UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ANTONIO FARIAS DE AZEVEDO SALES

UM SISTEMA DE RECOMENDAÇÃO PARA APRENDIZAGEM UBÍQUA NO CONTEXTO DA EDUCAÇÃO FORMAL E INFORMAL

MOSSORÓ – RN

2014

ANTONIO FARIAS DE AZEVEDO SALES

UM SISTEMA DE RECOMENDAÇÃO PARA APRENDIZAGEM UBÍQUA NO CONTEXTO DA EDUCAÇÃO FORMAL E INFORMAL

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação – associação ampla entre a Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semi-Árido, para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Francisco Milton Mendes Neto – UFERSA.

MOSSORÓ – RN

2014

Catalogação da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.

Sales, Antonio Farias de Azevedo.

Um sistema de recomendação para aprendizagem ubíqua no contexto da educação formal e informal. / Antonio Farias de Azevedo Sales. – Mossoró, RN, 2014

82 f.
Orientador(a): Prof. Dr. Francisco Milton Mendes Neto.

Dissertação (Mestrado em Ciência da Computação). Universidade do Estado do Rio Grande do Norte. Universidade Federal Rural do Semi-Árido. Programa de Pós-Graduação em Ciência da Computação.

1. Sistema de recomendação - Dissertação. 2. Ubiquidade - Dissertação. 3. Mobilidade - Dissertação. I. Mendes Neto, Francisco Milton. II. Universidade do Estado do Rio Grande do Norte. III.Título.

UERN/BC CDD 004.678

Bibliotecária: Elaine Paiva de Assunção CRB 15 / 492

ANTONIO FARIAS DE AZEVEDO SALES

UM SISTEMA DE RECOMENDAÇÃO PARA APRENDIZAGEM UBÍQUA NO CONTEXTO DA EDUCAÇÃO FORMAL E INFORMAL

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação para a obtenção do título de Mestre em Ciência da Computação.

APROVADA EM: 27 / 02 / 2014 .

BANCA EXAMINADORA

Prof. Dra. Cicilia Raquel Maia Leite – UERN
Presidente e Primeiro Membro

Prof. Dr. Francisco Milton Mendes Neto – UFERSA
Orientador e Segundo Membro

Prof. Dr. Gustavo Augusto Lima de Campos – UECE
Membro Externo

Aos meus pais Antonio Sales e Neumar Farias de Azevedo Sales, a minha irmã Neuzamar Farias de Sales, a minha sobrinha Luiza Maézia Farias Dantas, a minha namorada Wénneth Santos, a toda a minha família, aos meus professores, especialmente ao meu orientador Milton Mendes e aos meus amigos, pela confiança e apoio empregados na minha formação.

AGRADECIMENTOS

Agradeço à Deus por me permitir a vida, e por nela interferir nos momentos que mais preciso. Por ter me dado uma mente sã e um coração para guiá-la.

À todas as luzes que me iluminam, não me deixando faltar forças, me dando inspiração e coragem e sobretudo me protegendo nos momentos difíceis.

Aos meus pais Neumar Farias de Azevedo Sales e Antonio Sales por me conceberem e não medirem esforços durante toda a minha vida sacrificando as suas próprias em meu benefício.

À minha irmã Neuzamar Farias por se preocupar e ajudar a reerguer-me em todos os momentos de queda, para que eu pudesse novamente levantar, voltar a andar, correr e sonhar em voar.

À minha sobrinha Luiza Maézia por iluminar a minha vida e proporcionar-me tantos momentos de felicidade.

À Wénneth Santos, que me deu amor, compreensão e felicidade, e assim me abastece de motivos para lutar e continuar seguindo em frente na busca de novas conquistas.

Aos amigos que nesta jornada da vida conquistei e os guardo como uma de minhas maiores riquezas, e em especial que eu posso citar sem prejuízo dos demais: Marcelo Fernandes, Junior Silvio, Segundo Menezes, Leonardo da Vinci, Brito Filho, Robenildo Lima, Marcelo Amaral, Ferreira Junior, Lauro Gurgel, José Edson, Canindé de Freitas, Vinicius Amorim e Josenildo Amorim *(in memoriam)*.

Às luzes de minha vida: Gina Fernandes, Ludmila Amorim, Raquel Soares, Raylma Ferreira, Vanessa Carlos e Arlete Fernandes por me escutarem, me orientarem, e me apoiarem até mesmo em meus erros, e por iluminarem minha vida.

Em especial à Tatiane Morais por estar sempre ao meu lado, comprometendo até mesmo a paz de sua vida para não abrir mão de nossa amizade e de nossos momentos.

Aos amigos da faculdade: João Paulo, Pablo, Sairo, João Phelippe, Thiago, Ramon, Reuber, André, Ítalo, David, Ferdinand, Victor André, que convivemos juntos nos momentos felizes e em outros menos. Em especial a Luiz Junior por dividir o convívio e consequentemente os meus problemas, ajudando sempre que possível em tudo que está ao alcance, e a Laysa Mabel por estar sempre pronta para me ouvir, me apoiando e orientando meus passos.

À todos os professores, em especial ao meu orientador Prof. Dr. Sc. Francisco Milton Mendes Neto por dispor seu tempo e conhecimento para orientar este trabalho de dissertação,

mesmo com os obstáculos por mim enfrentados, não medindo compreensão e disposição para apoiar quando precisei.

À Universidade Federal Rural do Semi-Árido (UFERSA) e Universidade do Estado do Rio Grande do Norte (UERN), em particular ao Programa de Pós-Graduação em Ciência da Computação (PPGCC), pela oportunidade de realização desse curso.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pela concessão da bolsa de estudo, o que proporcionou o apoio necessário durante todo o curso.

Ao Governo Federal, por apoiar as instituições de ensino superior e pós, na construção de uma sociedade plena, com menos diferenças e com mais justiça.

E novamente a Deus por ter colocado todas estas pessoas na minha vida. Sem elas eu não seria eu, seria um outro, pois foi o convívio com todos eles, uns mais, outros menos, que construíram meu caráter, minha dignidade, minha mente e principalmente que faz com que meu coração bata neste ritmo tão alegre.

“Senhor, dai-me força para mudar o que pode ser mudado...

Resignação para aceitar o que não pode ser mudado...

E um Sistema de Recomendação para distinguir uma coisa da outra”.

Antonio Farias de Azevedo Sales

RESUMO

O ambiente de aprendizagem tradicional tem passado por uma grande evolução e, nesse sentido, o avanço tecnológico tem caminhado em conjunto, criando soluções e mecanismos, diminuído as distâncias e proporcionando oportunidades aos que antes eram limitados por fatores físicos ou econômicos. O Ensino a Distância (EaD) tem facilitado o acesso à informação e à formação, ganhando qualidade e, consequentemente, credibilidade. Por outro lado, a educação informal está se tornando uma grande aliada à formação, não só acadêmica, mas também social. Nesse sentido foi modelado e desenvolvido o sistema MobiLE+, uma solução para acesso e compartilhamento de conteúdo por dispositivos móveis, sendo sensível ao contexto do usuário e personalizado às suas características e históricos. O Sistema de Recomendação (SR), do sistema MobiLE+, é objeto dessa dissertação, e foi desenvolvido de forma a contemplar as características de ubiquidade e mobilidades, sendo dotado de um subsistema que considera o enriquecimento semântico e proporciona personalização do conteúdo ao usuário. O SR desenvolvido foi validado e apresentou características para o qual se propôs, tendo alcançando, portanto, êxito nos seus objetivos.

Palavras-Chave: Sistema de Recomendação, Ubiquidade, Mobilidade, MobiLE+.

ABSTRACT

Currently the traditional learning environment has undergone a major evolution, and in that sense the advent of technological advancement has walked together, creating solutions and mechanisms, decreased the distance and providing opportunities to those who were previously limited by physical or economic factors. The distance learning (DL) has facilitated access to information and training. It has lost its stigma and evolved quality and consequently gaining credibility. On the other hand, informal education is becoming a great ally to the formation not only academic, but also social. This effect was modeled and developed the Mobile+ system, which is proposed as a solution for accessing and sharing content for mobile devices, and context sensitive user and customized to their characteristics and features. The Recommendation System (RS) of MobiLE+, object of this dissertation, was developed to take account of the features ubiquity and mobility, endowed with a system that considers the semantic enrichment, and provides customization of content to the user. The RS was validated and had presented sufficient features for which it was proposed, and therefore successful in achieving project objectives.

Keywords: Recommendation System, Ubiquity, Mobility, MobiLE+.

LISTA DE TABELAS

Tabela 1 – Resultados possíveis de uma recomendação. 32
Tabela 2 – Métricas de performance de um sistema de recomendação. 32
Tabela 3 – Classes relevantes ao funcionamento do SRHU. 43
Tabela 4 – Dimensões e segmentos da amostra. 69
Tabela 5 – Distribuição do atributo idade. 70
Tabela 6 – Distribuição do atributo sexo. 70
Tabela 7 – Distribuição do atributo renda. 70
Tabela 8 – Distribuição das relações entre usuários. 71
Tabela 9 – Distribuição das relações entre usuário e conteúdo. 71
Tabela 10 – Distribuição das relações entre usuário e local. 71
Tabela 11 – Distribuição das relações entre conteúdo e local. 72
Tabela 12 – Resultado da simulação com contexto neutro e usuário de perfil definido. 73
Tabela 13 – Resultado da simulação com contexto definido e usuário de perfil neutro. 74
Tabela 14 – Resultado da simulação da lista de colaboradores. 74
Tabela 15 – Resultado da simulação da lista colaborativa ponderada. 74
Tabela 16 – Resultados da simulação da lista baseada em conteúdo. 75
Tabela 17 – Resultado da simulação da lista baseada na ubiquidade. 75

LISTA DE FIGURAS

Figura 1 – Exemplo da Estratégia de Reputação do Produto. 27
Figura 2 – Exemplo de Recomendação por Associação..... 27
Figura 3 – Exemplo de Análise das Sequências. 28
Figura 4 – Filtragem Híbrida 31
Figura 5 – Modelo de Componentes do Sistema MobiLE+ 42
Figura 6 – Sistema de Recomendação Híbrido Ubíquo. 54
Figura 7 – Fluxo de Entrada e Saída da Recomendação Colaborativa..... 55
Figura 8 – Fluxo de Entrada e Saída da Recomendação Baseada em Conteúdo..... 59
Figura 9 – Fluxo de Entrada e Saída da Recomendação Baseada na Ubiquidade..... 62
Figura 10 – Fluxo de Entrada e Saída da Recomendação Híbrida Ubíqua 64

LISTA DE EQUAÇÕES

Equação 1 – Cálculo do Índice de Semelhança de Usuários – ISU 57
Equação 2 – Cálculo do Índice de Semelhança de Conteúdo - ISC 58
Equação 3 – Cálculo do Índice de Semelhança Semântica entre Usuários 58
Equação 4 – Cálculo do Índice Baseado em Conteúdo – IBC 60
Equação 5 – Cálculo do Índice de Semelhança Semântica entre Usuário e Conteúdo 61
Equação 6 – Cálculo do Índice de Recomendação Ubíqua – IRU 63
Equação 7 – Cálculo do Índice de Recomendação Híbrido – IRH 64
Equação 8 – Cálculo do Peso da Recomendação Colaborativa 65
Equação 9 – Cálculo do Peso da Recomendação Baseada em Conteúdo 65
Equação 10 – Cálculo do Peso da Recomendação Ubíqua 65
Equação 11 – Cálculo do Número de Recomendações Possíveis 66

LISTA DE SIGLAS

AG – Algoritmo Genético;

API – *Application Programming Interface*;

CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior;

DC – *Dublin Core*;

EaD – Ensino a Distância;

FN – Falso Negativo;

FP – Falso Positivo;

GCM – *Google Cloud Messaging*;

IBC – Índice Baseado em Conteúdo;

IBGE – Instituto Brasileiro de Geografia e Estatística;

IDE – *Integrated Development Environment*;

IP – *Internet Protocol*;

IRH – Índice de Recomendação Híbrida;

IRU – Índice de Recomendação Ubíqua;

ISU – Índice de Semelhança de Usuário;

ISC – Índice de Semelhança Colaborativa;

JCP – *Java Community Process*;

EJB3 – *Enterprise Java Beans* 3;

JPA – *Java Persistence API*;

LBC – Lista Baseada em Conteúdo;

LBU – Lista Baseada na Ubiquidade;

LC – Lista de Colaboradores;

LCP – Lista Colaborativa Ponderada;

LES – Laboratório de Engenharia de Software;

LHU – Lista Híbrida Ubíqua;

LorSys – *Learning Object Recommender System;*

MAC – *Media Access Control*;

MILOS – *Multiagent Infrastructure for Learning Object Support*;

Moodle – *Modular Object-Oriented Dynamic Learning Environment*;

M-learning – Mobile Learning;

OBAA – Objeto de Aprendizagem Baseado em Agentes;

OA – Objeto de Aprendizagem;

OWL – *Web Ontology Language*;

PDA – *Personal Digital Assistant*;

PHP – *Hypertext Preprocessor*, originalmente *Personal Home Page*;

PPgCC – Programa de Pós-Graduação em Ciência da Computação;

RBC – Recomendação Baseada em Conteúdo;

RBU – Recomendação Baseada na Ubiquidade;

RDF – *Resource Description Framework*;

RHU – Recomendação Híbrida Ubíqua;

RI – Recuperação da Informação;

SCORM – *Sharable Content Object Reference Model*;

Scroll – *System for Capturing and Reminding of Learning Log;*

SES – Sistema de Enriquecimento Semântico;

SGBDOR – Sistema de Gerenciamento de Banco de Dados Objetod-Relacional;

SR – Sistema de Recomendação;

SRBC – Sistema de Recomendação Baseada em Conteúdo;

SRC – Sistema de Recomendação Colaborativo;

SRH – Sistema de Recomendação Híbrido;

SRHU – Sistema de Recomendação Híbrido Ubíquo;

SVN – Subversion;

VN – Verdadeiro Negativo;

VP – Verdadeiro Positivo;

TD – Traços Digitais;

UECE – Universidade Estadual do Ceará;

UERN – Universidade do Estado do Rio Grande do Norte;

UFERSA – Universidade Federal Rural do Semi-Árido;

UFPE – Universidade Federal do Pernambuco;

U-learning – Ubiquitous Learning;

XML – *eXtensible Markup Language*;

SUMÁRIO

1 INTRODUÇÃO 16
1.1 CONTEXTUALIZAÇÃO 16
1.2 PROBLEMÁTICA 17
1.3 OBJETIVO 18
1.4 METODOLOGIA E FERRAMENTAS 19
1.5 ORGANIZAÇÃO DA DISSERTAÇÃO 20

2 REFERENCIAL TEÓRICO 22
2.1 APRENDIZAGEM MÓVEL 22
2.2 AMBIENTES SENSÍVEIS AO CONTEXTO 23
2.3 APRENDIZAGEM UBÍQUA 23
2.4 APRENDIZAGEM INFORMAL 24
2.5 ENRIQUECIMENTO SEMÂNTICO 25
2.6 SISTEMAS DE RECOMENDAÇÃO 26
2.6.1 Estratégias de Recomendação 26
2.6.2 Sistema de Recomendação Baseada em Conteúdo 28
2.6.3 Sistema de Recomendação Colaborativa 29
2.6.4 Sistema de Recomendação Híbrida 30
2.6.5 Sistema de Recomendação Baseado em Outros Contextos 31
2.6.6 Avaliação de Sistemas de Recomendação 32
2.7 ALGORITMOS GENÉTICOS 33

3 TRABALHOS RELACIONADOS 35
3.1 RESUMO COMPARATIVO 37

4 A PLATAFORMA DO SISTEMA MOBILE+ 39
4.1 SUBPROJETOS DO SISTEMA 39
4.2 PROJETO DO SISTEMA 40
4.3 O ESCOPO DESSA DISSERTAÇÃO 43
4.3.1 O Sistema de Recomendação e a Base de Dados 43
4.3.2 O Sistema de Recomendação e o Enriquecimento Semântico 53

5 UM SISTEMA DE RECOMENDAÇÃO HÍBRIDO UBÍQUO 54

5.1 RECOMENDAÇÃO COLABORATIVA PONDERADA 55
5.1.1 Lista de Colaboradores 55
5.1.2 Lista Colaborativa Ponderada 57
5.1.3 Índice de Semelhança Semântica entre Usuários 58
5.2 RECOMENDAÇÃO BASEADA EM CONTEÚDO 59
5.2.1 Índice de Semelhança Semântica entre Usuário e Conteúdo 61
5.3 RECOMENDAÇÃO BASEADA NA UBIQUIDADE 61
5.4 RECOMENDAÇÃO HÍBRIDA 63
5.5 PERSONALIZAÇÃO AUTOMÁTICA DOS PESOS 65
5.6 RECOMENDAÇÃO CONJUNTA 65

6 VALIDAÇÃO DO SISTEMA 68
6.1 SIMULAÇÕES 68
6.2 AMOSTRA 69

7 ANALISE DOS RESULTADOS 73
7.1 RESULTADOS 73
7.2 ANÁLISE 75

8 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS 76

APÊNDICE A - ALGORITMOS EM DESCRIÇÃO NARRATIVA DO SRHU 82

1 INTRODUÇÃO

Este trabalho é parte integrante de um projeto desenvolvido pelo Grupo de Pesquisa do Laboratório de Engenharia de Software – LES/UFERSA. Este projeto tem como objetivo a modelagem e desenvolvimento de um sistema de discursão, compartilhamento e recomendação de conteúdo acadêmico – MobiLE+, para aprendizagem formal e informal levando em consideração características relevantes do contexto dos usuários.

A contribuição científica destacada do projeto MobiLE+ que corresponde a este trabalho de dissertação é a construção de um sistema de recomendação que considere características colaborativas, baseadas em conteúdo, ubíqua, e que considere o enriquecimento semântico de usuário e conteúdo, e além disso avalie a recomendação conjunta e ainda possua mecanismos de aprendizagem automática e personalizada para cada usuário, de acordo com suas escolhas.

1.1 CONTEXTUALIZAÇÃO

Os avanços das tecnologias móveis e de comunicação sem fio contribuíram para demarcar um novo horizonte aos processos de ensino e aprendizagem já estabelecidos. Os dispositivos móveis possibilitam o acesso a ambientes virtuais de ensino e aprendizagem, caracterizando, assim, a aprendizagem móvel, do inglês *Mobile Learning* ou *m-learning* (SACCOL, A., SCHLEMMER, E., BARBOSA, 2010).

A aprendizagem móvel permite que a aprendizagem ocorra a qualquer hora e em qualquer lugar. Porém, apesar de prover mobilidade, ela não fornece uma aprendizagem capaz de considerar informações sensíveis ao contexto do estudante (MANDULA et al., 2011). Ao considerarmos as características da aprendizagem móvel e a capacidade de detectar as informações de contexto do estudante, a fim de fornecer conteúdo educacional de forma personalizada, surge o conceito de Aprendizagem Ubíqua (também conhecida como *u-learning*, de *ubiquitous learning*) (SACCOL, A., SCHLEMMER, E., BARBOSA, 2010).

Este meio de oferecer ensino permite que estudantes e professores possam tirar vantagens dos recursos oferecidos pelas tecnologias móveis, dentre os quais se destaca a possibilidade de acessar, visualizar e prover conteúdo independentemente do horário e a partir de qualquer localidade (AHMED; CHANG, 2012).

Entretanto, as tecnologias móveis promovem o acesso a uma ampla quantidade e diversidade de conteúdo educacional, o que pode acarretar em perda de desempenho dos estudantes, por provocar a dispersão destes. Para lidar com essa situação, ao fornecer conteúdo educacional, deve-se levar em consideração as características particulares de cada estudante, como também as restrições dos seus dispositivos móveis, uma vez que os mesmos possuem recursos distintos e limitados.

A fim de desenvolver um ambiente de aprendizagem que melhor atenda às necessidades dos estudantes e que busque o aprimoramento do processo de ensino/aprendizagem, é fundamental considerar os aspectos relacionados a situações reais do cotidiano do estudante, quebrando as barreiras do formalismo educacional. A aprendizagem ubíqua pode auxiliar a educação informal fornecendo um meio favorável para autoaprendizagem e interações do estudante com o ambiente real, através de experiências obtidas pelas mídias sociais.

