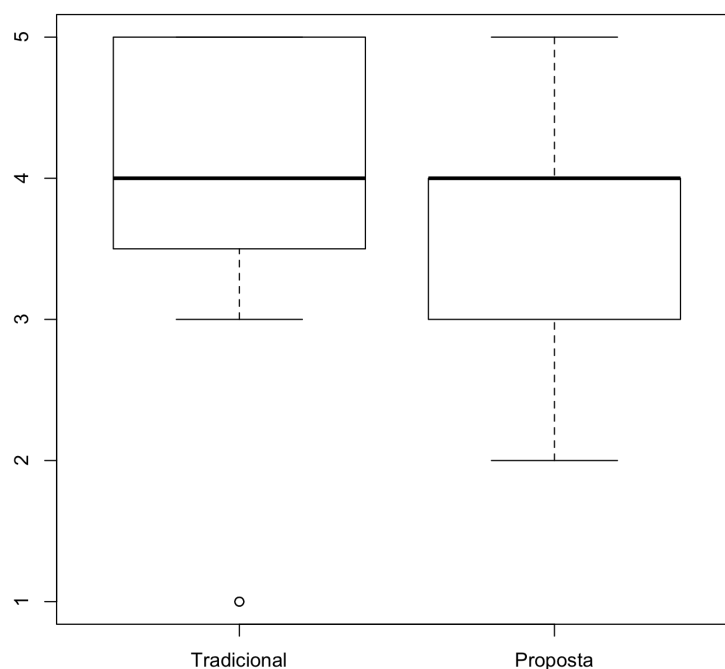
W = 378, p-value = 0.8198

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese nula - Sem diferença significativa

E.5 QUESTÃO 5: O SISTEMA DE RECOMENDAÇÃO EXPLICA PORQUE OS LINKS SÃO RECOMENDADOS PARA MIM.

Figura 17 – Boxplot da questão 5



Fonte: O autor.

Tradicional

Min = 1.000
 1° Quad = 3.500
 Mediana = 4.000
 Média = 3.889
 3° Quad = 5.000
 Max = 5.000

Proposta

Min = 2.00
 1° Quad = 3.00
 Mediana = 4.00
 Média = 3.72
 3° Quad = 4.00
 Max = 5.00

Wilcoxon rank sum test with continuity correction

data: *data_5_tradicional* and *data_5_proposta*

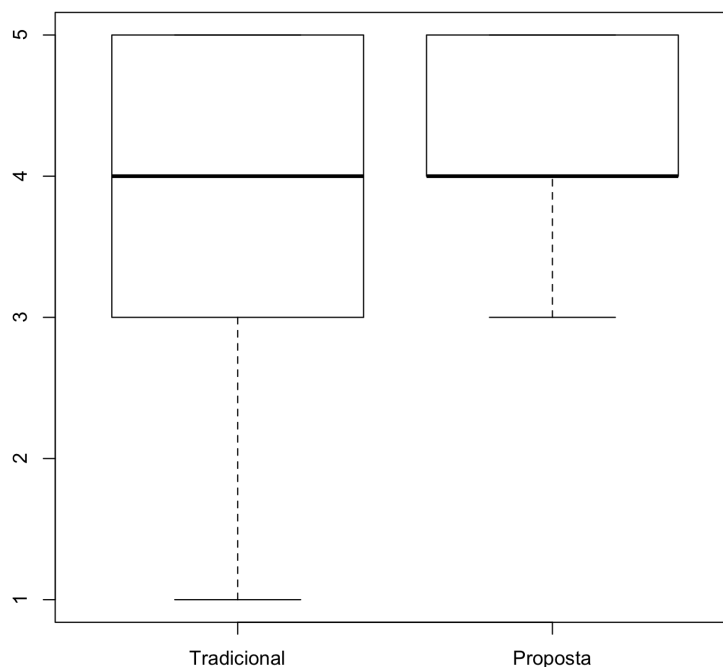
W = 396, p-value = 0.2665

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese nula - Sem diferença significativa

E.6 QUESTÃO 6: A INFORMAÇÃO APRESENTADA NA INTERFACE PARA OS ITENS RECOMENDADOS É SUFICIENTE PARA MIM.

