

MEC-SETEC
INSTITUTO FEDERAL MINAS GERAIS - Câmpus Formiga
Curso de Ciência da Computação

**IMPLEMENTAÇÃO E ANÁLISE EXPERIMENTAL DE UMA
MÁQUINA DE BUSCA A DOCUMENTOS PDF**

Roger Santos Ferreira

Orientador: Prof. Me. Diego Mello da Silva.

FORMIGA- MG
2016

ROGER SANTOS FERREIRA

**IMPLEMENTAÇÃO E ANÁLISE EXPERIMENTAL DE UMA
MÁQUINA DE BUSCA A DOCUMENTOS PDF**

Trabalho de Conclusão de Curso apresentado ao Instituto Federal Minas Gerais - Câmpus Formiga, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Diego Mello da Silva.

FORMIGA-MG
2016

F383i Ferreira, Roger Santos

Implementação e análise experimental de uma máquina de busca a documentos PDF / Roger Santos Ferreira. – Formiga, MG., 2016.

165p.: il.

Orientador: Prof. M.e Diego Melo da Silva

Trabalho de Conclusão de Curso – Instituto Federal Minas Gerais – Campus Formiga.

1. Recuperação da informação. 2. Ranking. 3. Busca. 4. PDF. I. Silva, Diego Mello da. II. Título.

025.04

CDD

ROGER SANTOS FERREIRA

IMPLEMENTAÇÃO E ANÁLISE EXPERIMENTAL DE UMA MÁQUINA DE BUSCA A DOCUMENTOS PDF

Trabalho de Conclusão de Curso apresentado ao
Instituto Federal de Minas Gerais-Câmpus Formiga,
como requisito parcial para obtenção do título de
Bacharel em Ciência da Computação.

Aprovado em: 18 de FEVEREIRO de 2016.

BANCA EXAMINADORA



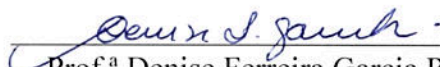
Prof. Diego Mello da Silva - Orientador



Prof. Mário Luiz Rodrigues Oliveira



Prof. Otávio de Souza Martins Gomes



Prof.^a Denise Ferreira Garcia Rezende

Dedico este trabalho a todos aqueles que verdadeiramente amam o estudo, a ciência, a prática do conhecimento e a valorização do saber.

AGRADECIMENTOS

O agradecimento é um meio de reconhecer que as dificuldades enfrentadas na vida são apenas percalços e que, mesmo assim, sozinho ninguém conseguiria superá-los. Cada fase desse caminho só foi possível de superar por que tive ao meu lado pessoas maravilhosas a quem só tenho a agradecer.

Agradeço primeiramente a Deus, toda honra e toda glória sejam atribuídas ao todo poderoso Senhor de nossas vidas; e ao seu filho, Jesus Cristo, expressão maior de amor incondicional pela humanidade - o meu muito obrigado pela dádiva da vida, da saúde e da racionalidade.

À minha esposa, Lívia, que tem perpetuado seus laços de amor, paciência, longanimidade, compreensão e zelo pelo caminhar de nossas vidas a dois. Entendo que não seja fácil partilhar de tudo isso!

Aos meus familiares, agradeço por todo o apoio, dedicação e amor de sempre.

Ao meu orientador prof. Diego, que sempre me apoiou e, mesmo na falta de tempo, conseguiu atender ao extenso trabalho na finalização deste estudo. Meu muito obrigado e admiração são nada perto do que tem feito por mim e pela primeira turma do curso de Bacharelado em Ciência da Computação do IFMG - Câmpus Formiga. Saiba que você é fonte de inspiração para muitos dessa turma.

Ao meu companheiro de longas horas de estudo e programação, Bonny! Te agradeço por me acompanhar e por sua incondicional amizade.

Aos meus amigos e companheiros de trabalho, agradeço pela atenção e paciência com minha rotina meio louca de estudos/trabalho. Espero poder retribui-lhes algum dia.

Aos colegas de classe da primeira turma de Ciência da Computação e em especial ao Raí, amigo e companheiro de trabalhos e estudo durante todo o curso. Desejo todo o sucesso em sua carreira, você é muito capaz.

À equipe de professores e servidores do IFMG - Câmpus Formiga, que com grande zelo me proporcionou esta grande oportunidade na vida - cursar Ciência da Computação, meu muito obrigado!

Os computadores são incrivelmente rápidos, precisos e burros; os homens são incrivelmente lentos, imprecisos e brilhantes; juntos, seu poder ultrapassa os limites da imaginação.