Modelos de aprendizagem modernos devem considerar experiências do mundo real. As redes sociais facilitam a obtenção de dados do mundo real, ao fornecer experiências reais dos seus usuários, experimentadas através de um meio digital (THAKKER et al., 2012).

Nesse contexto, um grupo de três estudantes de mestrado do Programa de Pós-graduação em Ciência da Computação da UFERSA/UERN está desenvolvendo o projeto “Um Modelo de Aprendizagem Ubíqua no Contexto da Educação Formal e Informal” (MobiLE+).

1.2 PROBLEMÁTICA

Dado o contexto apresentado, esse trabalho é responsável por criar um modelo de recomendação para aprendizagem ubíqua que seja capaz de recomendar recursos educacionais, através de informações obtidas do estudante e da sua relação com o ambiente real, de forma a apoiar aprendizagem formal e informal.

Desenvolver uma recomendação para o sistema MobiLE+, de acordo com as características desejadas apresenta vários desafios a serem vencidos, primeiramente por parte de algumas características do próprio andamento do projeto, e outras por características de suas funcionalidades. Citaremos as principais:

- Quanto ao Projeto:

a) O MobiLE+ é um sistema novo, ainda não funciona na prática, o que faz com que

suas características ainda não estejam completamente amadurecidas;

b) Ainda por ser novo, o Sistema MobiLE+ não apresenta uma base de dados real, o que dificulta a fase de testes e validação;
c) Outro fator que dificulta o trabalho é a documentação do sistema MobiLE+ que, por ainda está sendo finalizado, ainda apresenta limitações nesse aspecto;
d) A ubiquidade, fator importante no modelo, também terá de ter suas características simuladas;

- Quanto à solução:

a) O Sistema de Recomendação Híbrido Ubíquo (SRHU) apresenta também características colaborativas e baseada em conteúdo, as quais dependem de históricos de relações e acessos dos usuários, o que nesse caso também terá de ser testado e validado através de simulação;
b) No tocante ao modelo, a maior dificuldade é desenvolver uma solução que atenda aos critérios de uma recomendação híbrida, ou seja, com critérios colaborativos e baseados em conteúdo, que considere fatores do enriquecimento semântico e que seja sensível ao contexto do usuário, o que não é nada trivial para ser feito em conjunto;
c) Por fim, a solução deve apresentar uma resposta em tempo hábil, evitando assim que o contexto do usuário se altere antes da recomendação ser realizada;

1.3 OBJETIVO

De acordo com a problemática apresentada, o presente trabalho tem como objetivo modelar e desenvolver o SRHU, personalizado automaticamente para cada usuário, de conteúdos presentes na web[1], para o sistema MobiLE+, considerando as características do contexto onde o usuário está inserido.

[1] Conteúdos presentes na *Web* refere-se a qualquer página e seu conteúdo, vídeo, foto, hipertexto, formulários, desde que vinculado à um endereço link.

1.4 METODOLOGIA E FERRAMENTAS

Para alcançar os objetivos desse trabalho, foram estabelecidas uma série de atividades consideradas essenciais:

a) Realização de uma pesquisa bibliográfica sobre sistemas de recomendação, sobretudo os voltados a conteúdos educacionais;
b) Realização de uma pesquisa bibliográfica sobre os demais temas relacionados aos objetivos desse trabalho;
c) Realização de uma análise no modelo da estrutura e funcionamento do sistema MobiLE+;
d) Modelagem e implementação o SRHU de acordo com as especificações detalhadas nesse documento;
e) Modelagem e produção de uma base de dados simulada para servir de testes do SRHU, assim como para utilizar na validação final;
f) Validação do sistema com base na simulação desenvolvida;
g) Realização de uma análise dos resultados, enfatizando os pontos positivos e negativos da simulação; e
h) Documentação dos resultados obtidos comparando-os com as outras soluções disponíveis na literatura.

Para desenvolver o sistema e documentar este trabalho utilizamos as seguintes ferramentas e tecnologias:

Eclipse: é um IDE - *Integrated Development Environment* (Ambiente de Desenvolvimento Integrado) - para desenvolvimento Java, porém que suporta várias outras linguagens a partir de *plugins* como C/C++, PHP (*Hypertext Preprocessor)*, *ColdFusion*, Python, Scala e plataforma *Android*. Ele foi feito em Java e segue o modelo *open source* de desenvolvimento de software.

Microsoft Word: é o editor de texto proprietário da Microsoft e que possui ferramentas avançadas para controle de referências, índices, ortografia e concordância, além de permitir a visualização do documento paralelamente à sua produção.

Mendeley: Programa e *plugin* de gerenciamento de referências que pode ser integrado ao Microsoft Word.

Java: linguagem de programação orientada a objetos.

Protégé: é um software livre *open source,* para criar e editar ontologias e estruturar bases de conhecimentos. A plataforma Protégé suporta modelagem de ontologias por meio de um cliente web ou um cliente de *desktop* . Ontologias Protégé podem ser desenvolvidas em uma variedade de formatos, incluindo OWL (*Web Ontology Language* – Linguagem de Ontologia para *Web*), RDF (*Resource Description Framework*) e XML (*eXtensible Markup Language*) Schema.

O **PostgreSQL:** é um Sistema de Gerenciamento de Banco de Dados Objeto-Relacional (SGBDOR).

Subversion (SVN): é um sistema de controle de versão que gerencia arquivos e diretórios ao longo do tempo.

Java Persistence API (ou simplesmente **JPA**): é uma API (*Application Programming Interface*) padrão da linguagem Java para persistência de dados que deve ser implementada por *frameworks* que desejem seguir tal padrão.

1.5 ORGANIZAÇÃO DA DISSERTAÇÃO

Esse trabalho de dissertação está organizado em mais oito capítulos da seguinte forma:

O Capítulo 2 descreve a fundamentação teórica dos temas envolvidos nesse projeto, sendo eles: Aprendizagem Móvel, Ambientes Sensível ao Contexto, Aprendizagem Ubíqua, Aprendizagem Informal, Enriquecimento Semântico, Algoritmos Genéticos e, por fim, é dada uma ênfase especial aos Sistemas de Recomendação.

No Capítulo 3 é realizada uma pesquisa sobre os trabalhos relacionados ao tema dessa dissertação, elencando-se seus pontos fortes e fracos e comparando-os com o SRHU.

No Capítulo 4 é descrita uma visão geral do sistema MobiLE+, base desse projeto, e quais fatores são relevantes ao escopo desse trabalho.

Já o Capítulo 5 descreve a solução desenvolvida, todas as entradas e módulos que compõem o sistema, assim como os resultados, saídas, de cada um deles.

O Capítulo 6 descreve as características da simulação utilizada para validação da solução. As características da amostra são elencadas destacando-se os atributos relevantes considerados e os atributos livres. Além disso, também são descritos os tipos de casos simulados para análise.

No Capítulo 7 é realizada uma análise dos resultados da simulação, frisando os pontos positivos encontrados e as fragilidades a serem ainda vencidas.

Para finalizar, no Capítulo 8 temos as considerações finais e as descrições dos possíveis trabalhos que poderão dar continuidade ao projeto.

Além desses, ainda temos as Referências e o Apêndice A, que contém os algoritmos em descrição narrativa das classes que realizam as soluções da recomendação.

2 REFERENCIAL TEÓRICO

Neste capítulo é apresentada a fundamentação teórica sobre as áreas relacionadas ao escopo deste trabalho, sejam elas: Aprendizagem Móvel, Ambiente Sensível ao Contexto, Aprendizagem Ubíqua, Aprendizagem Informal, Enriquecimento Semântico, Sistemas de Recomendação e Algoritmos Genéticos.

2.1 APRENDIZAGEM MÓVEL

A aprendizagem móvel refere-se ao uso de dispositivos móveis, tais como computadores de bolso, *Personal Digital Assistant* (PDA) ou telefones celulares, em atividades de aprendizagem realizadas a partir de qualquer lugar e a qualquer hora, trazendo informação e conhecimento para uma situação e lugar onde a atividade de aprendizagem acontece (CASTILLO; AYALA, 2008).

Já Yau e Joy, (2011) definem a aprendizagem móvel de uma perspectiva pedagógica, sendo considerada como tal qualquer tipo de aprendizagem que acontece quando o estudante tira vantagem de oportunidades de aprendizagem oferecidas por tecnologias móveis.

De acordo com a primeira parte da definição, aprendizagem móvel não necessariamente implica o uso de dispositivos móveis, sendo suficiente apenas que a aprendizagem não aconteça em um local fixo. Nesse sentido, os autores consideram a mobilidade do estudante, e não a tecnologia móvel em si.

O presente trabalho considera a abordagem apresentada por Castilho e Ayala (2008). De acordo com estes autores, um ambiente de aprendizagem móvel implica o uso de dispositivos móveis computacionais para suportar atividades de aprendizagem, independentemente de local e horário. Nestes ambientes, os dispositivos móveis devem (i) desempenhar o papel de mediadores entre os estudantes, (ii) suportar essa interação e (iii) prover informação e recursos de conhecimento com base nas capacidades e nos interesses dos estudantes.

2.2 AMBIENTES SENSÍVEIS AO CONTEXTO

Sensibilidade ao contexto descreve um paradigma no qual o contexto de um usuário é levado em consideração para definir o seu perfil (MOORE; HU, 2010). Não existe um consenso a respeito da definição de "contexto", sendo este específico da aplicação e da intenção desejadas, requerendo a identificação das funções e propriedades dos domínios dos indivíduos (MOORE; HU, 2010). Não há como definir de forma precisa o que seria o contexto. Dessa forma, as características que serão consideradas para construção do contexto dependem da situação específica.

O contexto pode ser definido de acordo com informações relativas a propriedades que se combinam para descrever e caracterizar uma entidade e seu papel de uma forma legível pelo computador (MOORE; HU, 2010). A localização do estudante, por exemplo, é uma característica importante para a definição do seu contexto em um ambiente para aprendizagem móvel. Entretanto, o contexto inclui mais do que apenas a localização. De fato, quase todas as informações disponíveis no momento da interação podem ser vistas como informações contextuais, dentre as quais destacam-se (MOORE; HU, 2010):

- As diversas tarefas realizadas pelos usuários no sistema;
- A variada gama de dispositivos que se combinam para criar sistemas móveis, com a infraestrutura de serviços associada;
- Os recursos disponíveis (ex.: condição da bateria, tamanho de tela etc.);
- A situação física (ex.: nível de ruído, temperatura, nível de luminosidade etc.);
- A informação espacial (ex. localização, velocidade, orientação etc.);
- A informação temporal (ex. hora do dia, data etc.).

2.3 APRENDIZAGEM UBÍQUA

Como é possível perceber pela definição de aprendizagem móvel, a principal vantagem dessa forma de EaD é o fato de permitir que a aprendizagem ocorra a qualquer hora e em qualquer localidade. Porém, apesar de prover mobilidade, ela não fornece uma aprendizagem

que seja sensível ao contexto do estudante (MANDULA et al., 2011).

Reunindo as definições de aprendizagem móvel e de sensibilidade ao contexto, surge o conceito de Aprendizagem Ubíqua (também conhecida como *u-learning*, de *ubiquitous learning*). Ou seja, a aprendizagem ubíqua pode ser considerada como sendo a aprendizagem móvel que é realizada levando-se em consideração as características do contexto dos estudantes, provendo a estes conteúdos adaptados às suas necessidades (MANDULA et al., 2011). Assim, a aprendizagem ubíqua provê um novo paradigma, através do uso de dispositivos móveis, que provê um serviço de forma transparente aos estudantes (LOUREIRO et al., 2009).

A mobilidade permitida ao usuário devido ao uso de dispositivos móveis em aplicações ubíquas torna ainda mais importante a consideração do contexto de um estudante, visto que as características do estudante podem se modificar a qualquer momento. Essas modificações podem ocorrer em diversos aspectos (ex. condições físicas, recursos físicos disponíveis, recursos computacionais etc.) (LOUREIRO, et al., 2009).

2.4 APRENDIZAGEM INFORMAL

Os ambientes de aprendizagem podem ser classificados em dois tipos: formal e informal. A aprendizagem formal ocorre sob a supervisão de um professor, enquanto que na aprendizagem informal o aluno é responsável pelo seu aprendizado, utilizando-se de informações cotidianas do seu ambiente (WANG; SHEN, 2012). Portanto, a aprendizagem informal ocorre em todas as fases da vida, tendo como principais características (JIUGEN et al., 2011):

- Autonomia: o estudante é responsável por sua aprendizagem, onde ele determina o objetivo a ser alcançado, o conteúdo a ser abordado e como este será processado.
- Conhecimento: obtido por meio das interações sociais e profissionais.
- Diversidade: a fonte de informação se apresenta de várias formas, sem que seja necessário um local determinado. A aquisição de conhecimento pode ser obtida através de experiências diárias, recursos do ambiente, bibliotecas, redes sociais, dentre outros.

As tecnologias móveis, quando utilizadas em ambientes de aprendizagem formal, reforçam a ligação entre o aluno, o conteúdo e o professor, fornecendo estímulo e motivação. Neste cenário, os educadores têm o desafio de elaborar conteúdos educativos, para os dispositivos móveis dos alunos, com o objetivo de maximizar os resultados da aprendizagem.

2.5 ENRIQUECIMENTO SEMÂNTICO

O notório crescimento das mídias sociais proporciona o incremento de imensa quantidade de conteúdo gerado por seus usuários. Esses conteúdos têm influenciado radicalmente as técnicas aplicadas em muitas áreas.

Segundo Thakker et al. (2012), tendências apontam que a mídia social irá ter um forte impacto sobre a aprendizagem informal, fornecendo conteúdo gerado pelo usuário com o objetivo de auxiliar o processo de aprendizagem. Redes sociais podem oferecer diversas experiências reais:

- As pessoas podem comentar sobre experiências com pessoal, serviços ou lugares (*shoppings*, hotéis, praias);
- Compartilhar histórias (*blogs*);
- Comentar situações vivenciadas (assistir vídeo, ler livro).

A essas experiências reais experimentadas através das mídias sociais dá-se o nome de Traços Digitais (TDs). Quando selecionados com cuidado, os TDs possuem grande potencial para a aprendizagem informal. Ambientes de aprendizagem modernos podem explorar conteúdo gerado pelo usuário para proporcionar situações relacionadas com sua experiência do mundo real.

Os conteúdos gerados pelas redes sociais, geralmente, não possuem o domínio de aprendizagem bem definido, podendo apresentar múltiplas interpretações. No entanto, têm surgido novas arquiteturas para criar, agregar, organizar e explorar os TDs em situações de aprendizagem informal (THAKKER et al., 2012).

O enriquecimento semântico possibilita estender a compreensão do domínio de um determinado TD. Para isso, são realizados a identificação e o enriquecimento das formas superficiais (palavras-chave) a partir do conteúdo textual (comentários), considerando palavras relacionadas. As formas superficiais enriquecidas podem ser semanticamente associadas a conceitos oriundos de uma ontologia, a fim de realizar uma marcação semântica (DESPOTAKIS et al., 2011).

2.6 SISTEMAS DE RECOMENDAÇÃO

De acordo com Vieira e Nunes (2012), o aumento de meios de disponibilização de conteúdo, podendo ser produtos ou informação, através de sistemas Web, provoca uma situação onde o usuário possui muitas opções de escolha antes mesmo de estar apto a selecionar uma opção que atenda suas necessidades. Os SRs buscam amenizar os impactos gerados por essa sobrecarga de informação.

Os SRs utilizam repositórios de informação e dados de preferência dos usuários para direcionar conteúdos aos indivíduos com potenciais interesses. Um dos desafios dos SRs é realizar a indicação de produtos, serviços e/ou informação que melhor atendam as expectativas dos usuários (e ao seu perfil) (CAZELLA et al., 2010).

Segundo Vieira e Nunes (2012), os SRs podem ser classificados em três tipos: Sistema de Recomendação Baseada em Conteúdo (SRBC), Sistema de Recomendação Colaborativa (SRC) e Sistema de Recomendação Híbrida (SRH). A recomendação baseada em conteúdo é realizada através de informações recuperadas do usuário (perfil, comportamento) e/ou do seu histórico de escolhas. Já a recomendação por filtragem colaborativa leva em consideração escolhas realizadas por usuários com características similares. Por último, o SRH efetua a recomendação através da aplicação das técnicas utilizadas pelos SRBC e SRC, de forma a atingir um maior número de possibilidades e sugerir objetos que diretamente não aparentam estar relacionados (MARTINS et al., 2011).

A classificação tradicional dos SRs não exclui outras possibilidades de classificação, sob outras características ou técnicas, aplicadas da maneira mais adequada para solucionar um caso específico. As subseções seguintes descrevem algumas das técnicas e estratégias de mais destaque utilizadas na literatura.

2.6.1 Estratégias de Recomendação

Diferentes estratégias são usadas para personalizar uma recomendação para um usuário, sempre com o objetivo de recomendar algo que realmente atenda suas expectativas. Entre as principais estratégias estão:

- Reputação de Produto: É uma estratégia bastante utilizada em quase todas as áreas, e consiste na avaliação do produto ou item por parte dos usuários que o conheceram. Avaliações de clientes são muito úteis para assegurar outros consumidores da qualidade e utilidade dos produtos comercializados (CAZELLA et al., 2010). A Figura 1 exemplifica o uso da estratégia de reputação do produto, por meio da visualização das avalições de outros usuários.

Figura 1 – Exemplo da Estratégia de Reputação do Produto.
Fonte: www.americanas.com (adaptado pelo autor).

- Recomendação por Associação: Este tipo de recomendação é obtido através de técnicas capazes de encontrar em uma base de dados associações entre itens avaliados por usuários (comprados, lidos e outros) (CAZELLA et al., 2010). Muito comum em *sites* de comércio eletrônico, como mostra a Figura 2.

Figura 2 – Exemplo de Recomendação por Associação
Fonte: www.submarino.com (Adaptada pelo autor).

No exemplo ilustrado na Figura 2, a partir da seleção de um livro, outros livros relacionados são apresentados.

- Análise das Sequências de Ações: Na *web*, sequências podem ser utilizadas para capturar o comportamento de usuários através de históricos de atividade temporal, como em *weblogs* [2] e histórico de compras de clientes (CAZELLA et al., 2010). As informações

[2] O termo blog vem de weblog, contração de web e log, criado por John Barger com o pioneiro espaço Robot Wisdom, em 1997(BOSSLER; VENTURELLI).

encontradas nestas sequências temporais podem ser empregadas nas identificações de padrões de navegação e consumo, que em seguida podem servir aos SRs.

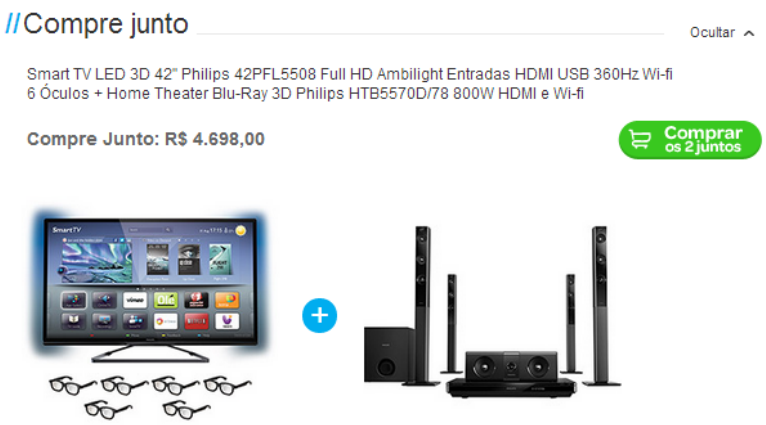

Figura 3 – Exemplo de Análise das Sequências.
Fonte: www.submarino.com (Adaptada pelo autor).

No exemplo ilustrado na Figura 3, a partir da seleção de um equipamento eletrônico, outro equipamento é apresentado com base nos históricos de outros usuários.

2.6.2 Sistema de Recomendação Baseada em Conteúdo

A abordagem baseada em conteúdo tem suas raízes na área de recuperação de informação. Devido aos significativos avanços feitos pelas comunidades de filtragem de informação e filtragem de conteúdo, muitos sistemas baseados em filtragem de conteúdo focam na recomendação de itens com informações textuais, como documentos e *sites* (CAZELLA et al., 2010).