Figura 18 – Boxplot da questão 6



Fonte: O autor.

Tradicional

Min = 1.000
 1° Quad = 3.000
 Mediana = 4.000
 Média = 3.655
 3° Quad = 5.000
 Max = 5.000

Proposta

Min = 3.000
 1° Quad = 4.000
 Mediana = 4.000
 Média = 4.192
 3° Quad = 5.000
 Max = 5.000

Wilcoxon rank sum test with continuity correction

data: *data_6_tradicional* and *data_6_proposta*

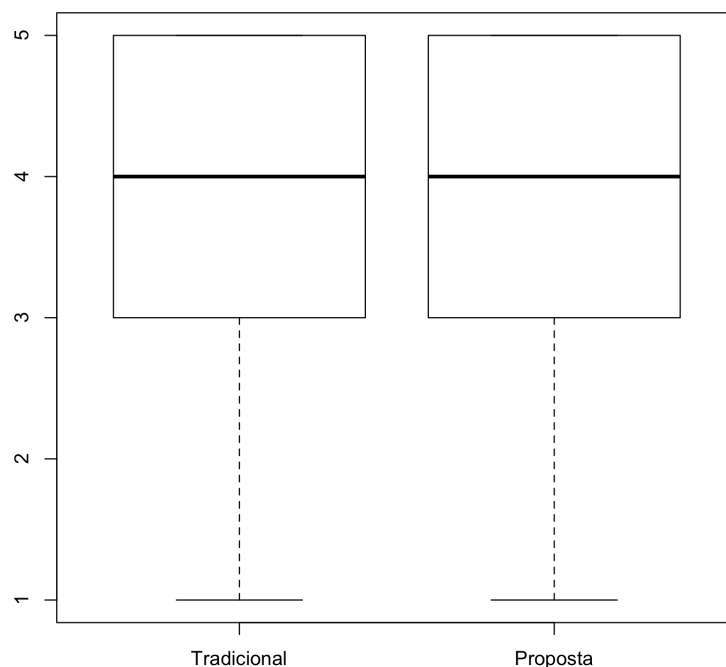
W = 301.5, p-value = 0.1799

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese nula - Sem diferença significativa

E.7 QUESTÃO 7: O LAYOUT DO SISTEMA DE RECOMENDAÇÃO É ATRATIVO E ADEQUADO.

Figura 19 – Boxplot da questão 7



Fonte: O autor.

Tradicional

Min = 1.000
 1° Quad = 3.000
 Mediana = 4.000
 Média = 3.676
 3° Quad = 4.750
 Max = 5.000

Proposta

Min = 1.0
 1° Quad = 3.0
 Mediana = 4.0
 Média = 3.8
 3° Quad = 5.0
 Max = 5.0

Wilcoxon rank sum test with continuity correction

data: *data_7_tradicional* and *data_7_proposta*

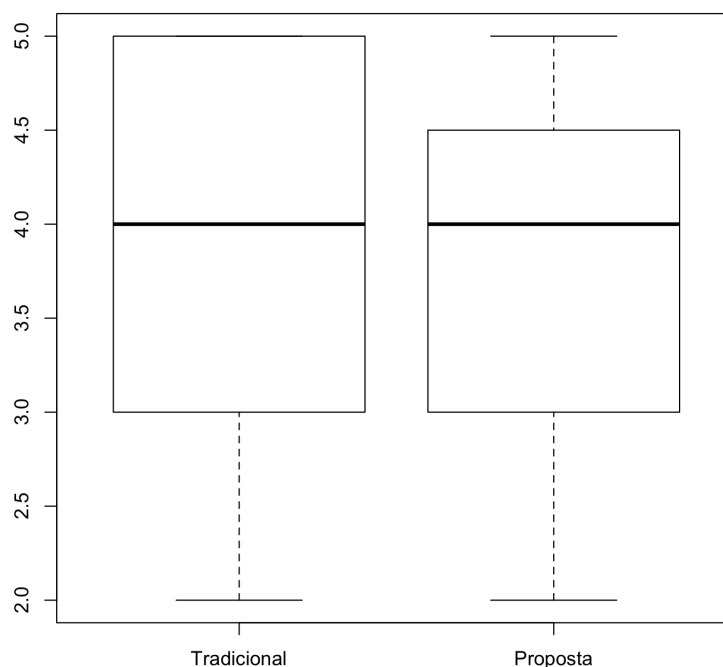
W = 413, p-value = 0.8536

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese nula - Sem diferença significativa

E.8 QUESTÃO 8: EU ENCONTREI FACILMENTE O LOCAL ONDE OS ITENS SÃO RECOMENDADOS.