Desta forma, a filtragem baseada em conteúdo parte do princípio de que os usuários tendem a interessar-se por itens similares aos que demonstraram interesse no passado, definindo então a similaridade entre os itens (HERLOCKER, 2000).

Em muitos casos, verificar esta similaridade pode não ser muito trivial. Por exemplo, para que seja estabelecida a similaridade entre itens como roupas e brinquedos, talvez seja necessária a identificação dos atributos nos itens a serem comparados (peso, preço ou marca) (ACOSTA; REATEGUI, 2012). Já se os itens são documentos textuais, este processo de comparação pode ser facilitado, por exemplo, pela análise de termos em comum.

Outros exemplos de filtragem baseada em conteúdo são: sistemas de recuperação booleanos, onde a consulta é um conjunto de palavras-chave combinadas com operadores booleanos; sistemas de filtragem probabilística, onde o raciocínio probabilístico é utilizado para determinar a probabilidade que um documento possui de atender as necessidades de informação de um usuário; e processamento de linguagem natural (HERLOCKER, 2000).

Os Sistemas de Recomendação Baseada em Conteúdo apresentam duas limitações que podem comprometer a solução:

a) Os dados do conteúdo devem ser bem estruturados, do contrário o acesso e a análise das informações para comparação tornam-se difíceis de serem realizados. Em conteúdo de áudio e vídeo essa tarefa é muito mais complexa do que em documentos de texto, por exemplo;
b) Um sistema de recomendação que baseia-se no histórico do usuário pode tornar-se superespecialista, uma vez que recomenda apenas conteúdos bem avaliados pelo usuário.

2.6.3 Sistema de Recomendação Colaborativa

A Filtragem Colaborativa se diferencia da filtragem baseada em conteúdo exatamente por não exigir a compreensão ou reconhecimento do conteúdo dos itens, atendendo pontos que ficaram em aberto na abordagem baseada em conteúdo (ACOSTA; REATEGUI, 2012).

Nos sistemas colaborativos, a essência está na troca de experiências entre as pessoas que possuem interesses comuns. Nestes sistemas, os itens são filtrados com base nas avaliações feitas pelos usuários (CAZELLA et al., 2010). Dessa forma, os sistemas colaborativos partem do princípio que usuários semelhantes irão gostar de itens semelhantes.

Esta pontuação permite ao usuário descobrir itens que são considerados de interesse pelo grupo e evitar os itens que são considerados de pouco interesse. Sistemas mais avançados descobrem de maneira automática relações entre usuários (vizinhos mais próximos), com base na descoberta de padrões comuns de comportamento (CAZELLA et al., 2010).

A filtragem colaborativa apresenta algumas vantagens como, por exemplo, a possibilidade de apresentar aos usuários recomendações inesperadas, ou seja, o usuário poderia receber recomendações de itens que não estavam sendo pesquisados de forma ativa (ACOSTA;

REATEGUI, 2012).

Segundo Cazella et al. (2010), a técnica de Filtragem Colaborativa segue basicamente três passos:

1. Calcular o peso de cada usuário em relação à similaridade ao usuário alvo (métrica de similaridade);

2. Selecionar um subconjunto de usuários com maiores similaridades (vizinhos) para considerar na predição;

3. Normalizar as avaliações e computar as predições ponderando as avaliações dos vizinhos com seus pesos.

Os sistemas de recomendação colaborativa também apresentam algumas limitações, sobretudo relativas à coleta de dados:

1. Quando um novo item é adicionado ao sistema, ele não terá avaliação de outros usuários anteriores para que seja recomendado a outro;
2. Se o número de usuários for pequeno em relação ao número de conteúdo, corre-se o risco das pontuações dos itens tornarem esparsas;
3. Se o usuário for muito diferente dos demais, não encontrando assim muitos semelhantes, suas recomendações podem tornar-se pobres.

2.6.4 Sistema de Recomendação Híbrida

A Filtragem Híbrida emprega técnicas advindas tanto da abordagem de Filtragem Colaborativa quanto da Filtragem Baseada em Conteúdo. A ideia é baseada na busca de itens de acordo com os perfis de interesse dos usuários, mas considerando também a similaridade de conteúdo entre os itens (ACOSTA; REATEGUI, 2012). Ao integrar ambas as técnicas, a Filtragem Híbrida busca incorporar as vantagens tanto da Filtragem Colaborativa quanto da Filtragem baseada em Conteúdo, ao mesmo tempo em que minimiza possíveis limitações de uma ou outra abordagem (HERLOCKER, 2000).

Essa abordagem é constituída de vantagens proporcionadas pela filtragem baseada em conteúdo e pela filtragem colaborativa, unindo o melhor das duas técnicas e eliminando as fraquezas individuais de cada uma, conforme apresentado pela Figura 4.

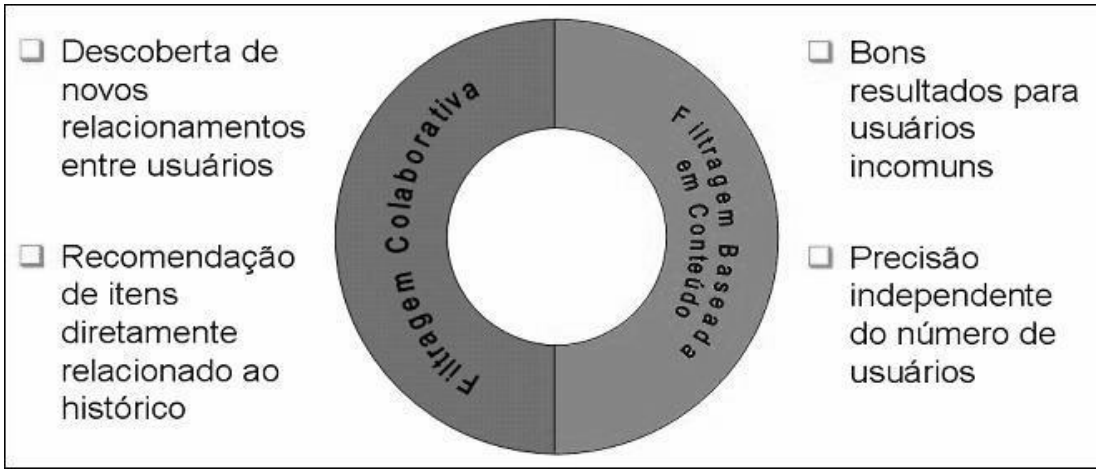

Figura 4 – Filtragem Híbrida
Fonte: Cazella, 2010.

2.6.5 Sistema de Recomendação Baseado em Outros Contextos

Humanos, quando tomam decisões, recomendam e personalizam informações, produtos e serviços para outros. Além de usar as informações convencionais utilizadas pelos sistemas computacionais tendem, também, a usar informações adicionais relacionadas a habilidades sociais e psicológicas humanas, tais como traços de personalidade e emoção (CAZELLA et al., 2010).

Para um sistema de computador atingir a mínima compreensão do modelo de como os humanos atuam nas situações em que realizam uma recomendação no ambiente real é uma tarefa que exige um certo grau de complexidade.

Usualmente, sistemas computacionais usam informações como competências, preferências, informações demográficas dos usuários, entre outras, para a tomada de decisão e posterior recomendação de uma informação, produto ou serviço.

Mesmo sabendo ser impossível perfeitamente antecipar as necessidades humanas individuais para recomendar o produto certo, sabe-se que quanto mais ricas forem as informações sobre o usuário, mais precisos serão os produtos, serviços e /ou pessoas recomendadas, bastando observar o que ocorre na vida real (CAZELLA et al., 2010).

2.6.6 Avaliação de Sistemas de Recomendação

Avaliar a performance de algoritmos de recomendação consiste essencialmente em avaliar a satisfação do utilizador. Tipicamente estamos interessados em perceber o grau de aceitação das recomendações, ou seja, quantificar o número de vezes que os utilizadores aceitam ou rejeitam itens recomendados (REIS, 2012).

Neste sentido, a Tabela 1 apresenta os métodos tradicionais de avaliação da recomendação, resumindo de forma simplificada três métricas mais comuns:

Tabela 1 – Resultados possíveis de uma recomendação.

	Recomendados	Não Recomendados
Preferidos	Verdadeiro-Positivo (VP)	Falso-Negativo (FN)
Não Preferidos	Falso-Positivo (FP)	Verdadeiro-Negativo (VN)

Fonte: Reis, 2012.

Na Tabela 2, conta-se então o número de ocorrências que se enquadram em cada uma das categorias, o que servirá para calcular as seguintes métricas:

Tabela 2 – Métricas de performance de um sistema de recomendação.

Métrica	Expressão	Interpretação
Precision	$\frac{\boldsymbol{nº\ de\ VP}}{\boldsymbol{nº\ de\ VP + nº\ de\ FP}}$	Mede a probabilidade de um item recomendado ser relevante.
Recall	$\frac{\boldsymbol{nº\ de\ VP}}{\boldsymbol{nº\ de\ VP + nº\ de\ FN}}$	Mede a probabilidade de um item relevante ser recomendado.
False Positive Rate	$\frac{\boldsymbol{nº\ de\ FP}}{\boldsymbol{nº\ de\ FP + nº\ de\ VN}}$	Mede a probabilidade de um item ser mal recomendado.
F-measure	$\boldsymbol{2x\frac{Precision + Recall}{Precison\ x\ Recall}}$	Combina as características do *Precision* e do *Recall* numa métrica conjunta.

Fonte: Reis, 2012.

Outras métricas também são relevantes que podem medir características determinadas por propriedades inerentes a cada sistema.

2.7 ALGORITMOS GENÉTICOS

Segundo Heinen e Osório (2006), os Algoritmos Genéticos (AG) são métodos de busca estocástica baseados na Teoria da Evolução Natural das Espécies, criados por John Holland nos anos 60 (HOLLAND, 1975). Os AGs trabalham com uma população de soluções iniciais, chamadas cromossomos, que através de diversas operações vão sendo evoluídas até que se chegue a uma solução que melhor atenda a algum critério específico de avaliação (MITCHELL, 1998). Para que isto ocorra, a cada geração, os cromossomos são avaliados segundo uma função que mede o seu nível de aptidão, chamada de função de *fitness*. Os cromossomos que tiverem o melhor *fitness* são selecionados para darem origem à próxima geração, através de operações como cruzamentos e mutações. Desta forma, a tendência é que a cada geração o conjunto de soluções vá sendo melhorado, até que se chegue a uma solução que atenda aos objetivos desejados (GOLDBERG; HOLLAND, 1988).

Algoritmos Genéticos (AGs) são particularmente aplicados em problemas complexos de otimização: problemas com diversos parâmetros ou características que precisam ser combinadas em busca da melhor solução; problemas com muitas restrições ou condições que não podem ser representadas matematicamente; e problemas com grandes espaços de busca. AGs têm sido aplicados a diversos problemas de otimização, tais como: Otimização de Funções Matemáticas, Otimização Combinatorial, Otimização de Planejamento, Problema do Caixeiro Viajante, Problema de Otimização de Rota de Veículos, Otimização de Layout de Circuitos, Otimização de Distribuição, Otimização em Negócios e Síntese de Circuitos Eletrônicos (PACHECO, 1999).

Os principais elementos de um AG são:

a) Gene: é a representação básica de uma característica;
b) Cromossomo (indivíduo): É um conjunto de genes que representam um indivíduo;
c) Geração: É todo o conjunto de indivíduos de uma determinada iteração;
d) População Inicial: É a primeira geração do sistema, geralmente criada de forma aleatória;
e) Função *Fitness*: É a função que avalia um cromossomo ou indivíduo, considerando os critérios de acordo com as características do problema.

Segundo LUCAS (2002), basicamente, o que um algoritmo genético faz é criar uma população de possíveis respostas para o problema a ser tratado (inicialização) para depois submetê-la ao processo de evolução, constituído pelas seguintes etapas:

a) Avaliação: avalia-se a aptidão das soluções (indivíduos da população) – feita uma análise para que se estabeleça quão bem elas respondem ao problema proposto;
b) Seleção: indivíduos são selecionados para a reprodução. A probabilidade de uma dada solução “i” ser selecionada é proporcional à sua aptidão;
c) Cruzamento: características das soluções escolhidas são recombinadas gerando novos indivíduos que irão compor uma nova geração;
d) Atualização: os indivíduos criados nesta geração são inseridos na população;
e) Finalização: verifica-se se as condições de encerramento da evolução foram atingidas retornando para etapa de avaliação em caso negativo e encerrando a execução em caso positivo.

3 TRABALHOS RELACIONADOS

Na literatura foram encontrados alguns trabalhos relacionados ao tema dessa pesquisa, entretanto nenhum deles aborda todos os pontos considerados no trabalho.

LI *et al*., (2012) desenvolveram o sistema SCROLL (*System for Capturing and Reminding of Learning Log*). Este sistema proporciona aos usuários registrarem suas experiências de aprendizagem, associando a elas fotos, áudio, vídeos, localização, dados de sensores e outras informações. Apesar de utilizar captura, armazenagem e organização de conteúdo de forma ubíqua, o mesmo não possui um sistema de recomendação sensível ao contexto, que possa disponibilizar essas informações direcionando-a aos usuários mais adequados.

Gluz e Vicari (2010) propuseram a MILOS (*Multiagent Infrastructure for Learning Object Support*), uma infraestrutura, combinando ontologias e agentes, que implementa as funcionalidades necessárias aos processos de autoria, gerência, busca e disponibilização de Objetos de Aprendizagem (OAs) compatíveis com a proposta de padrão de metadados de OAs denominada OBAA (Objetos de Aprendizagem Baseados em Agentes). Entretanto esse sistema não possui um sistema de recomendação propriamente, mas apenas um sistema de busca que utiliza-se de metadados para fornecer o resultado.

Já Ferreira e Raabe, (2010) apresentam o LorSys (*Learning Object Recommender System*), um sistema de recomendação de OAs, no formato SCORM (*Sharable Content Object Reference Model*), para o Ambiente Sophia, que é uma personalização do Moodle (*Modular Object-Oriented Dynamic Learning Environment*). O LorSys utiliza a técnica de recomendação híbrida, aplicada a informações de preferências dos estudantes e dados de acesso aos OAs. Apesar disso, tem como ponto fracos o fato de recomendar apenas OAs padronizados e não ser sensível ao contexto do usuário.

Trevisan et al. (2012), aplicou traços de personalidade em SRs com base em contextos focado em TV Digital podendo ser aplicado como solução no processo de ensino-aprendizagem, uma vez que considera questões específicas referentes a traços de personalidade do usuário/aluno. Além disso, utiliza-se de informações contextuais que auxiliam na recuperação de programas que mais se adaptem ao contexto no qual o usuário está inserido. Apesar de utilizar os traços de personalidade, estes são obtidos por meio de preenchimento de formulário, muitas vezes longos (até 300 questões), o que provoca o risco do não preenchimento

pelos usuários. Além disso, o contexto citado não considera a mobilidade, tornando-o assim frágil neste quesito.

Um framework para a recomendação de artigos científicos é apresentado por Nunes e Borges (2006) *apud* Barcellos et al. (2007). O sistema considera a opinião dos usuários e utiliza a técnica de filtragem colaborativa. Os usuários recebem recomendações de artigos com base nas informações de seu perfil. Apesar do sistema realizar uma busca por informações complementares no currículo Lattes do usuário, os dados do perfil do usuário são adquiridos de forma explícita, via questionário, com várias perguntas sobre a sua carreira acadêmica. Umas das vantagens deste trabalho é o uso da opinião dos outros usuários, entretanto o ponto fraco é o preenchimento do formulário.

Ávila (2008) apresenta o SisRecAC, que tem por objetivo encontrar na *Web*, através de uma meta-busca, artigos científicos que sejam de interesse de um usuário através de um texto exemplo. Uma Consulta por Exemplo é a possibilidade de recuperação de informações a partir de um texto exemplo informado pelo usuário e não a partir de palavras-chave, o que é, atualmente, o padrão para busca de informações. Os artigos localizados e recomendados pela ferramenta de meta-busca passam por uma avaliação do próprio usuário, que pode comentá-los e recomendá-los a outras pessoas, criando uma rede de cooperação. O sistema apresentou-se bastante eficaz, entretanto como sistema de busca e não de recomendação.

Em Nunes Borges (2006), é apresentado um SR para apoio à colaboração. Este trabalho apresenta uma técnica que identifica o perfil do usuário através das suas publicações. As recomendações são feitas a partir da análise do comportamento dos usuários em um *Web Chat* e do acesso a uma biblioteca digital. Entretanto o mesmo não apresenta características de sensibilidade ao contexto ou de mobilidade.

Lopes (2007) propõe um sistema de recomendação acadêmico para bibliotecas digitais cujo perfil do usuário é coletado de forma implícita. Os dados utilizados como fonte para a tarefa de recomendação consistem de: (i) informação do usuário, obtidas a partir do curriculum Lattes em XML; e (ii) informações sobre os documentos digitais obtidas através de metadados no formato Dublin Core (DC) codificados em XML. Apresenta como vantagem a captura implícita dos dados do usuário, entretanto também não possui suporte à mobilidade ou à sensibilidade ao contexto.

Barcellos et al. (2007) apresenta um modelo de referência para sistema de recomendação que dispensa a intervenção do usuário no momento de traçar seu perfil e utiliza técnicas de mineração de dados. O modelo foi implementado no protótipo “Ponto com Filtro”

para verificar sua aplicabilidade. Entretanto não considera as questões de ubiquidade ou mobilidade do usuário.

Neves e Brito (2012) oferecem um mecanismo que utiliza técnicas de Recuperação da Informação (RI) para buscar artigos científicos que possuam certo grau de similaridade com o conteúdo (materiais didáticos) de uma rede social acadêmica. Utilizando as informações do conteúdo da rede (título, descrição e *tags*), são aplicadas técnicas de RI para verificar o grau de similaridade entre os artigos e os materiais. Esses artigos são ordenados por maior grau de similaridade e podem ser acessados na página pública do material didático em que está vinculado. O sistema caracteriza-se pela recuperação de informação com base em conteúdo, mas não considera a sensibilidade ao contexto e não utiliza técnicas que possam personalizar a sugestão pelo perfil do usuário.

Acosta e Reategui (2012), apresentou um trabalho com foco na ampliação da ferramenta SMILE por meio da integração de um sistema de recomendação de conteúdo a esta, com o objetivo de facilitar os processos investigativos e ampliar a reflexão dos estudantes com relação aos temas abordados. As recomendações de conteúdo são baseadas em um texto selecionado pelo professor que determina o tema de uma atividade. Estas recomendações podem ser desde textos que auxiliem na elaboração de questionamentos, imagens, vídeos ou até autores que escrevem sobre tema proposto. Mas uma vez o sistema não apresenta sensibilidade ao contexto ou mobilidade, assim como neste caso também não personaliza o conteúdo recomendado de acordo com o perfil do usuário.

Costa (2013), apresentou um trabalho que tem como objetivo a concepção, implementação e apresentação de um modelo semântico de personalização para recomendação de conteúdo ao usuário de acordo com suas preferências e necessidades. O modelo é estruturado e beneficiado pela interoperabilidade, a qual expande o espaço de busca para bases de dados heterogêneas, permitindo a recomendação dinâmica de conteúdo em um maior volume de dados e garantindo maior eficiência à personalização. Entretanto também não apresenta suporte à mobilidade e à ubiquidade.

3.1 RESUMO COMPARATIVO

Conforme observa-se nos trabalhos relacionados, o perfil do usuário é sempre muito importante no processo de recomendação, porém preencher formulários exaustivos com suas características/preferências não é uma solução viável e, por outro lado, os sistemas utilizar-se

de poucos dados sobre o usuário dificulta o conhecimento sobre o mesmo. Nesse contexto, o enriquecimento semântico vem a suprir essa necessidade e, juntamente com a análise dos históricos de acessos e relações, podem prover uma solução de recomendação apropriada.

O segundo fator importante visto nos trabalhos citados é que, apesar de alguns deles até utilizarem-se da mobilidade, não utilizam-se da sensibilidade ao contexto, ou vice-versa, deixando a desejar na recomendação.

Por fim, mas não menos importante, citamos o fato de que quase todos os sistemas vistos recomendam conteúdos padronizados em algum formato, o que dificulta a aprendizagem informal, limitando a quantidade e a diversidade dos conteúdos.