Figura 20 – Boxplot da questão 8



Fonte: O autor.

Tradicional

Min = 2.0
 1° Quad = 3.0
 Mediana = 4.0
 Média = 3.8
 3° Quad = 5.0
 Max = 5.0

Proposta

Min = 2.00
 1° Quad = 3.00
 Mediana = 4.00
 Média = 3.75
 3° Quad = 4.25
 Max = 5.00

Wilcoxon rank sum test with continuity correction

data: *data_8_tradicional* and *data_8_proposta*

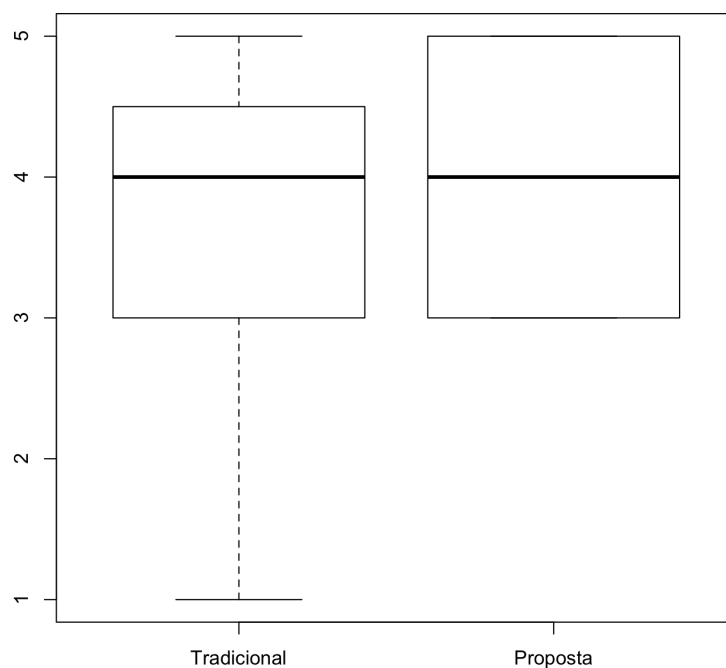
W = 381, p-value = 0.7093

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese nula - Sem diferença significativa

**E.9 QUESTÃO 9: EU PERCEBI QUE O SISTEMA DE RECOMENDAÇÃO APREN-
DIA SOBRE MINHAS NECESSIDADES/PREFERÊNCIAS CONFORME EU AVAN-
ÇAVA NA DISCIPLINA.**

Figura 21 – Boxplot da questão 9



Fonte: O autor.

Tradicional

Min = 1.000
1° Quad = 3.000
Mediana = 4.000
Média = 3.643
3° Quad = 4.250
Max = 5.000

Proposta

Min = 3.000
1° Quad = 3.000
Mediana = 4.000
Média = 3.875
3° Quad = 5.000
Max = 5.000