Dessa forma, a modelagem do SR desenvolvida destaca-se como sendo a principal contribuição científica desse trabalho, pois uni todos os pontos fortes dos trabalhos anteriores e tenta minimizar os pontos fracos. O SRHU modelado apresenta características de:

a) Colaboração, ponderando inclusive o grau de influência que um usuário tem em relação a outro;
b) Baseada em Conteúdo, analisando as características dos históricos de acessos de cada usuário com as características do conteúdo a ser recomendado;
c) Mobilidade, sendo voltado a dispositivos móveis;
d) Ubiquidade, levando em consideração características inerentes ao contexto do usuário e do dispositivo móvel que o mesmo está usando;
e) Recomendação Conjunta, evitando assim recomendar conteúdos que disputem pelo mesmo espaço de características, priorizando assim o êxito ao conjunto completo a ser recomendado;
f) Enriquecimento Semântico, apesar de não ser objeto do escopo desse trabalho a realização do enriquecimento semântico, o mesmo é utilizado para comparação entre usuários e para análise de recomendação de conteúdo a usuário.
g) Adaptação Automática e Personalização, com ajustes automático dos pesos com base no histórico dos usuários.

4 A PLATAFORMA DO SISTEMA MOBILE+

Para contextualizar as funcionalidades do SR faz-se necessário antes expor os elementos e o funcionamento do sistema de aprendizagem, MobiLE+[3], desenvolvido pelo grupo de pesquisa, no qual está inserido o SR proposto.

O trabalho do grupo de pesquisa visa desenvolver o sistema Mobile+, um ambiente de aprendizagem ubíqua que possibilita o acesso, a visualização e o fornecimento de conteúdo educacional independentemente do horário e a partir de qualquer localidade e que direcione os estudantes de modo a maximizar seu aprendizado, mitigando a ausência do professor nos ambientes virtuais de aprendizagem. Além disso, este ambiente pode beneficiar-se da mobilidade do estudante para fornecer oportunidades educacionais presentes no ambiente real, assim como, obter informações de experiências reais, através do meio digital, providas pelas redes sociais, a fim de favorecer a autoaprendizagem e interações entre os estudantes. Desta forma, o estudante terá acesso às vantagens didático-pedagógicas oferecidas por um ambiente de aprendizagem formal, considerando, também, a importância da aprendizagem informal, uma vez que esta faz parte da vida cotidiana do estudante.

4.1 SUBPROJETOS DO SISTEMA

O ambiente MobiLE+ é formado por uma arquitetura composta por três subprojetos principais. Cada subprojeto possui uma especialidade e colabora para construção do ambiente de aprendizagem ubíqua. De acordo com o modelo temos:

a) Interface Móvel e Serviço Web: Tem como finalidade: i) fornecer uma interface de interação do usuário com os conteúdos educacionais e mídias sociais; ii) monitorar e capturar as informações dos usuários; iii) prover toda a arquitetura necessária para o funcionamento do MobiLE+; e iv) criar uma arquitetura multiagente para monitorar os usuários, enriquecer os perfis e conteúdos gerados pelos usuários, utilizando o Sistema de Enriquecimento Semântico (SES) descrito na alínea "b", e realizar a recomendação

[3] O sistema MobiLE+ foi construído com base na arquitetura YouUbi descrita em www.youubi.com, Monteiro (2014).

no momento apropriado, utilizando o SRHU, descrito a seguir na alínea “c”.

b) SES: A solução desenvolvida consiste em um sistema de enriquecimento semântico de perfil de usuário com base nos acessos do mesmo a conteúdos relacionados com domínios de assuntos registrados no sistema através de ontologias, baseando-se em técnicas de processamento de linguagem natural, ontologias e algoritmos de seleção e marcação de conteúdo. Através do processo de enriquecimento sabemos: i) qual a relação entre um determinado recurso e um determinado assunto; e ii) qual a relação de interesse de um usuário para com um domínio específico.
c) SRHU: Este componente tem como funções principais: i) Realizar análise periódica na base de dados e no SES gerando pré-listas de conteúdos potenciais à recomendação, sendo estas a Lista Colaborativa Ponderada (LCP) e a Lista Baseada em Conteúdo (LBC); e ii) Com as pré-listas preparadas, ao ser solicitado pelo componente do cliente móvel, gerar a Lista Híbrida Ubíqua (LHU), considerando os dados do contexto, e por fim a Recomendação Conjunta ou Final, que será exposta para o usuário através da Interface Móvel.

4.2 PROJETO DO SISTEMA

Para proporcionar modularidade ao sistema e permitir uma comunicação transparente e segura, assim como também para melhor dividir as tarefas do grupo de pesquisa no desenvolvimento do projeto, optou-se por dividir todo o sistema em nove módulos, os quais estão descritos a seguir:

a) *ubi_client_android*: Aplicativo desenvolvido para a plataforma Android, responsável pela interação do usuário com os conteúdos educacionais e mídias sociais. Trata-se da interface da arquitetura do MobiLE+ que realiza comunicação com os demais componentes da arquitetura, através do serviço *web*. Além de servir como interface de interação, este módulo é responsável por capturar e monitorar algumas informações, tais como: a localização geográfica, as características e estado do dispositivo e dados gerados pela interação do usuário com o dispositivo móvel. Toda transmissão do cliente ao servidor, seja informação capturada através do monitoramento ou por alguma ação realizada pelo usuário, segue um protocolo concebido para a arquitetura do MobiLE+.

b) *ubi_api_interface*: Representa uma interface compartilhada entre o cliente e o servidor. Desta forma, este descreve os métodos que o servidor disponibiliza para o cliente acessar via serviço *web*.

c) *ubi_api_communic_ws*: Implementa o serviço *web* responsável por receber e tratar as requisições realizadas pelo cliente. Cada requisição é direcionada para ser atendida por uma classe especialista do módulo *ubi_manager*, descrito posteriormente. Assim como o cliente, o serviço web segue as especificações do protocolo de comunicação criado para o MobiLE+. Uma descrição detalhada do protocolo de comunicação utilizado neste projeto será apresentada posteriormente. Este módulo implementa também o serviço *Google Cloud Messaging* – GCM. Com o serviço GCM é possível enviar dados para o aplicativo cliente mesmo quando este não está em execução e, consequentemente, sem a necessidade de uma requisição inicial. Logo, um agente de software pode se beneficiar dessa característica do sistema para enviar uma mensagem de notificação diretamente para cliente.

d) *ubi_model*: Este módulo compreende as classes que constituem as regras de negócio do sistema, assim como seu modelo de dados. A lógica do sistema é representada por essas classes e suas relações. Também implementa o modelo de dados do sistema, através do uso de anotações do JPA[4], refletindo diretamente na estrutura das entidades do bando de dados.

e) *ubi_manager*: Gerencia o fluxo de trabalho para o conjunto de classes do *ubi_model*. Desta forma, controla a instanciação e a colaboração (comunicação) entre as classes, levando em consideração as relações já definidas no modelo de dados.

f) *ubi_server_collector*: Realiza o monitoramento e a coleta de informações geradas pelos usuário através da sua interação com mídias sociais. Leva em consideração que essa interação não será realizada, exclusivamente, por meio de um dispositivo móvel, mas poderá ocorrer mediante outros recursos computacionais. Portanto, este módulo monitora determinadas atividades, buscando traçar um histórico de ações ocorridas fora dos limites do ambiente MobiLE+. Vale ressaltar que essas atividades são de suma importância para acompanhar a evolução do perfil do usuário, apesar de não conterem informações contextuais deste. Este módulo atua também no refinamento de

[4] O Java Persistence API (JPA) é uma especificação de persistência padrão aprovada pela JCP (Java Community Process) em meados de 2006 como parte da especificação EJB3 (Enterprise Java Beans 3) (BARRETO; CAMPOS, 2010).

informações coletadas, transformando coordenadas geográficas em informações legíveis a humanos e identificando pontos de interesses próximos à localização atual do usuário.

g) *ubi_server_analyzer*: Fornece uma interface genérica de enriquecimento semântico de perfil de usuário e recursos com base em interações em redes sociais, armazenando as informações coletadas em ontologias. Este módulo possui interfaces tanto para sistemas de recomendação de conteúdo quanto para sistemas de captura de dados de usuários e recursos. Ele recebe as informações capturadas pela interface móvel e os recursos (metadados de vídeos, comentários, endereços de páginas web etc.) utilizados pelo usuário e aplica o processo de enriquecimento semântico. Os perfis e recursos enriquecidos semanticamente são armazenados em ontologias formando uma base de dados enriquecida.

h) *ubi_server_recommender*: Consiste de um SRHU, onde são aplicadas técnicas de Recomendação Colaborativa e Recomendação Baseada em Conteúdo, levando em consideração os perfis e conteúdos enriquecidos semanticamente e o contexto do usuário no momento da recomendação.

i) *ubi_server_persistence*: É formado pelas classes que realizam o acesso ao banco de dados, fornecendo uma interface única de acesso. Tem como objetivo separar as regras de acesso ao banco de dados das regras de negócio da aplicação, contribuindo para diminuir o acoplamento do sistema como um todo.

Os componentes deste ambiente podem ser observados no Diagrama de Componentes apresentado na Figura 5.

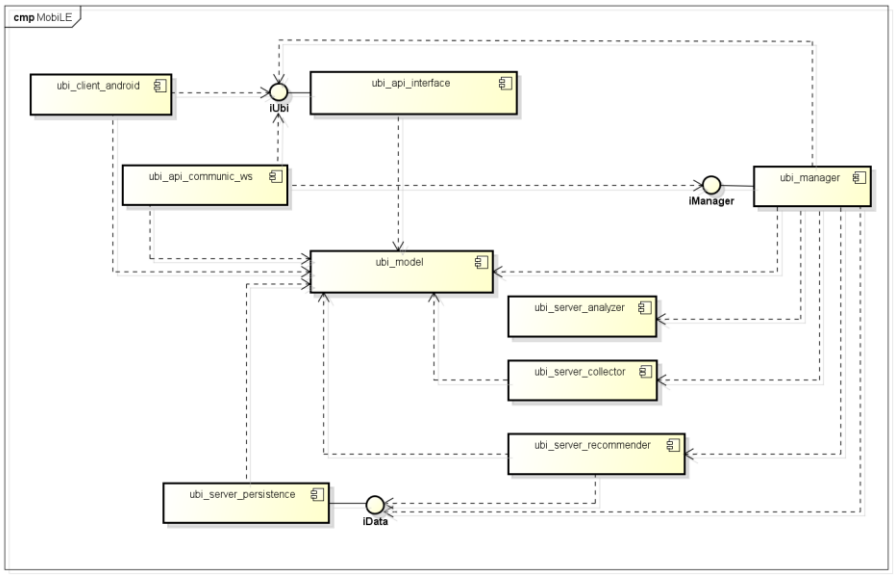

Figura 5 – Modelo de Componentes do Sistema MobiLE+
Fonte: Jácome, Luiz JR (2014).

4.3 O ESCOPO DESSA DISSERTAÇÃO

Como citado anteriormente, devido à complexidade do problema, modelou-se um sistema formado por três subprojetos, sendo: (i) um componente capaz de capturar as interações dos usuários com os dispositivos móveis, assim como as características desses dispositivos e do ambiente onde estão sendo utilizados; (ii) um componente que, a partir dos dados coletados no item anterior, enriquece semanticamente os perfis dos usuários e do conteúdo acessado; e (iii) um componente que realiza a recomendação de conteúdos para usuários.

No contexto do problema citado, o objetivo desse trabalho se refere apenas ao item (iii), sistema de recomendação, sendo esse o escopo dessa dissertação.

4.3.1 O Sistema de Recomendação e a Base de Dados

Em relação ao escopo desse trabalho, a abordagem proposta para solucionar o problema é a criação de um modelo de um SRHU, que será melhor detalhado no Capítulo 5, mas que, para facilitar o entendimento desse detalhamento, faremos aqui uma explanação sobre as classes, métodos, atributos e relações que estão envolvidas no SRHU.

A Tabela 3 descreve as classes do MobiLE+ envolvidas nas principais funcionalidades do sistema proposto em relação aos aspectos da recomendação.

Tabela 3 – Classes relevantes ao funcionamento do SRHU.

Tipo	Nome	Descrição
Básicas	*Person*	Representa os usuários cadastrados no sistema e que vão interagir e receber recomendações de conteúdo.
	Content	Representa todo o tipo de conteúdo do sistema que pode servir de objeto de aprendizagem para um usuário e possui um endereço web.
	Location	Representa uma posição no globo, longitude e latitude, e os detalhes que identificam o que está localizado nessa posição.

Relações	*RelationPersonPerson*	Representa as formas de relacionamento entre os usuários.
	RelationPersonContent	Representa as formas de relacionamento entre os usuários e os conteúdos.
	RelationPersonLocation	Representa as formas de relacionamento entre os usuários e os locais.
	RelationContentLocation	Representa as formas de relacionamento entre os conteúdo e os locais.
Auxiliares	*Device*	Representa os aparelhos ou dispositivos, e suas características e limitações, pelos quais o usuário pode acessar o sistema.
	Context	Representa um "retrato" de um instante de um usuário, com seu respectivo dispositivo, em um determinado local, interagindo de alguma forma com o sistema, acessando ou não um conteúdo.
	Learning	Representa uma extensão da classe *Person* com características específicas que identificam a formação e a forma de aprendizagem do usuário.
	Frequency	Representada por quatro histogramas, sendo classificados em diários (dois) ou semanais (dois), positivos ou negativos. Os positivos representam o ato de aceitação ou visualização de conteúdo, enquanto que os negativos representam a rejeição a uma recomendação de conteúdo. Se relacionam atributos estendidos das classes básicas *person*, *content* e *location.*
	Tags	Representadas por uma lista de termos que em conjunto possuem alguma relevância e identificam os interesses ou assuntos relacionados a uma das classes básicas *Person*, *Content* ou *Location*.

Fonte: Autor.

As classes descritas na Tabela 1 são utilizadas pelo SRHU sobretudo com a finalidade de estabelecer semelhanças entre as suas instâncias e, para isso, utiliza-se da comparação de

seus atributos, incrementando as variáveis que servem de índices na comparação, quando os valores dos atributos são semelhantes, e decrementando quando são diferentes.

Dessa forma torna-se importante descrever os atributos considerados em cada uma dessas classes:

a) *Person*:

- *id*: Identificador padrão da instância e chave primária da tabela no banco de dados.
- *email*: Representa o email do usuário, único e utilizado como *login*.
- *pass*: Armazena a senha do usuário para acessar o sistema.
- *nameFirst*: Identifica o primeiro nome do usuário.
- *nameLast*: Identifica o último nome do usuário.
- *dateRegister*: Representa a data de registro do usuário no sistema.
- *dateBirth*: Representa a data de nascimento do usuário.
- *genre*: Identifica o sexo do usuário.
- *phone*: Representa o número do telefone do usuário.
- *income*: Indentifica a renda do usuário, expressa em reais.
- *religion*: Identifica a religião do usuário.
- *race*: Identifica a cor ou raça do usuário.
- *profession*: Identifica a profissão do usuário.
- *rateDurationContents*: Indica, em segundos, a média de duração dos conteúdos visualizados pelo usuário.
- *learning*: Representa uma instância da classe *Learning* e caracteriza a formação e modo de aprendizado do usuário. Seus atributos são melhor detalhados no item que explica a própria classe *Learning*.
- *frequency*: Representa os histogramas de aceitação e rejeição, semanais e diários do usuário. Seus atributos são melhor detalhados no item que explica a própria classe *Frequency*.
- *listContexts*: Representa o histórico de contextos vividos pelo usuário. Seu último registro representa o momento atual. Seus atributos são melhor detalhados no item que explica a própria classe *Context*.
- *listDevices*: Representa a lista de aparelhos do usuário. Seus atributos são melhor detalhados no item que explica a própria classe *Device*.
- *listTags*: Representa a lista de *tags* de um usuário. Seus atributos são melhor

detalhados no item que explica a própria classe *Tag*.

- *listRelatePersonPerson*: Representa todas as relações entre os usuários. Seus atributos são melhor detalhados no item que explica a própria classe *RelationPersonPerson*.
- *listRelatePersonContent*: Representa todas as relações entre os usuários e os conteúdos. Seus atributos são melhor detalhados no item que explica a própria classe *RelationPersonContent*.
- *listRelatePersonLocation*: Representa todas as relações entre os usuário e locais. Seus atributos são melhor detalhados no item que explica a própria classe *RelationPersonLocation*.

b) *Content*:

- *id*: Identificador padrão da instância e chave primária da tabela no banco de dados.
- *type*: Identifica se tipo de conteúdo (postagem, evento, grupo ou desafio). Apenas as postagens estão sendo consideradas nesse primeiro momento para o sistema de recomendação.
- *subtype*: Identifica o subtipo do conteúdo. Para o tipo Postagem, por exemplo, o subtipo pode assumir os valores de vídeo, áudio, página, imagem ou texto.
- *visibility*: Define os níveis de privacidade de um conteúdo. Pode estar visível a nível de todos, amigos ou privado. Apenas os visíveis a cada usuário são utilizados no sistema de recomendação.
- *title*: Representa o título do conteúdo.
- *description*: Representa a descrição do conteúdo.
- *dateCreation*: Representa a data de criação de um conteúdo.
- *urlOnline*: Representa o endereço *web* vinculado ao conteúdo.
- *bytesOnline*: Representa o tamanho (em *bytes*) do conteúdo.
- *secondsOnline*: Representa a duração (em segundos) necessária para conhecer todo o conteúdo.
- *rating*: Representa a média das avaliação dos usuários em relação ao conteúdo.
- *rateAcceptance*: Representa o percentual de aceitação dos usuários em relação ao conteúdo, quando este é recomendado.
- *perso*n: Representa o usuário que inseriu este conteúdo no sistema.

- *author*: Representa o autor do conteúdo.
- *frequency*: Representa os histogramas de aceitação e rejeição, semanais e diários do conteúdo. Seus atributos são melhor detalhados no item que explica a própria classe *Frequency*.
- *listTags*: Representa a lista de *tags* de um conteúdo. Seus atributos são melhor detalhados no item que explica a própria classe *Tag*.
- *listRelatePersonContent*: Representa todas as relações entre os conteúdos e os usuários. Seus atributos são melhor detalhados no item que explica a própria classe *RelationPersonContent*.
- *listRelateContentLocation*: Representa todas as relações entre os conteúdos e os locais. Seus atributos são melhor detalhados no item que explica a própria classe *RelationContentLocation*.

c) *Location*:

- *id*: Identificador padrão da instância e chave primária da tabela no banco de dados.
- *title*: Representa o título do local ou nome que o identifica.
- *description*: Representa a descrição do local.
- *altitude*: Representa, em metros, a altitude do local em relação ao nível do mar.
- *longitude*: Representa a coordenada geográfica de longitude onde está situado o local.
- *latitude*: Representa a coordenada geográfica de latitude onde está situado o local.
- *country*: Representa a parte do endereço que identifica o país onde está localizado o local.
- *state*: Representa a parte do endereço que identifica o estado onde está localizado o local.
- *city*: Representa a parte do endereço que identifica a cidade onde está localizado o local.
- *neighborhood*: Representa a parte do endereço que identifica o bairro onde está localizado o local.
- *streetName*: Representa a parte do endereço que identifica o nome da rua onde está localizado o local.

- *streetNumber*: Representa a parte do endereço que identifica o número onde está localizado o local.
- *zipcode*: Representa a parte do endereço que identifica o código de endereçamento postal onde está localizado o local.
- *idGoogle*: Representa o identificador do local no sistema de mapeamento utilizado.
- *rating*: Representa a média das avaliação dos usuários em relação ao local.
- *rateAcceptanceHere*: Representa o percentual de aceitação dos usuários em relação aos conteúdos recomendados nesse local.
- *rateRatingHere*: Representa a média das avaliações dos usuários em relação aos conteúdos recomendados nesse local.
- *person*: Representa o usuário que inseriu este local no sistema.
- *frequency*: Representa os histogramas de aceitação e rejeição, semanais e diários, de conteúdo no local. Seus atributos são melhor detalhados no item que explica a própria classe *Frequency*.
- *listTypes*: Representa os tipos de funcionalidades relativas ao local, sendo extraída do sistema de mapeamento.
- *listTags*: Representa a lista de *tags* de um local. Seus atributos são melhor detalhados no item que explica a própria classe *Tag*.
- *listRelatePersonLocation*: Representa todas as relações entre os usuário e locais. Seus atributos são melhor detalhados no item que explica a própria classe *RelationPersonLocation*.
- *listRelateContentLocation*: Representa todas as relações entre os conteúdos e os locais. Seus atributos são melhor detalhados no item que explica a própria classe *RelationContentLocation*.

d) *RelationPersonPerson*:

- *person1*: Identifica o primeiro usuário da relação.
- *person2*: Identifica o segundo usuário da relação.
- *status*: Identifica qual tipo de relação há entre os usuários. Pode assumir os valores de: usuários amigos, usuário A solicitou amizade de usuário B, usuário A teve amizade solicitada por usuário B, usuário A rejeitou amizade de usuário B, usuário A teve amizade rejeitada por usuário B, usuário A deletou amizade com usuário B, usuário A teve amizade deletada por usuário

B, usuário A visualizou o perfil do usuário B, ou usuário A teve perfil visualizado por usuário B.