Wilcoxon rank sum test with continuity correction

data: *data_9_tradicional* and *data_9_proposta*

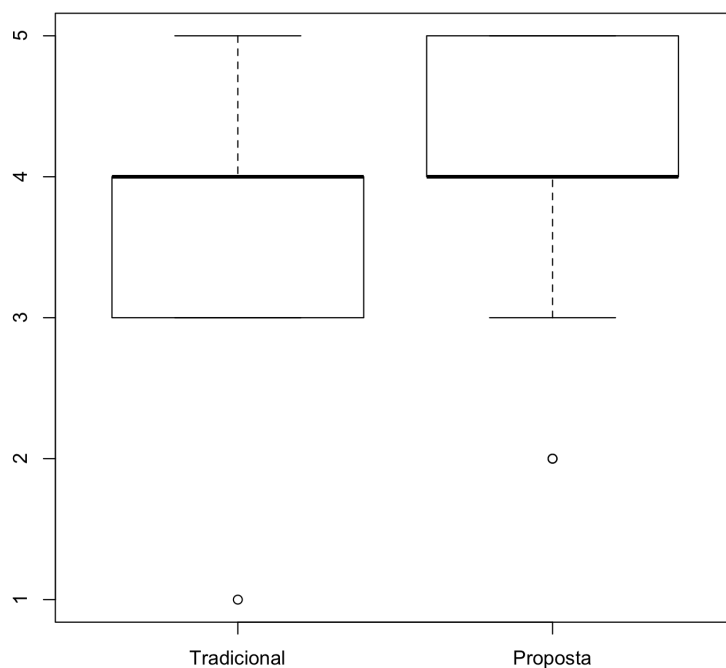
W = 313.5, p-value = 0.6722

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese nula - Sem diferença significativa

E.10 QUESTÃO 10: É FACIL ENCONTRAR UM ITEM PARA ESTUDAR COM A AJUDA DO SISTEMA DE RECOMENDAÇÃO.

Figura 22 – Boxplot da questão 10



Fonte: O autor.

Tradicional

Min = 1.00
 1° Quad = 3.00
 Mediana = 4.00
 Média = 3.69
 3° Quad = 4.00
 Max = 5.00

Proposta

Min = 2
 1° Quad = 4
 Mediana = 4
 Média = 4
 3° Quad = 5
 Max = 5

Wilcoxon rank sum test with continuity correction

data: *data_10_tradicional* and *data_10_proposta*

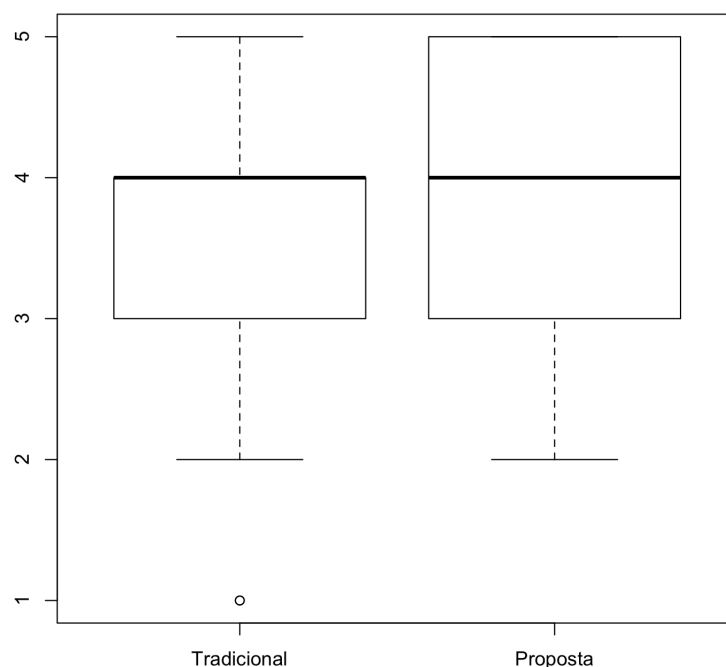
W = 296, p-value = 0.1452

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese nula - Sem diferença significativa

E.11 QUESTÃO 11: EU ME SENTI APOIADO PARA ENCONTRAR ITENS DO MEU INTERESSE COM A AJUDA DO SISTEMA DE RECOMENDAÇÃO.

Figura 23 – Boxplot da questão 11



Fonte: O autor.

Tradicional

Min = 1.000
 1º Quad = 3.000
 Mediana = 4.000
 Média = 3.607
 3º Quad = 4.000
 Max = 5.000

Proposta

Min = 2
 1º Quad = 3
 Mediana = 4
 Média = 4
 3º Quad = 5
 Max = 5

Wilcoxon rank sum test with continuity correction

data: *data_11_tradicional* and *data_11_proposta*

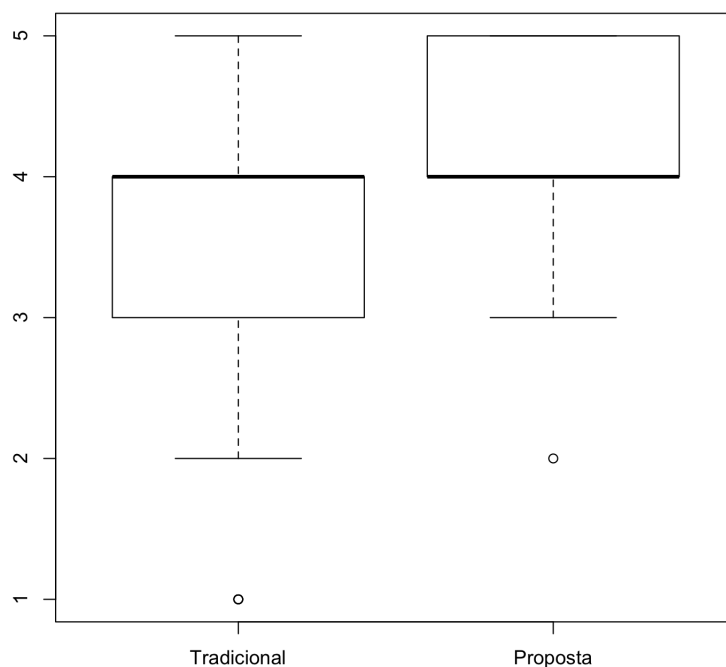
W = 286, p-value = 0.2309

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese nula - Sem diferença significativa

E.12 QUESTÃO 12: EU ENTENDI PORQUE OS ITENS FORAM RECOMENDADOS PARA MIM.

Figura 24 – Boxplot da questão 12



Fonte: O autor.

Tradicional

Min = 1.0
 1° Quad = 3.0
 Mediana = 4.0
 Média = 3.4
 3° Quad = 4.0
 Max = 5.0

Proposta

Min = 2.00
 1° Quad = 4.00
 Mediana = 4.00
 Média = 4.12
 3° Quad = 5.00
 Max = 5.00

Wilcoxon rank sum test with continuity correction

data: *data_12_tradicional* and *data_12_proposta*

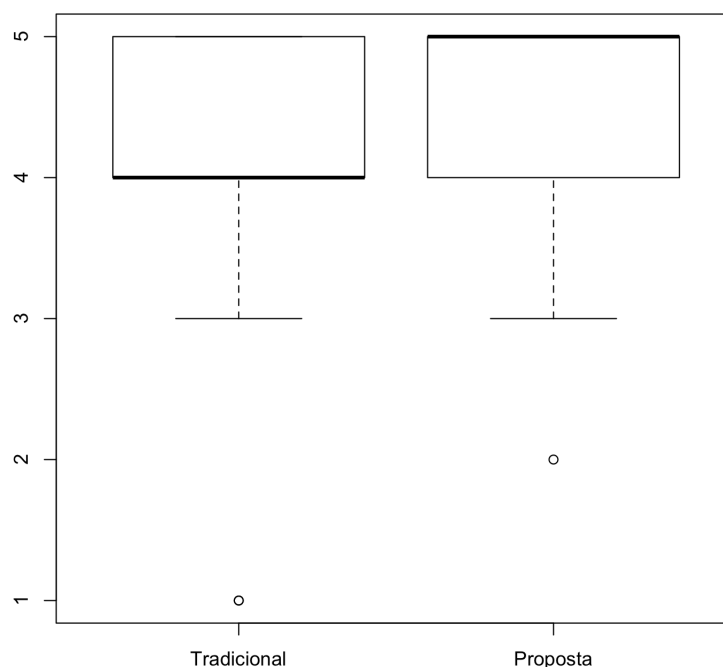
W = 240, p-value = 0.01513

alternative hypothesis: true location shift is not equal to 0

Resultado: Aceita a hipótese alternativa - Com diferença significativa

E.13 QUESTÃO 13: NO GERAL, ESTOU SATISFEITO COM O SISTEMA DE RECOMENDAÇÃO.

Figura 25 – Boxplot da questão 13



Fonte: O autor.