- *affinityRate*: Identifica o grau de afinidade entre *person1* e *person2*. Este índice é calculado com base em vários atributos das classes e dos históricos de cada um dos usuários.
- *commomPersons*: Representa o número de amigos em comum entre *person1* e *person2*.
- *dateRelation*: Representa a data da criação da relação.

e) *RelationPersonContent*:

- *person*: Identifica o usuário da relação.
- *content*: Identifica o conteúdo da relação.
- *status*: Identifica qual o tipo de relação existente entre o usuário e o conteúdo. Pode assumir os valores de: conteúdos favoritos do usuário, conteúdos sugeridos ao usuário, conteúdos visualizados pelo usuário, conteúdos rejeitados pelo usuário ou conteúdos avaliados pelo usuário.
- *dateRelation*: Representa a data da criação da relação.
- *countRelation*: Representa quantas vezes essa relação ocorreu com esse status.
- *ratePerson*: Avaliação de *person* sobre esse conteúdo. Caso o conteúdo ainda não tenha sido avaliado por *person*, *ratePerson* assumi o valor de 5,0.
- *i1*: No caso de conteúdo sugerido a um usuário, *i1* representa o peso da recomendação colaborativa desse conteúdo para o usuário.
- *i2*: No caso de conteúdo sugerido a um usuário, *i2* representa o peso da recomendação baseada em conteúdo desse conteúdo para o usuário.
- *i3*: No caso de conteúdo sugerido a um usuário, *i3* representa o peso da recomendação baseada na ubiquidade[5] desse conteúdo para o usuário.

f) *RelationPersonLocation*:

- *person*: Identifica o usuário da relação.
- *location*: Identifica o local da relação.
- *status*: Identifica qual o tipo de relação existente entre usuário e conteúdo

[5] Recomendação Baseada na Ubiquidade é voltada para a aprendizagem ubíqua (também conhecida como *u-learning*, de *ubiquitous learning*), ou seja, com características de mobilidade e sensibilidade ao contexto, de acordo com a Subseção 2.3.

nesse local. Pode assumir os valores de: locais favoritos do usuário, conteúdos foram sugeridos ao usuário nesse local, conteúdos foram visualizados pelo usuário nesse local, conteúdos foram rejeitados pelo usuário nesse local, ou conteúdos foram avaliados nesse local.

- *finality*: Identifca a finalidade do uso do local pelo usuário nessa relação. Pode assumir os valores de: educação, trabalho, casa/lar, lazer ou outro.
- *ratePerson*: Para o caso do status ser "avaliado", identifica a avaliação do usuário sobre esse local, caso contrário recebe o valor padrão de 5,0.
- *dateRelation*: Representa a data da criação da relação.

g) *RelationContentLocation*:

- *content*: Identifica o conteúdo da relação.
- *location*: Identifica o local da relação.
- *dateRelation*: Representa a data da criação da relação.
- *status*: Identifica qual o tipo de relação ocorreu para o conteúdo nesse local. Pode assumir os valores de: conteúdo sugerido nesse local, conteúdo rejeitado nesse local, conteúdo criado nesse local, conteúdo visualizado nesse local, o assunto principal do conteúdo é o local, o assunto principal do conteúdo é relacionado ao local, o local é citado no conteúdo ou o conteúdo foi avaliado nesse local.

h) *Device*:

- *id*: Identificador padrão da instância e chave primária da tabela no banco de dados.
- *type*: Identifica o tipo de dispositivo. Pode assumir os valores de *smartphone*, *tablet*, *notebook* ou *desktop*.
- *os*: Identifica qual o sistema operacional utilizado pelo dispositivo.
- *osVersion*: Identifica a versão do sistema operacional utilizado pelo dispositivo.
- *hasWifi*: Informa se o dispositivo tem o recurso de *Wifi*.
- *hasGPS*: Informa se o dispositivo tem o recurso de GPS (*Global Position System* – Sistema de Posicionamento Global).
- *hasBluetooth*: Informa se o dispositivo tem o recurso de *Bluetooth*.
- *displayWidth*: Informa a largura, em pixel, do *display* do dispositivo.
- *displayHeight*: Informa a altura, em pixel, do *display* do dispositivo.

- *productBrand*: Informa a marca do dispositivo.
- *productModel*: Informa o modelo do dispositivo.
- *mac*: Informa o endereço MAC (*Media Access Control*) do dispositivo.
- *description*: Informa a descrição do dispositivo.
- *person*: Identifica o usuário relacionado ao dispositivo.

i) *Context*:

- *id*: Identificador padrão da instância e chave primária da tabela no banco de dados.
- *type*: Identifica o tipo de evento que disparou a criação do contexto. Pode ser uma alteração de local, um recebimento, criação, ou postagem de um conteúdo, ou alguma outra ação do usuário no dispositivo.
- *timeStamp*: Identifica o momento da criação do contexto.
- *status*: Identifica se o usuário está *online* ou *off-line* no sistema.
- *battery*: Informa o percentual da carga da bateria do dispositivo no momento da criação do contexto.
- *ip*: Identifica o endereço IP (*Internet Protocol*) do dispositivo no momento da criação do contexto.
- *connectionSpeed*: Informa a velocidade da conexão, em kbps, no momento da criação do contexto.
- *personSpeed*: Informa a velocidade do usuário, em m/s, no momento da criação do contexto.
- *person*: Identifica o usuário relacionado ao contexto.
- *location*: Identifica o local da criação do contexto.

j) *Learning*:

- *id*: Identificador padrão da instância e chave primária da tabela no banco de dados.
- *educationalLevel*: Identifica o nível educacional do usuário. Pode ser: fundamental completo ou incompleto, médio completo ou incompleto, superior completo ou incompleto, ou pós-graduação.
- *formationCourse*: Para usuário com nível educacional superior incompleto ou acima, informa o nome do curso de graduação.
- *formationBigArea*: Para usuário com nível educacional superior incompleto ou acima, informa a grande área do curso de graduação.

- *formationArea*: Para usuário com nível educacional superior incompleto ou acima, informa a área do curso de graduação.
- *formationSubarea*: Para usuário com nível educacional superior incompleto ou acima, informa a subárea de trabalho ou pesquisa do usuário.
- *styleDimenPerception*: Informa o estilo de aprendizagem quanto à dimensão de percepção do usuário. Pode ser: *sensorial* (gosta de aprender a partir de materiais concretos e tende a ser mais prático) *ou intuitiva* (prefere materiais abstratos, como teorias e seus significados, e costuma ser mais inovador) (LINDEMANN, 2008).
- *styleDimenRetention*: Informa o estilo de aprendizagem quanto à dimensão de retenção do usuário. Pode ser: visual (lembra-se melhor daquilo que usualmente ver, como imagens, gráficos, animações, entre outros) ou verbal (maior facilidade em lembrar-se de palavras, sejam elas escritas ou faladas) (LINDEMANN, 2008).
- *styleDimenProcessing*: Informa o estilo de aprendizagem quanto à dimensão de processamento do usuário. Pode ser: ativo (prefere aprender por meio de suas experiências. Geralmente trabalha bem com outros alunos) ou reflexivo (diferentemente do ativo, prefere aprender sozinho refletindo sobre o assunto em questão) (LINDEMANN, 2008).
- *styleDimenOrganization*: Informa o estilo de aprendizagem quanto à dimensão organizacional do usuário. Pode ser: sequencial (prefere aprender por meio de passos lineares, levando a um processo gradual de aprendizagem) ou global (aprende através de passos largos. É holístico, isto é, prefere entender primeiro o todo e em seguida suas partes) (LINDEMANN, 2008).

k) *Frequency*:

- *id*: Identificador padrão da instância e chave primária da tabela no banco de dados.
- *h24Positive*: É composto de vinte e quatro posições que representam as horas do dia, indicando o percentual de aceitação de um conteúdo por parte de um usuário em um determinado local, para cada uma dessas horas.
- *h24Negative*: É composto de vinte e quatro posições que representam as horas do dia, indicando o percentual de rejeição de um conteúdo por parte de um usuário em um determinado local, para cada uma dessas horas.

- *d7Positive*: É composto de sete posições que representam os dias da semana, indicando o percentual de aceitação de um conteúdo por parte de um usuário em um determinado local, para cada um desses dias.
- *d7Negative*: É composto de sete posições que representam os dias da semana, indicando o percentual de rejeição de um conteúdo por parte de um usuário em um determinado local, para cada um desses dias.

l) *Tag*:

- *id*: Identificador da instância e chave primária da tabela no banco de dados.
- *name*: Representa o termo que identifica a *tag*.
- *count*: Informa o número de vezes que a *tag* é representada.

Todos essas classes e atributos descritos influenciam em alguma etapa do sistema de recomendação. A forma como cada um deles é utilizado é melhor detalhado no Capítulo 5.

4.3.2 O Sistema de Recomendação e o Enriquecimento Semântico

Outro módulo do sistema MobiLE+ que é utilizado na recomendação é o SES. O SES é utilizado tanto na criação da Lista de Colaboradores (LC) assim como na LBC.

O SES é composto pela seguinte estrutura de ontologias:

a) Ontologias de Domínio: Representam os mais variados temas ou assuntos que podem ser implantados no sistema, como: Computação, Direito, Medicina, Administração, entre outros.
b) Ontologias de Perfil: Representam os usuários ou os conteúdos do sistema.

O Sistema de Recomendação interage com SES por meio de sua interface. Os métodos a seguir retornam os índices a serem utilizados na geração das listas:

a) *getUserInterest*(...): Retorna um índice compreendido entre 0,0 e 10,0 que representa o quanto o usuário se interessa por um determinado domínio.
b) *getResourceDomainRelation*(...): Retorna um índice compreendido entre 0,0 e 10,0 que representa o quanto um conteúdo se relaciona com um determinado domínio.

Não pertence ao escopo desse trabalho o detalhamento do funcionamento do SES, portanto sendo suficiente as interfaces expostas. No Capítulo 5 demostraremos como estes índices influenciam a recomendação.

5 UM SISTEMA DE RECOMENDAÇÃO HÍBRIDO UBÍQUO

A solução desenvolvida consiste de uma sistema de recomendação que envolve a Recomendação Colaborativa e a Recomendação Baseada em Conteúdo, mas além disso ainda considera o Enriquecimento Semântico do usuário e do conteúdo, e também o contexto no momento da recomendação, implementando-se, dessa forma, o que chamamos de Recomendação Híbrida Ubíqua, como ilustrado na Figura 6.

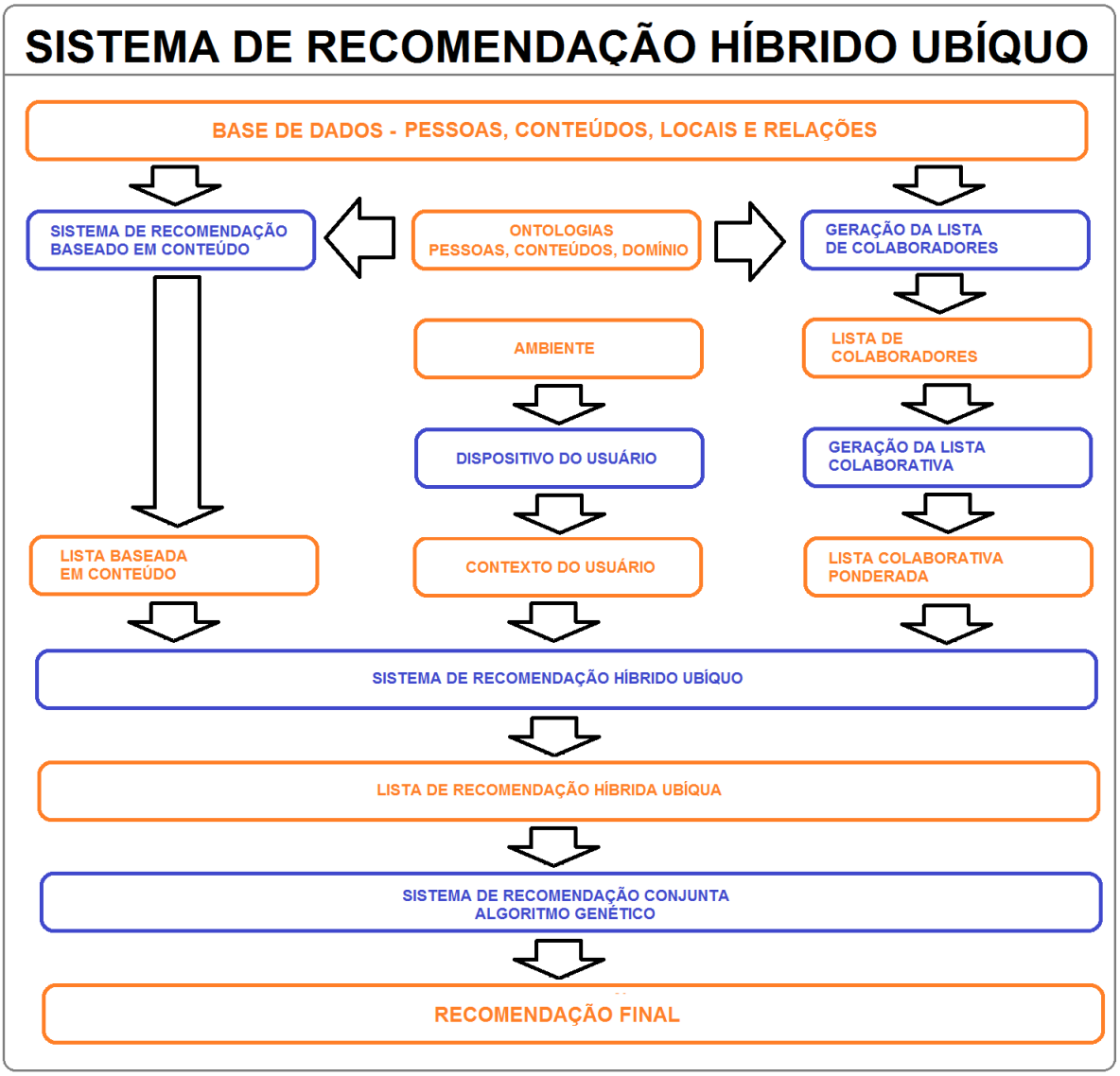

Figura 6 – Sistema de Recomendação Híbrido Ubíquo.
Fonte: Autor.

Na Figura 6 é ilustrado o fluxo de execução e a geração de listas de dados auxiliares para o sistema de recomendação.

5.1 RECOMENDAÇÃO COLABORATIVA PONDERADA

A Recomendação Colaborativa consiste na análise das escolhas e avaliações, para conteúdo, de usuários semelhantes ao usuário alvo. Dessa forma, surgiu a necessidade primeiramente de se escolher, para cada usuário, uma lista de usuários semelhantes a este, os quais denominamos de Lista de Colaboradores e posteriormente, a partir das relações destes com os conteúdos, gera-se a Lista Colaborativa Ponderada. A Figura 7 apresenta o esboço do fluxo de dados dessa operação.

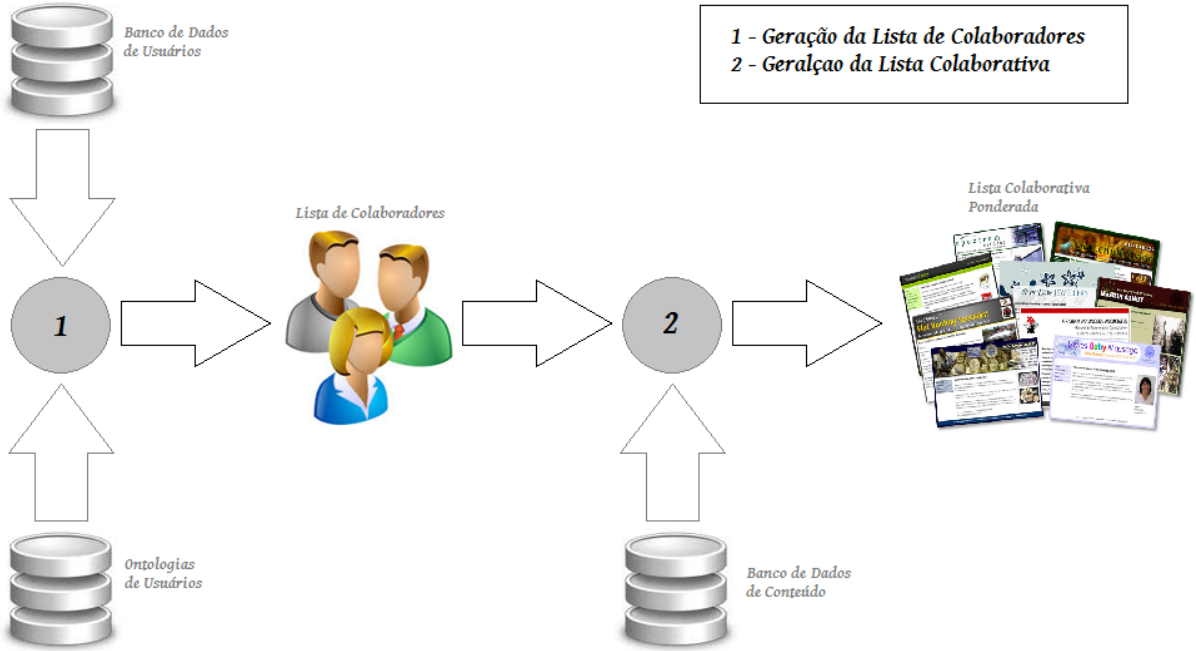

Figura 7 – Fluxo de Entrada e Saída da Recomendação Colaborativa
Fonte: Autor.

5.1.1 Lista de Colaboradores

O sistema executa uma rotina onde escolhe, para todos os usuários, quem são seus respectivos colaboradores, ou seja, pessoas com maior grau de semelhança, ordenando-os assim na Lista de Colaboradores e vinculando-os ao Índice de Semelhança entre Usuários - ISU.

Para o cálculo do ISU, leva-se em consideração os seguintes aspectos:

a) Meta-dados (*SI*): Analisa os dados de ambos os usuário, como idade, sexo, renda, religião, cor/raça, profissão, nível educacional, estilo de aprendizagem, área de

formação e suas especificações, de forma que dados coincidentes acrescentam o índice de semelhança de meta-dados, enquanto que dados divergentes decrescem este índice.

b) Relações (*S2*): As relações existentes entre dois usuários podem ser: amigos, requisitou amizade, amizade requisitada, rejeitou amizade, amizade rejeitada, deletou amizade, amizade deletada, visualizou usuário e usuário visualizado. Neste item o sistema analisa quais tipos de relações existem entre os usuários alvo, assim como as relações em comum destes com outros. Usuários onde ambos possuem relação "amigos" referente à relação entre eles acrescentará a pontuação na semelhança de relações entre eles, enquanto que se um usuário possui uma relação "amigos" e o outro uma relação "amizade deletada" isso decrescerá o índice entre ambos.

c) Conteúdos (*S3*): Analisa quais relações com conteúdo são comuns aos usuários alvo. As possíveis relações analisadas são: aceitação e rejeição de recomendação, e a visualização de conteúdo. Usuários que aceitam visualizar o mesmo conteúdo são acrescidos em sua pontuação de semelhança de conteúdo, enquanto que um usuário que rejeita um conteúdo visualizado por outro terá sua semelhança de conteúdo entre eles decrescida.

d) Locais (*S4*): Analisa quais relações com locais são comuns aos usuários alvo. Para fins da recomendação, as relações com locais são de aceitação e rejeição de recomendação, e de visualização de conteúdo. Se dois usuários relacionam-se da mesma forma com um local, isso acrescentará pontos na semelhança de locais, entretanto se um usuário sempre aceita conteúdos e outro sempre rejeita conteúdos no mesmo local, isso decrescerá a semelhança de locais entre esses usuários.

e) Frequência (*S5*): Os histogramas de frequência armazenam as respostas de aceitação e rejeição em relação às recomendações geradas para os usuário, conteúdo e locais. Eles são de dois tipos: semanais, que acumulam os valores classificando-os em cada dia da semana; e os diários, que classificando-os por hora do dia. Neste item o sistema analisa a semelhança entre os histogramas de aceitação e rejeição de conteúdos pelos usuários alvo, ou seja, se dois usuários possuem percentuais de aceitação semelhantes na mesma hora do dia, ou no mesmo dia da semana, isso incrementará o índice de semelhança de frequências, enquanto que se divergir decrescerá este índice.

f) *Tags (S6)*: Analisa a semelhança entre a lista de *tags* dos usuários alvos, onde *tags* semelhantes incrementam o índice de semelhança de *tags*.

g) Enriquecimento Semântico (*S7*): Analisa a semelhança entre os enriquecimentos semânticos dos usuários alvo. O sistema de recomendação solicita ao sistema de enriquecimento quais as afinidades de dois usuários com cada um dos domínios instalados, e as compara, extraindo assim a semelhança semântica entre os usuários.