Tradicional

Min = 1
 1º Quad = 4
 Mediana = 4
 Média = 4
 3º Quad = 5
 Max = 5

Proposta

Min = 2.000
 1º Quad = 4.000
 Mediana = 5.000
 Média = 4.308
 3º Quad = 5.000
 Max = 5.000

Wilcoxon rank sum test with continuity correction

data: *data_13_tradicional* and *data_13_proposta*

W = 356.5, p-value = 0.2388

alternative hypothesis: true location shift is not equal to 0

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.*

ANEXO B – INTERVENÇÕES

No decorrer do minicurso, foram realizadas 12 intervenções por e-mail e pelo mural de recados para estimular os alunos de ambos os grupos (i.e., os alunos que usaram o Sistema de Recomendação Tradicional ou a proposta desse trabalho) a acessar o ambiente de aula e se aprofundar nos materiais instrucionais liberados. Cada uma dessas intervenções e sua respectiva data é apresentada abaixo. As intervenções utilizadas foram adaptadas de Santos (2017).

Figura 26 – Intervenção 1 (16/04/18)

Seja bem-vind@ ao Minicurso de Algoritmos e Programação!

A partir de agora você já pode acessar o ambiente do minicurso através do sistema [AdaptWeb](#).

Para ganhar o **certificado de participação**, você precisa:

- Estudar os conceitos, exemplos, exercícios e materiais complementares que achar importante, acessando o ambiente quando desejar (no período de 16/04 a 10/05);
- Fazer a Avaliação (que estará aberta no período de 11/05 a 14/05);
- Responder o Questionário de Satisfação (que estará aberto no período de 11/05 a 14/05).

Após completado esses itens, você ganhará o certificado de participação com carga horária de 20 horas que poderá ser validado como **Atividade Complementar**. Além disso, teremos um brinde surpresa como um Bônus de participação!

Teremos mensagens periódicas! Durante o curso, você receberá mensagens como esta, para lembrá-l@ de:

- Datas de início e término de atividades.
- Sugestões de estudo para que você não esqueça dos conteúdos abordados, e nem deixe acumular conteúdos para a última hora.
- Propostas de desafios de programação, baseados nos conteúdos sugeridos nas mensagens.

Sugerimos agora que você acesse e explore o ambiente. Aproveite este período inicial para tirar dúvidas e resolver eventuais problemas de acesso.



Obs.: Elaboramos o conteúdo deste minicurso com muito carinho e atenção, mas nem tudo é perfeito!

Caso encontre algum erro, discorde de alguma informação ou tenha sugestões de melhoria, por favor, não hesite em nos informar. Sua participação é muito importante para nós.

Obrigad@!

Fonte: Adaptado de Santos (2017)

Figura 27 – Intervenção 2 (18/04/18)

Vamos começar?

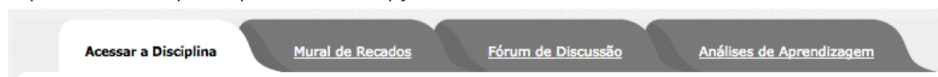
Iremos sugerir os conteúdos a serem estudados em cada semana, de forma que você chegue ao final dos 2 meses aprendendo todos os tópicos propostos. Porém, você pode estudar da maneira que desejar! Seguindo nossa sugestão, você poderá se organizar durante este período, se beneficiando também em sua disciplina presencial.

Para esta semana, sugerimos que você estude os seguintes tópicos e seus respectivos sub-tópicos:

- Da lógica à programação;
- Interpretadores e compiladores;
- Algoritmos;
- Sobre narrativas;
- Sobre pseudocódigos.

Acessando o [minicurso](#), você poderá estudar os conceitos, ver os exemplos e resolver os exercícios, além de consultar os materiais complementares.

Aproveite também para explorar as outras opções do sistema:



Acessar a Disciplina: você pode estudar todos os tópicos do minicurso

Mural de recados: você pode enviar e visualizar as mensagens recebidas

Fórum de discussão: você pode tirar dúvidas e colaborar com seus colegas

Análises de Aprendizagem: você pode acompanhar o andamento da sua aprendizagem.

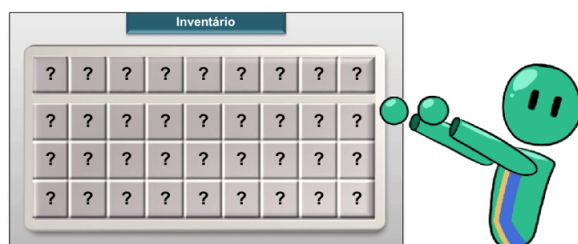
Até a próxima!

Fonte: Adaptado de Santos (2017)

Figura 28 – Intervenção 3 (20/04/18)

Hora de programar!!!

Para aprender Algoritmos, diferentes tópicos devem ser trabalhados. No nosso [minicurso](#), estes tópicos podem ser considerados **Coleções de Conhecimentos** para serem usados mais adiante.



Vamos conquistar estes conhecimentos? Sugerimos então o estudo do seguinte tópico:

O que é um programa?

- Estrutura de um programa (main, system, include);
- Constantes e comandos de atribuição;
- Comandos de entrada e saída (printf, scanf);
- Strings.

Até a próxima!

Fonte: Adaptado de Santos (2017)

Figura 29 – Intervenção 4 (22/04/18)

Ah, a lógica matemática e o raciocínio abstrato... Uhul!

Vamos entendê-los juntos! (:p) Para esta semana, veja este tópico:

DADOS, VARIÁVEIS E OPERADORES:

- O que são dados e para que servem;
- Como declarar variáveis;
- Operadores aritméticos, relacionais e lógicos.

Estes itens podem ser úteis para o seu aprendizado:



Colocamos uma atividade no Fórum de Discussão! Participe! Você tem até 25/04 para responder. Dia 26/04 daremos a resposta!

Bom estudo!

Fonte: Adaptado de Santos (2017)

Figura 30 – Intervenção 5 (25/04/18)

Muito bem, vamos colocar um pouco mais de emoção no nosso aprendizado!

Até agora, sugerimos os seguintes itens:



Com estes recursos, acreditamos que você já possa resolver alguns DESAFIOS! Estes desafios estão no Fórum de Discussão! Dia 29/04 daremos as respostas!

Boa sorte!

Fonte: Adaptado de Santos (2017)

Figura 31 – Intervenção 6 (29/04/18)

Ensinando o computador a tomar decisões!

A partir de agora você pode aprofundar seus conhecimentos ao estudar Estruturas de Controle. Para esta semana, sugerimos o tópico:

Estruturas de Controle - Decisão
(if, if-else, switch-case).

Nossa coleção está assim:



Compartilhe seus conhecimentos com os colegas ou ajude alguém com os assuntos que você domina! Para isso, use o Fórum de Discussão. Aproveite para tirar dúvidas dos assuntos que você tem dificuldade. Afinal, contrariando as leis da Matemática, o conhecimento é a única riqueza que dividimos mas não perdemos nada!

Bons estudos!

Fonte: Adaptado de Santos (2017)

Figura 32 – Intervenção 7 (02/05/18)

Está ficando cada vez mais interessante!

Deixe o computador trabalhar por você, As estruturas de repetição podem ajudar em diversas tarefas. Durante esta semana, sugerimos que você acesse o seguinte tópico:

Estruturas de Controle - Laços (repetição)
(while, do-while, for)

Veja o que conquistamos até agora:



Tenha uma ótima semana!

Fonte: Adaptado de Santos (2017)

Figura 33 – Intervenção 8 (06/05/18)

Queimando neurônios...!

Este é um tópico bem interessante! Então, vamos aproveitar ao máximo! :D

Com mais estes conhecimentos, você já tem um vasto conjunto de informações, que podem ajudar a resolver e entender a maioria dos problemas de lógica de programação! Para isso propomos o seguinte tópico:

Vetores e Matrizes

- Vetores
- Matrizes

Nosso inventário, neste momento, é este:



Bom estudo!

Fonte: Adaptado de Santos (2017)

Figura 34 – Intervenção 9 (09/05/18)

Mas já está acabando????

Amanhã, dia 10/05, é o último dia para acessar o ambiente de aula!