O ISU entre o usuário "i" e o usuário "j" é calculado como sendo a média ponderada de cada uma das semelhanças descritas nos itens anteriores, de acordo com a Equação 1.

$$\mathrm{ISU}_{i,j} = \frac{\mathrm{S1}*\mathrm{P1} + \mathrm{S2}*\mathrm{P2} + \mathrm{S3}*\mathrm{P3} + \mathrm{S4}*\mathrm{P4} + \mathrm{S5}*\mathrm{P5} + \mathrm{S6}*\mathrm{P6} + \mathrm{S7}*\mathrm{P7}}{\mathrm{P1} + \mathrm{P2} + \mathrm{P3} + \mathrm{P4} + \mathrm{P5} + \mathrm{P6} + \mathrm{P7}}$$

Equação 1 – Cálculo do Índice de Semelhança de Usuários – ISU

Onde S_k representa o nível de semelhança de determinada característica e P_k representa o peso dessa característica na avaliação.

Os *n* maiores resultados de ISU, vinculados aos seus respectivos pares de usuários, são armazenados na Lista de Colaboradores de tamanho *n* de um usuário.

Todo esse processo é executado no servidor periodicamente sobre todos os usuários, atualizando-se assim sua Lista de Colaboradores para posterior uso no cálculo da Lista Colaborativa Ponderada.

5.1.2 Lista Colaborativa Ponderada

A Lista Colaborativa Ponderada (LCP) é resultante da análise do histórico das relações com conteúdo dos usuários presentes na Lista de Colaboradores de cada usuário alvo. Logo, a LCP é uma lista de conteúdos escolhidos com base nas escolhas dos colaboradores dos usuários.

O sistema percorre toda a Lista de Colaboradores de cada usuário analisando as relações destes com os conteúdos e suas respectivas avaliações. Cada conteúdo novo identificado é adicionado a uma lista temporária que contém o identificador do conteúdo e um Índice de Semelhança Colaborativa – ISC.

Dado um usuário alvo identificado por *Ui*, sabe-se que:

a) *Ui* possui sua Lista de Colaboradores contendo *n* usuários e seus respectivos ISU relativos a U_i;
b) Cada usuário *j*, pertencente à Lista de Colaboradores, possui uma lista de relações com conteúdo;

c) Sendo $A_{j,k}$ a avaliação dada pelo usuário *j* (pertencente à Lista de Colaboradores) ao conteúdo *k*, pertencente à sua Lista de Relações de Conteúdo;
d) *t* é o total de vezes que um conteúdo *k* aparece nas listas de relações com conteúdo dos usuários presentes na Lista de Colaboradores.

O $ISC_{k,i}$ de um determinado conteúdo "*k"* com um usuário "i" é calculado como sendo a média de todas as avaliações deste conteúdo pelos *n* usuários presentes na Lista Colaborativa do usuário *i*, ponderada por seus respectivos $ISU_{i,j}$, como mostra a Equação 2.

$$ISC_{i,k} = \frac{1}{t}\sum_{j=0}^{n-1} A_{j,k*} ISU_{i,j}$$

Equação 2 – Cálculo do Índice de Semelhança de Conteúdo - ISC

Os *m* maiores resultados de ISC são armazenados como a Lista Colaborativa Ponderada (LCP) de tamanho *m* de um usuário.

Esse processo é executado no servidor periodicamente sobre todos os usuários, atualizando-se assim sua Lista Colaborativa Ponderada para posterior uso no cálculo da Lista Ubíqua e da Lista Híbrida.

5.1.3 Índice de Semelhança Semântica entre Usuários

Na Subseção 5.1.1, destaca-se o cálculo do ISU aparece o índice *S7*, calculado como resultado do retorno do SES. Como o enriquecimento semântico é um dos pilares da recomendação, vamos detalhar como se dá o cálculo de *S7*, que aqui representa a semelhança semântica entre dois usuários "i" e "j".

No Capítulo 4 o SES calcula apenas o quanto um usuário se interessa por um determinado domínio. Logo, para calcular o quanto dois usuários, "i" e "j" se assemelham semanticamente utiliza-se a Equação 3.

$$S7_{i,j} = \frac{10*\mathrm{n} - \sum_{k=1}^{n} |A_{i,k} - A_{j,k}|}{n}$$

Equação 3 – Cálculo do Índice de Semelhança Semântica entre Usuários
Onde:

a) $S7_{i,j}$ é o índice de semelhança semântica entre o usuário *i* e o usuário *j*;

b) n é o número de domínios implantados no SES;
c) $A_{i,k}$ é o índice de interesse do usuário i pelo domínio k, retornado pelo SES;
d) $A_{j,k}$ é o índice de interesse do usuário j pelo domínio k, retornado pelo SES;

5.2 RECOMENDAÇÃO BASEADA EM CONTEÚDO

A Recomendação Baseada em Conteúdo (RBC) é executada sobre toda a base de usuários e gera, para cada um deles, uma Lista Baseada em Conteúdo (LBC). A RBC calcula o Índice Baseado em Conteúdo (IBC) que representa o quanto um conteúdo é recomendável a um usuário, considerando os históricos do conteúdo e do usuário. Na Figura 8 é apresentado o fluxo de dados e a geração da Lista Baseada em Conteúdo.

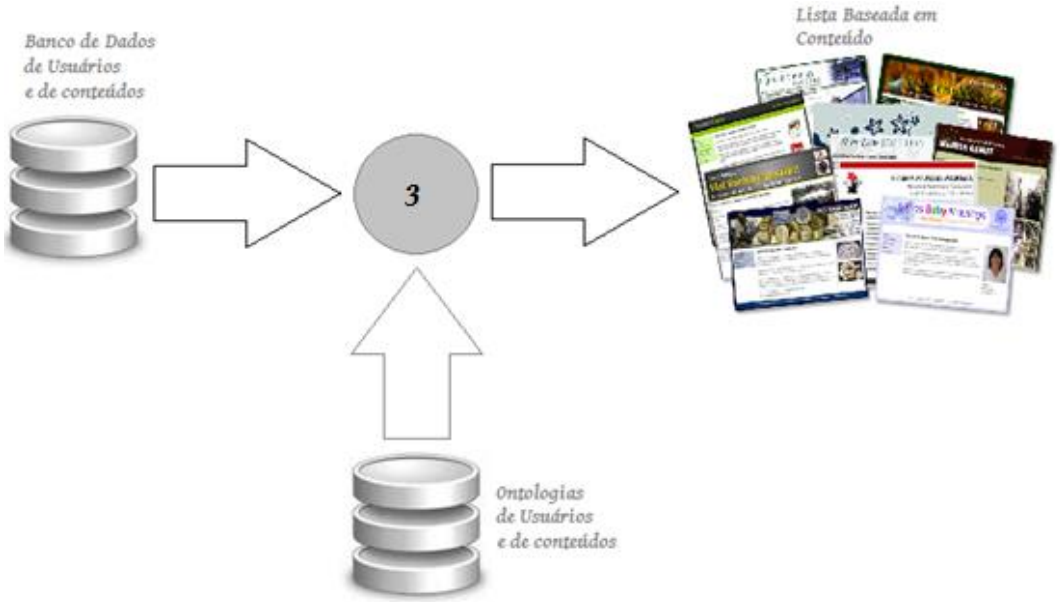

Figura 8 – Fluxo de Entrada e Saída da Recomendação Baseada em Conteúdo
Fonte: Autor.

Para o cálculo do IBC, leva-se em consideração os seguintes aspectos:

a) Meta-dados (*S1*): Analisa os dados do histórico de relacionamento de visualização de conteúdo do usuário, como tipo, tamanho, duração, autor, tempo da publicação, e compara as mesmas características às características do conteúdo alvo. Se as características são coincidentes ou semelhantes, incrementa-se o índice de semelhança de meta-dados. O sistema também considera os índices de aceitação e de avaliação do conteúdo alvo, onde conteúdos com índices altos são priorizados.

b) *Tags (S2)*: Analisa a semelhança entre a lista de *tags* do usuário e do conteúdo, onde *tags* semelhantes incrementam o índice de semelhança de *tags*.

c) Locais (*S3*): Analisa quais relações com locais são comuns ao usuário e ao conteúdo. Para fins da recomendação, as relações com locais podem ser de aceitação ou de rejeição de recomendação, e de visualização de conteúdo. Se um usuário e um conteúdo relacionam-se da mesma forma com um local, isso acrescentará pontos na semelhança de locais, entretanto se um usuário sempre aceita conteúdos nesse local e um determinado conteúdo sempre é rejeitado nesse mesmo local, isso decrescerá a semelhança de locais entre esse usuário e o conteúdo.

d) Frequência (*S4*): Os histogramas de frequência armazenam as respostas de aceitação e de rejeição em relação às recomendações geradas para os usuário, conteúdos e locais. Eles são de dois tipos: semanais, que acumulam os valores classificando-os em cada dia da semana, e os diários, que classificando-os por hora do dia. Neste item o sistema analisa a semelhança entre os histogramas de aceitação e de rejeição de conteúdo do usuário alvo e do conteúdo, ou seja, se ambos possuem percentuais de aceitação semelhantes na mesma hora do dia, ou no mesmo dia da semana, isso incrementará o índice de semelhança de frequências, enquanto que se divergir decrescerá este índice.

e) Enriquecimento Semântico (*S5*): Analisa a semelhança entre os enriquecimentos semânticos do usuário alvo e do conteúdo. O sistema de recomendação solicita ao sistema de enriquecimento quais as afinidades do usuários alvo com cada um dos domínios instalados, e as compara com as afinidade do conteúdo com os mesmos domínios, extraindo assim a semelhança semântica entre o usuário e o conteúdo.

O IBC entre o usuário "i" e o usuário "j" é calculado como sendo a média ponderada de cada uma das semelhanças descritas nos itens anteriores, de acordo com a Equação 4.

$$\mathrm{IBC}_{i,j} = \frac{\mathrm{S1}*\mathrm{P1} + \mathrm{S2}*\mathrm{P2} + \mathrm{S3}*\mathrm{P3} + \mathrm{S4}*\mathrm{P4} + \mathrm{S5}*\mathrm{P5}}{\mathrm{P1} + \mathrm{P2} + \mathrm{P3} + \mathrm{P4} + \mathrm{P5}}$$

Equação 4 – Cálculo do Índice Baseado em Conteúdo – IBC

Onde S_k representa o nível de semelhança de determinada característica e P_k representa o peso dessa característica na avaliação.

Os *n* maiores resultados de IBC, vinculados aos seus respectivos pares, usuário e conteúdo, são armazenados como a Lista Baseada em Conteúdo de tamanho *n* de um usuário.

5.2.1 Índice de Semelhança Semântica entre Usuário e Conteúdo

No cálculo do IBC aparece o índice *S5*, calculado como resultado do retorno do SES. Como o enriquecimento semântico é um dos pilares da recomendação, evidenciou-se como se dá o cálculo de *S5*, que aqui representa a semelhança semântica entre um usuário e um conteúdo.

No Capítulo 4, o SES calcula apenas o quanto um recurso (conteúdo) se relaciona com um determinado domínio. Logo, para calcular o quanto um usuário "i" se assemelha semanticamente a um conteúdo "j", utiliza-se a Equação 5.

$$S5_{i,j} = \frac{10 * \mathrm{n} - \sum_{k=1}^{n} |A_{i,k} - B_{j,k}|}{n}$$

Equação 5 – Cálculo do Índice de Semelhança Semântica entre Usuário e Conteúdo

Onde:

a) $S5_{i,j}$ é o índice de semelhança semântica entre o usuário *i* e o conteúdo *j*;
b) *n* é o número de domínios implantados no SES;
c) $A_{i,k}$ é o índice de interesse do usuário *i* pelo domínio *k*, retornado pelo SES;
d) $B_{j,k}$ é o índice que mede o quanto o conteúdo *j* se relaciona com domínio *k*, retornado pelo SES.

5.3 RECOMENDAÇÃO BASEADA NA UBIQUIDADE

A Recomendação Baseada na Ubiquidade (RBU) é executada em tempo real[6], para um determinado usuário alvo, e calcula o Índice de Recomendação Ubíqua (IRU), que representa o quanto um conteúdo é adequado ao contexto do usuário alvo. A Figura 9 apresenta o dispositivo do usuário capturando a informação do contexto e, com base na Lista Baseada em Conteúdo e na Lista Colaborativa Ponderada gerando a Lista Híbrida.

[6] Neste contexto, considerou-se tempo real, o tempo de resposta em que não se altere o contexto da recomendação.

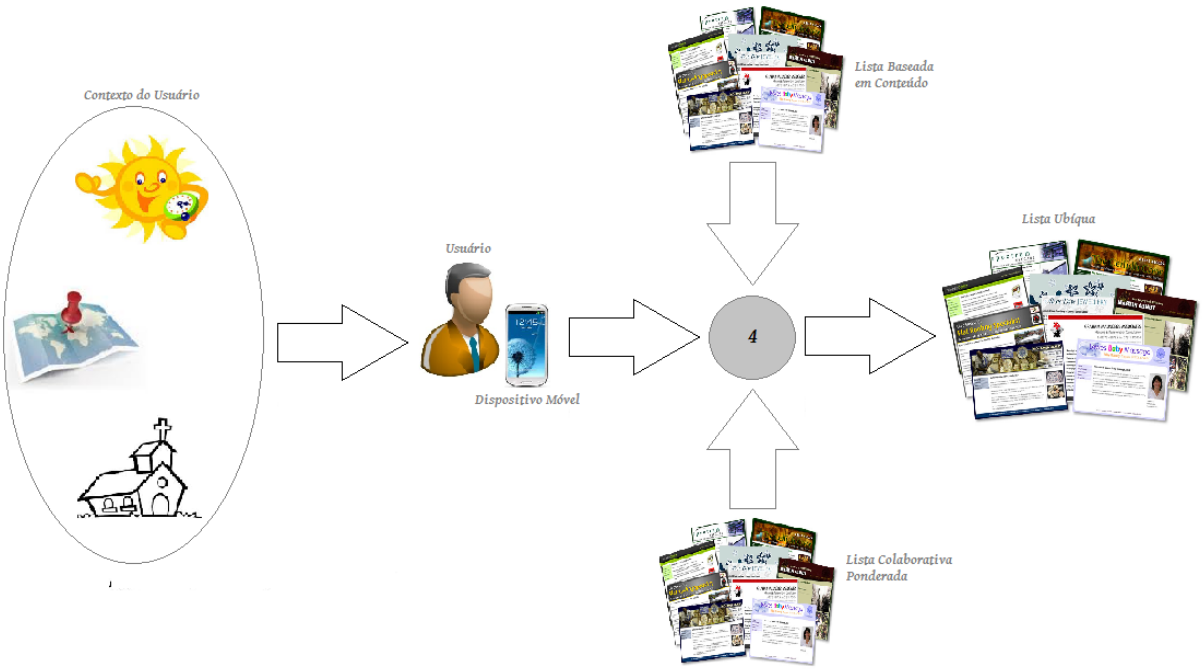

Figura 9 – Fluxo de Entrada e Saída da Recomendação Baseada na Ubiquidade
Fonte: Autor.

É importante ressaltar que a RBU não é executada sobre toda a base de dados, mas apenas sobre a Lista Baseada em Conteúdo e sobre a Lista Colaborativa Ponderada, garantindo assim a resposta em tempo hábil ao contexto do usuário e agregando um novo índice aos conteúdos dessas listas, o IRU.

Para o cálculo do IRU, leva-se em consideração os seguintes aspectos:

a) Frequência do Conteúdo (*S1*): Analisa os histogramas de aceitação, rejeição e visualização do conteúdo em relação ao horário considerado no contexto em que está inserido o usuário alvo. Se o conteúdo possui forte aceitação no dia da semana e no horário do contexto, isso incrementará a semelhança de frequência, no entanto se o conteúdo possui forte rejeição isso decrescerá o índice.

b) Histórico de Locais do Conteúdo (*S2*): Analisa o histórico de relações com locais do conteúdo e sua adequação ao local considerado no contexto do usuário alvo. Se o conteúdo possui forte aceitação no local do contexto do usuário, isso incrementará a semelhança de locais, no entanto se o conteúdo possui forte rejeição isso decrescerá o índice. Além disso, analisa a semelhança entre a lista de *tags* do local do contexto e do conteúdo alvo, onde *tags* semelhantes incrementam o índice de semelhança de locais do conteúdo.

c) Situação do Usuário (*S3*): Analisa a adequação do conteúdo em relação à velocidade em que está se locomovendo o usuário alvo. Para usuários em velocidades maiores serão priorizados conteúdos de áudio, posteriormente vídeos, e por último textos, com menor incremento.

d) Situação do Dispositivo (*S4*): Analisa a situação do dispositivo (percentual de bateria e velocidade da conexão com a internet) e sua adequação ao contexto para executar a visualização do conteúdo. Para conexões consideradas lentas, são priorizados conteúdos de pequeno tamanho em *bytes*, como textos, em detrimento dos vídeos, principalmente se forem longos.

O IRU do conteúdo "j" em relação ao contexto é calculado como sendo a média ponderada de cada uma das semelhanças descritas nos itens anteriores, de acordo com a Equação 6.

$$\mathrm{IRU}_j = \frac{\mathrm{S1}*\mathrm{P1} + \mathrm{S2}*\mathrm{P2} + \mathrm{S3}*\mathrm{P3} + \mathrm{S4}*\mathrm{P4}}{\mathrm{P1} + \mathrm{P2} + \mathrm{P3} + \mathrm{P4}}$$

Equação 6 – Cálculo do Índice de Recomendação Ubíqua – IRU

Onde S_k representa o nível de semelhança de determinada característica e P_k representa o peso dessa característica na avaliação.

Os *n* maiores resultados de IRU, vinculados aos seus respectivos pares, usuário e conteúdo, são armazenados como a Lista Baseada na Ubiquidade (LBU) de tamanho *n* de um usuário.

5.4 RECOMENDAÇÃO HÍBRIDA

A Recomendação Híbrida Ubíqua (RHU) é executada instantaneamente após a geração da LBU para um determinado usuário alvo, e calcula o Índice de Recomendação Híbrida (IRH).

A RHU gera a LHU que é uma lista de conteúdos vinculados aos seus respectivos IRHs, sendo estes resultantes da análise da Lista Colaborativa Ponderada, da Lista Baseada em Conteúdo e da Lista Baseada na Ubiquidade. A Figura 10 apresenta a geração da Lista Híbrida Ubíqua por meio da ponderação das listas Ubíqua, Baseada em Conteúdo e Colaborativa Ponderada, considerando os pesos (I_1, I_2, I_3) do usuário.

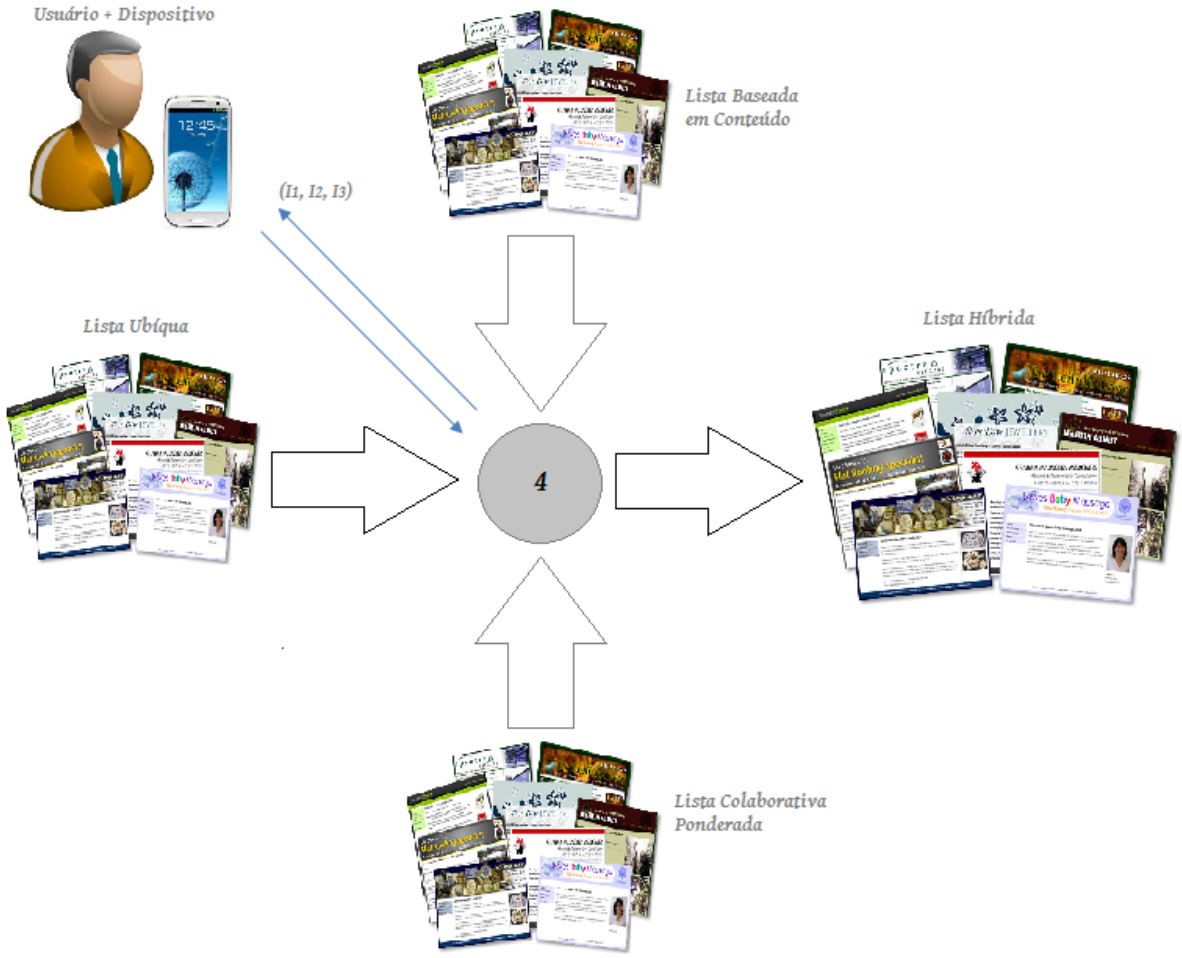

Figura 10 – Fluxo de Entrada e Saída da Recomendação Híbrida Ubíqua
Fonte: Autor.

O IRH, do conteúdo "j" para o usuário "i" considerando seu contexto, é calculado como sendo a média ponderada de cada um dos índices das listas descritas no item anterior, de acordo com a Equação 7.

$$\mathrm{IRH}_{i,j} = \frac{\mathrm{ICP}*\mathrm{P1} + \mathrm{IBC}*\mathrm{P2} + \mathrm{IBU}*\mathrm{P3}}{\mathrm{P1} + \mathrm{P2} + \mathrm{P3}}$$

Equação 7 – Cálculo do Índice de Recomendação Híbrido – IRH

Os *n* maiores resultados de IRH, vinculados aos seus respectivos pares, usuário e conteúdo, são armazenados na LHU de tamanho *n* de um usuário.

De acordo com os valores dos pesos P_1, P_2 e P_3, determinamos quanto a recomendação é mais colaborativa, baseada em conteúdo ou sensível ao contexto (ubíqua). Os valores desses pesos não são fixos e variam de acordo com as ações do usuário (aceitação ou rejeição) em relação a cada recomendação.

5.5 PERSONALIZAÇÃO AUTOMÁTICA DOS PESOS

Na primeira recomendação de um usuário, seus respectivos pesos são todos iguais a 5,0. O número de recomendações respondidas pelo usuário, denominado aqui de NR, também é armazenado e serve para o cálculo da adequação dos pesos, que ocorre de acordo com as equações 8, 9 e 10.

$$P1 = \frac{P1 * NR + k * ICP}{NR + 1}$$

Equação 8 – Cálculo do Peso da Recomendação Colaborativa

$$P2 = \frac{P2 * NR + k * IBC}{NR + 1}$$

Equação 9 – Cálculo do Peso da Recomendação Baseada em Conteúdo

$$P3 = \frac{P3 * NR + k * IBU}{NR + 1}$$

Equação 10 – Cálculo do Peso da Recomendação Ubíqua

O valor de *k* é igual a 1 se a recomendação for aceita e *k* é igual a -1 se a recomendação for rejeitada.

Após a ação de aceitação ou rejeição, por parte do usuário alvo, o valor de NR é incrementado em 1.

Dessa forma, os pesos se ajustam automaticamente as preferências de cada usuário, personalizando assim as futuras recomendações para os mesmos.

5.6 RECOMENDAÇÃO CONJUNTA

A Recomendação Conjunta utiliza-se de técnicas de formação de grupos, priorizando assim que elementos da recomendação final não disputem pelas mesmas características do usuário. Para isso, temos que analisar todas as possibilidades de combinações na lista de recomendação híbrida.

Se tem-se uma Lista Híbrida Ubíqua que possui *n* conteúdos e se deseja fazer uma recomendação de *p* conteúdos, o número *N* de combinações que deverão ser analisadas será calculado como apresentado na Equação 11.

$$N = \frac{n!}{p!(n-p)!}$$

Equação 11 – Cálculo do Número de Recomendações Possíveis

O caráter fatorial de *N* inviabiliza a possibilidade de retornarmos essa análise em tempo real. A complexidade da análise de todas as possibilidades cresce de forma fatorial, o que justifica o uso de uma técnica aproximativa, no caso AG, para escolha da melhor recomendação conjunta.

No exemplo a seguir, dada a LHU de conteúdos de tamanho *n = 100* a serem recomendados, teremos uma recomendação conjunta de tamanho *p = 5*.

Lista Híbrida Ubíqua

Conteúdo 1	Conteúdo 2	Conteúdo 3	Conteúdo 4	Conteúdo 5	...	Conteúdo 100
IRH=9,87	IRH=9,70	IRH=9,63	IRH=8,98	IRH=8,85	...	IRH=5,60

Recomendação Conjunta (Cromossomo melhor avaliado pela função *fitness*)

Conteúdo 23	Conteúdo 1	Conteúdo 39	Conteúdo 60	Conteúdo 3
IRH=8,01	IRH=9,87	IRH=7,64	IRH=6,18	IRH=9,63

Neste exemplo vimos que a recomendação conjunta não foi formada simplesmente escolhendo-se os melhores IRHs da Lista Híbrida Ubíqua, mas a análise foi feita considerando-se a melhor avaliação do cromossomo pela função *fitness*, que considerou critérios como a soma dos tempos, repetições de autores, subtipo, e defeitos genéticos (conteúdos repetidos).

A função *fitness* prioriza principalmente a diversidade, evitando assim que na lista final apareçam autores repetidos, ou apenas um subtipo de conteúdo (vídeo, texto ou áudio) ou mesmo que a soma dos tempos totais ultrapassem em muito o que é característico do usuário aceitar em um determinado intervalo de tempo. Dessa forma, objetiva-se alcançar uma maior aceitação das recomendações realizadas.

O modelo de AG desenvolvido possui as seguintes características:

f) Gene: Cada gene é representado por uma recomendação de conteúdo para o usuário alvo, escolhidos a partir da LHU;
g) Cromossomo (indivíduo): É um conjunto de genes que representam uma possível recomendação conjunta;
h) População Inicial: É gerada a partir da LHU, escolhendo-se aleatoriamente recomendações de conteúdo que serão os genes que formarão cada indivíduo;
i) Função *Fitness*: Avalia um cromossomo ou indivíduo, considerando os critérios de IRH, evitando as características conflitantes (subtipo, autor, tempo), e reduzindo a avaliação dos cromossomos com defeitos genéticos (genes repetidos);
j) Cruzamento: Para melhorar o processo, foi utilizado o conceito de dominância entre os genes, onde um gene (recomendação) com maior *rating* será dominante sobre um de menor.
k) Seleção: Os critérios de seleção foram estabelecidos com base em Estado Estacionário[7] e Elitismo[8], onde metade da população (melhor avaliada) sobrevive e gera mais uma metade de população nova (nova geração).
l) Mutação: A mutação se dá pela troca de posições de genes de um mesmo cromossomo, evitando assim a amarração em falsos máximo da função *fitness.*

A recomendação conjunta é o resultado de todo o sistema de recomendação, sendo essa o retorno da solicitação realizada pelo dispositivo móvel.

[7] Este não é um método particular de seleção de pais. A ideia principal deste tipo de seleção é que a nova população deve ter uma grande parte de cromossomos (menor *fitness*) que sobreviverão para a próxima geração, e os demais (menor *fitness*) são substituídos pela nova geração.
[8] Elitismo é o nome do método que primeiro copia os melhores cromossomos (ou os poucos melhores cromossomos) para a nova população. O resto da população é construída das formas descritas acima. Elitismo pode aumentar rapidamente o desempenho do AG, porque previne a perda da melhor solução já encontrada.

6 VALIDAÇÃO DO SISTEMA

O sistema de validação conta com quatro domínios e com a criação de dados fictícios de instâncias das classes *Person*, *Location* e *Content*.

Além disso, foi criado um gerador de contextos com o objetivo de povoar todas as relações e históricos das classe básicas. Esse gerador simula ações que disparam gatilhos de contexto que geram históricos e relações para os usuários, conteúdos e locais, preenchendo alguns de seus campos e servindo como base para o cálculo de outros. Esse gerador funciona aleatoriamente mas com algumas variáveis propositalmente direcionadas, de forma a garantir as características desejadas nos históricos de cada instância das classes básicas.

6.1 SIMULAÇÕES

Posteriormente à criação de toda a população simulada de instâncias das classes, assim como de seus históricos e relações, foram desenvolvidas simulações de casos de teste, onde foram analisados os seguintes casos para a recomendação final:

a) Em um contexto neutro, foi solicitada pelo sistema uma recomendação para usuários que têm históricos voltados a cada um dos quatro domínios instalados no sistema;
b) Em contextos voltados a cada um dos quatro domínios instalados no sistema, foi solicitada pelo sistema uma recomendação para um usuário que tem um histórico sem domínio definido.

Além da análise da recomendação final também foram analisadas as recomendações intermediárias isoladamente, sendo elas:

a) Lista de Colaboradores: Foi verificada a real semelhança na lista de colaboradores;
b) Lista Colaborativa Ponderada: Foi verificada a real semelhança dos conteúdos da lista colaborativa;
c) Lista Baseada em Conteúdo: Foi verificada a real semelhança na lista baseada em conteúdo;
d) Lista Ubíqua: Foi verificada a real adequação da lista ubíqua ao contexto.

6.2 AMOSTRA

Para simular o funcionamento completo, foram adicionados quatro domínios ao sistema, sendo eles: Ciência da Computação, Medicina, Direito e Ciências Contábeis. Após instalados os domínios, inseriu-se dados instanciando-se as classe básicas (*person*, *content* e *location*) de forma que 20% (vinte por cento) de cada uma delas assumissem características preponderantes de um dos domínios. Além disso, instanciou-se mais 20% (vinte por cento) com instâncias sem domínio definido, totalizando assim cinco segmentos instanciados.

Para satisfazer e dar maior credibilidade à amostra, os valores foram instanciados de forma a diminuir as margens de erro do acaso. Sendo assim, definiu-se que a amostra contaria com as dimensões e segmentos[9] apresentados na Tabela 4.

Tabela 4 – Dimensões e segmentos da amostra.

Classes Básicas	Total de Instâncias	Número de Segmentos
Person	200	5
Content	200	5
Location	200	5
Tags	100	5
RelationPersonPerson	5-10/*person*	4
RelationPersonContent	5-10/*person*	4
RelationPersonLocation	5-10/*person*	3
RelationContentLocation	5-10/*content*	3
RelationTags	5/entidade (*person*, *content*, *location*)	5

Fonte: Autor.

Para o desenvolvimento das simulações e posterior realização das análises desejadas em cada caso de teste, fez-se necessário que a amostra fosse propositalmente manipulada em algumas características pontuais. Dessa forma, manipulou-se as classes *Person*, e as escolhas de *Content* e *Location* da seguinte forma:

[9] Segmentos, no caso das instâncias básicas, representam um grupo de características direcionadas a desenvolver perfis voltados a um dos domínios instalados no sistema, e no caso das relações representam o tipo de relação ocorrida entre as instâncias.

a) Person: Em especial as instâncias da classe *Person* foram alteradas para apresentar as características conforme as tabelas 5, 6 e 7.

- Quanto à idade:

Tabela 5 – Distribuição do atributo idade.

Faixa de Idade	**Percentual**
10-24 anos	17,80%
25-39 anos	32,51%
40-55 anos	30,55%
54-69 anos	21,91%
70 Acima	5,03%
Total	**100%**

Fonte: Autor.

- Quanto ao sexo:

Tabela 6 – Distribuição do atributo sexo.

Faixa de Sexo	**Percentual**
Masculino	47,90%
Feminino	52,10%
Total	**100%**

Fonte: Autor.

- Quanto à renda:

Tabela 7 – Distribuição do atributo renda.

Faixa de Renda	**Percentual**
Sem Renda	33,85%
Até ½ Salário Mínimo	8,93%
De ½ a 1 Salário Mínimo	24,21%
De 1 a 2 Salários Mínimo	18,61%
De 2 a 5 Salários Mínimo	9,94%
Acima de 5 Salários Mínimo	4,46%
Total	**100%**

Fonte: Autor.

Os percentuais atribuídos às distribuições foram baseados nas distribuições das características da população brasileira de acordo com informações do Instituto Brasileiro de Geografia e Estatística (IBGE). Quanto à escolaridade e formação: Consideramos que todos os usuários são estudantes universitários das áreas de Ciência da Computação, Medicina, Direito ou Ciências Contábeis.

b) *Content:* As características de *Content* foram definidas de forma a se adequar igualmente aos domínios (segmentos) selecionados, sendo assim: Ciência da Computação (20%), Medicina (20%), Direito (20%), Ciências Contábeis (20%) e mais o segmento sem domínio definido (20%).

c) *Location:* As características de *Location* foram definidas de forma a se adequar igualmente aos domínios (segmentos) selecionados, sendo assim: Ciência da Computação (20%), Medicina (20%), Direito (20%), Ciências Contábeis (20%) e mais o segmento sem domínio definido (20%).

d) *Relation:* As relações foram definidas de modo a simular todas as possibilidades de disparo de contexto de acordo com as proporções das tabelas 8, 9, 10 e 11.

Tabela 8 – Distribuição das relações entre usuários.

Relação	*Status*	Percentual	Número de Relações
RelationPersonPerson	Início de Amizade	50%	12-25
	Solicitações de Amizade	10%	2-5
	Rejeições de Amizade	20%	5-10
	Exclusões de Amizade	20%	5-10

Fonte: Autor.

Tabela 9 – Distribuição das relações entre usuário e conteúdo.

Relação	*Status*	Percentual	Número de Relações
RelationPersonContent	Adicionou aos Favoritos	20%	5-10
	Visualização de Conteúdo	60%	15-30
	Rejeição de Conteúdo	10%	5-10
	Avaliação de Conteúdo	10%	5-10

Fonte: Autor.

Tabela 10 – Distribuição das relações entre usuário e local.

Relação	*Status*	Percentual	Número de Relações
RelationPersonLocation	Adicionou aos Favoritos	20%	5-10
	Presença no Local	60%	15-30
	Avaliação de Local	20%	10-20

Fonte: Autor.

Tabela 11 – Distribuição das relações entre conteúdo e local.

Relação	*Status*	Percentual	Número de Relações
RelationContentLocation	Sobre o Local	10%	2-5
	Cita o Local	30%	7-15
	Relacionado ao Local	60%	15-30

Fonte: Autor.

Os atributos ou características não citadas nesse subitem permaneceram iguais, com valores padrões, para que não interferiram nos algoritmos de recomendação.

7 ANALISE DOS RESULTADOS

A análise dos resultados finais foi realizada sobre cada um dos dois eventos elencados no capítulo anterior, e consistiu na análise da métrica *Precision.*

A metodologia utilizada e características inerentes ao próprio sistema anulam a viabilidade da análise de Negativos[10], Verdadeiros ou Falsos, pois como o número de conteúdos Verdadeiros e Falsos é bem maior que o tamanho da recomendação final é evidente que sempre o número de Negativos será alto, nesse sentido justificou-se a escolha da métrica *Precision*, que não utiliza Negativos, descartando-se *False Positive Rate* e *Recall*.

7.1 RESULTADOS

Para testar a viabilidade do sistema, foram realizados os seguintes casos de teste (simulações) para a recomendação final, encontrando-se os respectivos resultados:

a) Em um contexto neutro, foi solicitada pelo sistema uma recomendação para usuários que têm históricos voltados a cada um dos quatro domínios instalados no sistema. A Tabela 12 mostra o resultado da simulação.

Tabela 12 – Resultado da simulação com contexto neutro e usuário de perfil definido.

Número de Recomendações	Verdadeiros Positivos (VP)	Falsos Positivos (FP)
40	726	74

Fonte: Autor.

Aplicando-se as métricas selecionadas, temos:

$$\boldsymbol{Precision = \frac{nº\ de\ VP}{nº\ de\ VP + nº\ de\ FP} = \frac{726}{726 + 74} = 0,9075}$$

b) Em contextos voltados a cada um dos quatro domínios instalados no sistema, foi solicitada pelo sistema uma recomendação para um usuário que tem um histórico sem domínio definido. A Tabela 13 apresenta o resultado da simulação.

[10] Ver Subseção 2.6.6 - Avaliação de Sistemas de Recomendação.

Tabela 13 – Resultado da simulação com contexto definido e usuário de perfil neutro.

Número de Recomendações	Verdadeiros Positivos (VP)	Falsos Positivos (FP)
40	740	60

Fonte: Autor.

Aplicando-se a métrica selecionada, temos:

$$\boldsymbol{Precision = \frac{nº\ de\ VP}{nº\ de\ VP + nº\ de\ FP} = \frac{740}{740 + 60} = 0,9250}$$

Além da análise da recomendação final, também foram analisadas as recomendações intermediárias isoladamente, sendo elas:

a) Lista de Colaboradores: Foi verificada a real semelhança na lista de colaboradores. A Tabela 14 apresenta o resultado da simulação.

Tabela 14 – Resultado da simulação da lista de colaboradores.

Número de Recomendações	Verdadeiros Positivos (VP)	Falsos Positivos (FP)
2000	2000	0

Fonte: Autor.

Aplicando-se a métrica selecionada, temos:

$$\boldsymbol{Precision = \frac{nº\ de\ VP}{nº\ de\ VP + nº\ de\ FP} = \frac{2000}{2000 + 0} = 1,0}$$

b) Lista Colaborativa Ponderada: Foi verificada a real semelhança dos conteúdos da lista colaborativa. A Tabela 15 apresenta o resultado da simulação.

Tabela 15 – Resultado da simulação da lista colaborativa ponderada.

Número de Recomendações	Verdadeiros Positivos (VP)	Falsos Positivos (FP)
40	1829	171

Fonte: Autor.

Aplicando-se a métrica selecionada, temos:

$$\boldsymbol{Precision = \frac{nº\ de\ VP}{nº\ de\ VP + nº\ de\ FP} = \frac{1829}{1829 + 171} = 0,9145}$$

c) Lista Baseada em Conteúdo: Foi verificada a real semelhança na lista baseada em conteúdo. A Tabela 16 mostra o resultado da simulação.