A partir de 11/05 a avaliação (prova) estará liberada juntamente com um questionário de satisfação, e o ambiente de aula estará bloqueado para acesso. Não perca, esta é a última oportunidade de estudar/revisar os conteúdos.

Ah, e ao final tem o CERTIFICADO e a possibilidade de validar as horas deste minicurso como ATIVIDADE COMPLEMENTAR, mas somente para quem realizar a avaliação e responder todo o Questionário de satisfação!

Participem!

Fonte: Adaptado de Santos (2017)

Figura 35 – Intervenção 10 (10/05/18)

Hoje é o ÚLTIMO DIA para acessar o ambiente de aula! Lembre-se de revisar todos os conceitos, fazer os exercícios, observar os links recomendados para você e os materiais complementares.

A partir de amanhã 11/05, a prova e o questionário de satisfação ficarão disponíveis até 14/05 às 23:59.

- Faça a prova e o questionário de satisfação para receber o seu certificado de conclusão.
- Não se esqueça que pode validar como atividade extra.

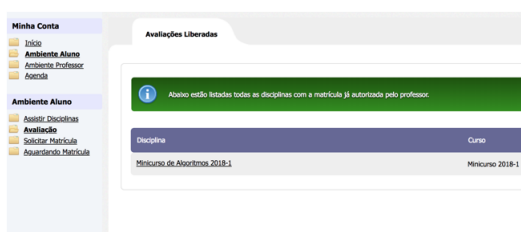
Esperamos ter ajudado. Agradecemos sua participação!

Fonte: Adaptado de Santos (2017)

Figura 36 – Intervenção 11 (11/05/18)

Começou o período de avaliação e de responder o questionário de satisfação!

Para realizar a **avaliação final**, acesse o menu lateral "Ambiente Aluno" do [AdaptWeb](#) e clique na opção Avaliação, conforme figura abaixo:



O link para acesso ao **questionário de satisfação** aparecerá ao final da correção da avaliação. Se preferir, é possível realizá-lo em outro momento (desde que seja completado até o dia 14/05).

Lembrando que é obrigatório realizar tanto a avaliação final quanto o questionário para **obter o certificado de participação**!

Boa avaliação!

Fonte: Adaptado de Santos (2017)

Figura 37 – Intervenção 12 (13/05/18)

Lembre-se, você só tem até amanhã para realizar a prova e responder o questionário de satisfação!
Todos que concluírem estes dois processos poderão receber certificado de participação!

Qualquer dúvida ou informação vocês podem entrar em contato com a professora Isabela Gasparini por email: isabela.gasparini@udesc.br ou diretamente em sua sala: Bloco F, 2o andar, sala 21.

Obrigad@ a tod@s pela participação!

Fonte: Adaptado de Santos (2017)

ANEXO C – DESAFIOS

Desafios extraídos de Santos (2017) para auxiliar nas intervenções do Minicurso de Algoritmos e Linguagem de Programação. Esses desafios foram postados no Fórum de Discussão do Ambiente AdaptWeb® durante o período de execução do Minicurso.

Desafio 1:

Qual é a resposta correta para as equações abaixo?

- a) $6 / 2 * (1 + 2) = ?$
- b) $7 + 8 * 0 - 2 = ?$
- c) $2 + 5 \times 3 + 4 = ?$
- d) $2 + 2 + 2 * 0 = ?$
- e) $7 + 7 / 7 + 7 * 7 - 7 = ?$
- f) $12 / 2 * (6 - 7 + 4) * 2 = ?$

Por quê?

Desafio 2:

Observe a sequência: 2, 5, 10, 17, 26, ...

Escreva um programa que forneça o n-ésimo elemento desta sequência.

Sabemos que não existe uma única forma de se resolver, e que cada pessoa pensa de maneira diferente. Compartilhe com os outros participantes, através do fórum de discussões, como você desvendou a sequência, e qual estratégia você usou para resolvê-la.

Desafio 3:

Crie uma calculadora de tempo. Por exemplo, calcular a diferença entre 3:15:33 (3 horas : 15 minutos : 33 segundos) e 17:28:22. A resposta também deve ser em horas:minutos:segundos.

Compartilhe através do fórum a sua solução, e qual estratégia usou para escrever o programa.