Tabela 16 – Resultados da simulação da lista baseada em conteúdo.

Número de Recomendações	Verdadeiros Positivos (VP)	Falsos Positivos (FP)
40	2000	0

Fonte: Autor.

Aplicando-se a métrica selecionada, temos:

$$\boldsymbol{Precision = \frac{nº\ de\ VP}{nº\ de\ VP + nº\ de\ FP} = \frac{2000}{2000 + 0} = 1,0}$$

d) Lista Ubíqua: Foi verificada a real adequação da lista ubíqua ao contexto. A Tabela 17 apresenta o resultado da simulação.

Tabela 17 – Resultado da simulação da lista baseada na ubiquidade.

Número de Recomendações	Verdadeiros Positivos (VP)	Falsos Positivos (FP)
40	3122	78

Fonte: Autor.

Aplicando-se as métricas selecionadas, temos:

$$\boldsymbol{Precision = \frac{nº\ de\ VP}{nº\ de\ VP + nº\ de\ FP} = \frac{3122}{3122 + 78} = 0,9756}$$

7.2 ANÁLISE

Apesar da simulação ter sido realizada com dados fictícios, as métricas apresentaram resultados superiores a 90%. Acreditamos ainda que em uma simulação com um maior número de relações esses índices possam melhorar ainda mais.

Entretanto é importante salientar que na prática outros fatores podem influenciar a aceitação ou rejeição de uma recomendação, e que não são possíveis simular. Lembramos ainda que a validação com a métrica de *Precision* avalia apenas se a recomendação ocorreu de forma correta, não sendo isso argumento suficiente para aceitação por parte de um usuário.

8 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Atualmente o ambiente de aprendizagem tradicional tem passado por uma grande evolução e, nesse sentido, o avanço tecnológico tem caminhado em conjunto, criando soluções e mecanismos, diminuído as distâncias e proporcionando oportunidades aos que antes eram limitados por fatores físicos ou econômicos.

O EaD tem facilitado o acesso à informação e à formação, ganhando qualidade e, consequentemente, credibilidade. Vários sistemas de TIC surgiram no apoio a essa atividade, o que permitiu o avanço em muitas áreas e a disponibilização de ferramentas cada vez mais personalizadas e completas para apoiar o ensino e o estudante no que ele realmente precisa.

Por outro lado, a educação informal está se tornando uma grande aliada à formação, não só acadêmica, mas também social. Cada vez mais temos informação distribuída em todos os veículos e cada vez mais os veículos de comunicação tornam-se mais presentes na vida das pessoas, passando a ser portáteis e assim proporcionando a mobilidade e, consequentemente, a aprendizagem móvel.

Nos últimos anos tem-se percebido um esforço da comunidade acadêmica em disponibilizar sistemas que auxiliem a aprendizagem e, consequentemente, possa oferecer oportunidades personalizadas de acordo com o perfil de cada usuário, e não só isso, mas também que essas oportunidades sejam compatíveis com o contexto desse usuário.

Nesse sentido, o sistema MobiLE+, se propõe como solução para acesso e compartilhamento de conteúdo por dispositivos móveis, sendo sensível ao contexto do usuário e personalizado às suas características e históricos.

O SR do sistema MobiLE+, objeto dessa dissertação, foi desenvolvido de forma a contemplar as características de ubiquidade, mobilidades e personalização ao usuário. Como foi citado na introdução, alguns obstáculos aconteceram pelo o sistema está em construção, fato que dificultou a modelagem dos requisitos e a fase de teste, principalmente a possibilidade de uma validação real.

Entretanto, mesmo com os obstáculos apresentados, o SR foi modelado, desenvolvido, testado e validado, apresentando resultados convincentes em relação à métrica utilizada.

Nesse contexto, a contribuição desse trabalho, apresenta-se por:

- Mobilidade: o sistema é voltado para recomendação em dispositivos móveis, facilitando assim a educação informal com a possibilidade de aprendizagem em

qualquer lugar e a qualquer tempo;

- Ubiquidade: *U-Learning* é a junção das propriedades de mobilidade e sensibilidade ao contexto. O SR contempla essa característica utilizando-se de informações como hora do dia, dia da semana, geo-posicionamento e identificação do lugar onde está presente o usuário, identificação da situação do usuário e das características e estado do seu dispositivo móvel;
- SR Híbrido e Adaptação dos Pesos: O SR desenvolvido para o MobiLE+ abrange várias tecnologias e estratégias de recomendação, sendo estas: recomendação colaborativa, recomendação baseada em conteúdo, recomendação sensível ao contexto, recomendação conjunta, personalizadas ao perfil de aprendizagem do usuário e utilizando um sistema de enriquecimento semântico. Além disso, a recomendação híbrida ajusta seus pesos de acordo com as escolhas dos usuários, aprendendo com seu histórico de aceitação ou rejeição de conteúdo; e,
- Recomendação Conjunta: O SR, além de contemplar toda a gama de estratégias de recomendação do item anterior, ainda personaliza a recomendação conjunta, evitando assim que conteúdos semelhantes disputem pelas mesmas necessidades do usuário, facilitando assim a aceitação de vários itens.

Apesar de apresentar contribuições significativas, ainda pode-se projetar futuros trabalhos a serem desenvolvidos:

- Unificação do Sistema Completo: Finalização do MobiLE+ e integração de todos os módulos;
- Validação Real: Validação do sistema utilizando usuários, dispositivos, locais e conteúdos reais;
- Outras Métricas: Com uma validação real será possível a utilização de outras métricas que certifiquem o funcionamento e a qualidade do sistema;
- Inserir Outras Técnicas (personalidade): Junto ao enriquecimento semântico e à análise de metadados, deve-se considerar outras técnicas como análise de emoções e traços de personalidade; e,
- Captura Automática de Dados: Um sistema capaz de capturar de forma automática e implícita os dados sobre o usuário, evitando assim que o mesmo tenha que preencher formulários, muitas vezes extensos, pode proporcionar melhores resultados.

Por fim, o SR desenvolvido e o Sistema MobiLE+ ainda apresentam melhorias e

avanços possíveis, entretanto o SR apresentou características suficientes para o qual se propôs, tendo alcançando, portanto, êxito nos objetivos desta dissertação.

REFERÊNCIAS

ACOSTA, O.; REATEGUI, E. Recomendação de Conteúdo em Ambientes de Aprendizagem Baseados em Questionamento. **RENOTE - Revista Novas Tecnologias na Educação**, p. 1–9, 2012. Disponível em: <http://seer.ufrgs.br/renote/article/view/30936>. Acesso em: 16/1/2014.

AHMED, S. F.; CHANG, M. Recommend computer studies courses for students taken based on supported mobile learning modes. **2012 IEEE International Conference on Technology Enhanced Education (ICTEE)**, p. 1–5, 2012. Ieee. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6208652>. .

ÁVILA, C. M. O. **Um Sistema de Recuperação de Artigos Científicos Baseado em Consultas por Exemplo**, 2008.

BARCELLOS, C. D.; MUSA, D. L.; BRANDÃO, A. L.; WARPECHOWSKI, M. Sistema de Recomendação Acadêmico para Apoio a Aprendizagem. **RENOTE - Revista Novas Tecnologias na Educação**, 2007. Disponível em: <http://www.cinted.ufrgs.br/ciclo10/artigos/3fDaniela.pdf>. Acesso em: 6/2/2014.

BARRETO, R.; CAMPOS, P. D. E. ANÁLISE COMPARATIVA DE FRAMEWORKS. ,2010.

BOSSLER, A. P.; VENTURELLI, D. Sites e blogs : definição , conceitos e passo a passo. .

CASTILLO, S.; AYALA, G. ARMOLEO, an ARchitecture for MObile LEarning Objects based on our concept of mobile learning objects. Electronics, Communications and Computers, 2008. CONIELECOMP 2008, 18th International Conference on. **Anais...** , 2008.

CAZELLA, S. C.; AUGUSTA, M.; NUNES, S. N.; BERNI, E. A Ciência da Opinião: Estado da arte em Sistemas de Recomendação. **XXX Congresso da SBC Jornada de Atualização da Informática.**, p. 52, 2010.

COSTA, F. S. DA. **PerSoN: Um Modelo Semântico para Personalização no Âmbito das Redes Socias**, 2013. UERN - UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE.

DESPOTAKIS, D.; LAU, L.; DIMITROVA, V. Capturing the semantics of individual viewpoints on social signals in interpersonal communication. **Semantic Web Journal, Special Issue on Personal and Social Semantic Web**, 2011.

FERREIRA, V.; RAABE, A. LORSys–Um Sistema de Recomendação de Objetos de Aprendizagem SCORM. **RENOTE - Revista Novas Tecnologias na Educação**, v. 8 Nº2, 2010. Disponível em: <http://seer.ufrgs.br/renote/article/view/15223>. Acesso em: 15/1/2014.

GLUZ, J.; VICARI, R. Milos: Infraestrutura de agentes para suporte a objetos de aprendizagem obaa. XXI Simpósio Brasileiro de Informática na Educação, SBIE. **Anais...** , 2010. João Pessoa-PB. Disponível em: <http://ceie-sbc.tempsite.ws/pub/index.php/sbie/article/view/1450>. Acesso em: 15/1/2014.

GOLDBERG, D. E.; HOLLAND, J. H. Genetic Algorithms and Machine Learning. , p. 95–99, 1988.

HEINEN, M.; OSÓRIO, F. Algoritmos genéticos aplicados ao problema de roteamento de veículos. **HÍFEN**, v. 30, nº58, n. i, 2006. Disponível em: <http://revistaseletronicas.pucrs.br/ojs/index.php/hifen/article/view/3781>. Acesso em: 7/2/2014.

HERLOCKER, J. **Understanding and improving automated collaborative filtering systems**, 2000. UNIVERSITY OF MINNESOTA. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.5842&rep=rep1&type=pdf>. Acesso em: 16/1/2014.

HOLLAND, J. H. **Adaptation in natural and artificial systems**. Oxford, England., 1975.

JIUGEN, Y.; RUONAN, X.; XIAOQIANG, H. Constructing informal learning mode based on social software. **2011 6th International Conference on Computer Science & Education (ICCSE)**, p. 1227–1230, 2011. Ieee. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6028853>. .

LI, M.; OGATA, H.; HOU, B.; UOSAKI, N.; YANO, Y. Personalization in Context-aware Ubiquitous Learning-Log System. **2012 IEEE Seventh International Conference on Wireless, Mobile and Ubiquitous Technology in Education**, p. 41–48, 2012. Ieee. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6185078>. .

LINDEMANN, V. **Estilos de aprendizagem: buscando a sinergia**, 2008. Universidade Federal do Rio Grande do Sul.

LOPES, G. **Sistema de recomendação para bibliotecas digitais sob a perspectiva da web semântica**, 2007. Disponível em: <http://www.lume.ufrgs.br/handle/10183/10747>. Acesso em: 6/2/2014.

LOUREIRO, F.; ALFREDO, A.; AUGUSTO, R.; et al. Computação Ubíqua Ciente de Contexto: Desafios e Tendências. **XXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SRBC´09)**, 2009.

LUCAS, Diogo C. **Algoritmos Genéticos: Uma Introdução**. 2002. Disponível em: <http://www.inf.ufrgs.br/~alvares/INF01048IA/ApostilaAlgoritmosGeneticos.pdf>. acesso em: 03/01/2014.

MANDULA, K.; MEDA, S. R.; JAIN, D. K.; KAMBHAM, R. Implementation of Ubiquitous Learning System Using Sensor Technologies. **2011 IEEE International Conference on Technology for Education**, Jul. 2011. India: Ieee. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6004374>. .

MARTINS, H. N. J.; COSTA, E. B.; OLIVEIRA, T. T. M.; SILVA, A. P.; BITTENCOURT, I. I. Sistema de Recomendacão Híbrido para Bibliotecas Digitais que Suportam o Protocolo OAI-PMH. XXII Simpósio Brasileiro de Informática na Educação, SBIE. **Anais...** p.140–149, 2011. Aracajú-SE.

MITCHELL, M. Introduction to Genetic Algorithms. ,1998. Berlin, Heidelberg: Springer Berlin Heidelberg. Disponível em: <http://link.springer.com/10.1007/978-3-540-73190-0>. .

MOORE, P.; HU, B. " Intelligent Context " for Personalized Mobile Learning. p.236–238, 2010.

MONTEIRO, Bruno. **YouUbi Project - Visão Geral**. YouUbi. 2014. Disponível em: <www.youubi.com>. Acesso em: 05 de jan. 2014.

NEVES, D.; BRITO, P. F. DE. Desenvolvimento de um aplicativo de recomendação de artigos científicos para materiais didáticos. , p. 63–72, 2012.

NUNES, M.; BORGES, T. Técnica para Analisar a Evolução do Perfil do Usuário com base nas suas Publicações. II Escola Regional de Banco de Dados. **Anais...** , 2006. Passo Fundo. RS. Disponível em: <http://paginas.ucpel.tche.br/~loh/pdfs/erbd-2007-marcos.pdf>. Acesso em: 6/2/2014.

PACHECO, M. Algoritmos genéticos: princípios e aplicações. **… Elétrica. Pontifícia Universidade Católica do Rio …**, 1999. Disponível em: <http://www.ica.ele.puc-rio.br/downloads/38/ce-apostila-comp-evol.pdf>. Acesso em: 6/2/2014.

REIS, L. F. M. **Sistema de Recomendação Baseado em Conhecimento**, 2012.

SACCOL, A., SCHLEMMER, E., BARBOSA, J. **M Learning e U Learning: Novas perspectivas da aprendizagem móvel e ubíqua**. São Paulo-SP: Person, 2010.

THAKKER, D.; DESPOTAKIS, D.; DIMITROVA, V.; LAU, L.; BRNA, P. Taming Digital Traces for Informal Learning : A Semantic-driven Approach. **21st Century Learning for 21st Century Skills. Lecture Notes in Computer Science**, v. v. 7563, p. p. 348–362, 2012.

TREVISAN, L.; CAZELLA, S.; NUNES, M. Aplicando Traços de Personalidade e Contextos em Sistemas de Recomendação para TV Digital: um facilitador do processo de ensino-aprendizagem. **Semantic Web Journal, Special Issue on Personal and Social Semantic Web**, p. 1792–1799, 2012. Disponível em: <http://200.17.141.213/~gutanunes/hp/publications/trevisan.pdf>. Acesso em: 6/2/2014.

VIEIRA, F. J. R.; NUNES, M. A. S. N. Sistema de Recomendação de Objetos de Aprendizagem Baseado em Conteúdo. **Scientia Plena**, v. v. 8, 2012.

WANG, M.; SHEN, R. Message design for mobile learning: Learning theories, human cognition and design principles. **British Journal of Educational Technology**, v. v43 n4, p. p561–575, 2012. Disponível em: <http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8535.2011.01214.x/full>. Acesso em: 6/2/2014.

YAU, J. Y. K.; JOY, M. A context-aware personalised m-learning application based on m-learning preferences. **International Journal of Mobile Learning and Organisation**, v. 5, n. 1, p. 1, 2011. Disponível em: <http://www.inderscience.com/link.php?id=38688>. .

APÊNDICE A - Algoritmos em Descrição Narrativa do SRHU

Sistema de Recomendação Híbrido Ubíquo

Inicio () {

```
// Gera lista de colaboradores para todas as pessoas...
geraListaColaboradores()
// Gera lista colaborativa ponderada para todas as pessoas...
geraListaColaborativaPonderada()
// Gera lista baseada em conteúdo para todas as pessoas...
geraListaBaseadaConteudo()
// Gera lista ubíqua para uma pessoa alvo...
geraListaUbiqua()
// Gera lista híbrida para uma pessoa alvo...
geraListaHibrida()
// Gera lista conjunta para uma pessoa alvo...
geraListaConjunta()
// Retorna o resultado
retorna listaConjunta
```

Fim ().

Geração da Lista de Colaboradores
Inicio () {

// Ler os dados iniciais de todas as pessoas

lerTodasPessoas()

// Compara a semelhança entre todas as pessoas

comparaTodasPessoas() {

comparaMetaDados()

comparaAmigos()

comparaConteudos()

comparaLocais()

comparaFrequencia()

comparaTags()

comparaSemantica()

}

// Calcula semelhaças entre as pessoas

calculaSemelhanca() {

$$ISU = \frac{S1*P1 + S2*P2 + S3*P3 + S4*P4 + S5*P5 + S6*P6+S7*P7}{P1 + P2 + P3 + P4 + P5 + P6 + P7}$$

}

// Seleciona os mais semelhantes entre si para compor as suas listas de colaboradores

listaColaborativa = MaioresISU()

// Retorna a listaColaborativa como resultado

retorna listaColaborativa

Fim ().

Geração da Lista de Colaborativa Ponderada

Inicio () {

// Ler os dados iniciais de todas as pessoas e conteúdos

lerTodasPessoas()

lerListaColaboradores()

lerTodosConteudos()

// Compara a semelhança entre todas as pessoas e os conteúdos

calculaIndiceColaborativo() {

$$ISC_k = \frac{1}{t}\sum_{j=0}^{n-1} A_{j,k*} ISU_{i,j}$$

}

// Seleciona os mais semelhantes entre si para compor as suas listas colaborativas

listaColaborativaPonderada = MaioresISC()

// Retorna a listaColaborativa como resultado

retorna listaColaborativaPonderada

Fim ().

Geração da Lista de Baseada em Conteúdo

Inicio () {

// Ler os dados iniciais de todas as pessoas e conteúdo

lerTodasPessoas()

lerTodosConteudos()

// Compara a semelhança entre todas as pessoas e conteúdos

comparaTodasPessoas() {

comparaMetaDados()

comparaTags()

comparaLocais()

comparaFrequencia()

comparaTags()

comparaSemantica()

}

// Calcula semelhaças entre as pessoas

calculaSemelhanca() {

$$IBC = \frac{S1*P1 + S2*P2 + S3*P3 + S4*P4 + S5*P5}{P1 + P2 + P3 + P4 + P5}$$

}

// Seleciona os mais semelhantes entre si para compor as suas listas baseadas em conteúdo.

listaBaseadaConteudo = MaioresIBC()

// Retorna a listaColaborativa como resultado

retorna listaBaseadaConteudo

Fim ().

Geração da Lista de Ubíqua

Inicio () {

// Recebe os dados iniciais de uma pessoas

lerPessoa()

//Captura o Contexto

CapturaContexto()

// Compara a semelhança entre todas as pessoas

comparaTodasPessoas() {

comparaConteudos()

comparaLocais()

comparaFrequencia()

comparaDispositivo()

}

// Calcula semelhaças entre as pessoas

calculaSemelhanca() {

$$IRU = \frac{S1*P1 + S2*P2 + S3*P3 + S4*P4}{P1 + P2 + P3 + P4}$$

}

// Seleciona os mais semelhantes entre si para compor as suas listas baseadas na ubiquidade.

listaUbiqua = MaioresIRU()

// Retorna a listaUbiqua como resultado

retorna listaUbiqua

Fim ().

Geração da Lista de Híbrida

Inicio () {

// Recebe as listas geradas

lerListaColaborativaPonderada()

lerListaBaseadaConteudo()

lerListaUbiqua()

// Compara a semelhança híbrida

calculaIndiceHibrido() {

$$IRH = \frac{ICP*P1 + IBC*P2 + IBU*P3}{P1 + P2 + P3}$$

}

// Seleciona os mais semelhantes entre si para compor as suas listas baseadas na ubiquidade.

listaHibrida = MaioresIRH()

// Retorna a listaHibrida como resultado

retorna listaHibrida

Fim ().

Geração da Lista de Conjunta

```
Inicio () {
        // Recebe a lista Híbrida
        lerListaHíbrida()
        // Escolhe qual melhor combinação de itens para ser recomendados em conjunto utilizando de um Algoritmo Genetico
        geraPopulaçãoInicial()
        algoritmoGenetico() {
                avaliarPopulacao()
                selecionarPais()
                cruzar()
                mutar()
                avaliadaCondicaoParada()
        }
        // Escolhe o melhor cromossomo para ser a lista conjuta
        listaConjunta = melhorCromossomo()
        // Seleciona os mais semelhantes entre si para
        retorna listaConjunta
Fim ().
```