Albert Einstein (*apud* MEYER, 1996)

RESUMO

A preocupação com a recuperação de informações em sistemas computacionais precede até mesmo o surgimento dos computadores pessoais. O presente trabalho busca apresentar o estado da arte fundamental para a implementação de Sistemas de Recuperação da Informação, bem como apresentar conceitos inerentes à criação e manutenção de índices de documentos digitais em formato PDF. São abordados os modelos computacionais de Recuperação da Informação clássicos, a saber: modelo Booleano, modelo Vetorial e modelo Probabilístico. O estudo ainda aborda a análise experimental realizada em um *corpus* composto por 200 artigos científicos da área da Ciência da Computação e afins, julgados por profissionais especialistas com relação à sua relevância a 12 expressões de busca pré-definidas. Por fim, resultados da análise experimental são apresentados e discutidos sob o ponto de vista de como os modelos Booleano, Vetorial e Probabilístico se comportaram no referido *corpus*. A saber, a análise de relevância dos modelos testados experimentalmente apresentou o modelo Probabilístico com melhor desempenho, seguido do modelo Vetorial e por fim o modelo Booleano. A área de Recuperação da Informação é extensa e de forma alguma se limita ao conteúdo presente neste estudo, pois está em constante expansão.

Palavras-chave: Recuperação da Informação; *Ranking*; Busca; Indexação; PDF.

ABSTRACT

The concern about information retrieval in computer systems precedes even the arising of personal computers. This study aims to present the fundamental state of the art to the implementation of the Information Retrieval Systems and present concepts inherent in creating and maintaining digital document index in PDF format. Are covered classic computer Information Recovery models, namely: Boolean model, Vector model and Probabilistic model. The study also covers the experimental analysis on a corpus of 200 scientific papers in the area of Computer Science and related, judged by professional experts with regard to their relevance to 12 pre-defined search expressions. Finally, experimental results of the analysis are presented and discussed from the point of view of how Boolean, Vector and Probabilistic models behaved in that corpus. To whom it may concern, the relevance analysis of the experimentally tested models gives the Probabilistic model with the best performance, followed by the Vector model and finally the Boolean model. The Information Retrieval area is extensive and by no means limited to the content contained in this study, after all it is constantly expanding.

Keywords: Information Retrieval; Ranking; Search; Indexing; PDF.

LISTA DE ILUSTRAÇÕES

Gráfico 1 – Curvas de cobertura e precisão representando duas funções de ranking	63
Gráfico 2 – Exemplo de gráfico de cobertura e precisão média interpolada.....	65
Gráfico 3 – Os mais populares frameworks PHP até fevereiro de 2015, segundo o GitHub...	70
Gráfico 4 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 1	124
Gráfico 5 – Tempo de execução em segundos da consulta 1 nos três modelos de RI implementados.....	124
Gráfico 6 – Tempo de busca aferido para cada consulta por cada modelo de RI	125
Gráfico 7 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 2	126
Gráfico 8 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 3	127
Gráfico 9 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 4	129
Gráfico 10 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 5	130
Gráfico 11 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 6	131
Gráfico 12 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 7	132
Gráfico 13 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 8	134
Gráfico 14 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 9	135
Gráfico 15 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 10	136
Gráfico 16 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 11	137
Gráfico 17 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 12	138
Quadro 1 – Função de <i>ranking</i> por modelo de RI.....	40
Quadro 2 – Representações no modelo Booleano	42

Quadro 3 – Representações no modelo Vetorial	46
Quadro 4 – Tabela auxiliar de incidência de termos	51
Quadro 5 – Exemplo do <i>benchmark</i> TREC 2007 Enterprise Track	57
Quadro 6 – Exemplo dos cálculos para construção da curva de cobertura e precisão média interpolada em 11 pontos.....	65
Quadro 7 – Materiais e métodos para o desenvolvimento do estudo	67
Quadro 8 – Estruturada resposta do servidor à camada de visão do SRI	80
Quadro 9 – Estatísticas de implementação do trabalho proposto	97
Quadro 10 – Expressões de busca definidas para o julgamento de relevância do corpus	115
Figura 1 – Representação do processo de Recuperação da Informação	19
Figura 2 – Processo de Recuperação da Informação	24
Figura 3 – Componentes de um Sistema de Recuperação da Informação.....	25
Figura 4 – Representação lógica de documentos e consultas	27
Figura 5 – Matriz de incidência termo-documento	28
Figura 6 – Matriz de frequência termo-documento	29
Figura 7 – Atribuição de peso ao termo "do"	30
Figura 8 – Atribuição de pesos por <i>tf</i> , <i>idf</i> e <i>tf-idf</i>	31
Figura 9 – Tipos comuns de índice.....	34
Figura 10 – Processo de indexação	36
Figura 11 – Processo de especificação de consulta	37
Figura 12 – Definição formal de um modelo de Recuperação da Informação.....	39
Figura 13 – Combinações booleanas de conjuntos visualizadas como diagramas de Venn	41
Figura 14 – Medida de similaridade por cosseno no modelo Vetorial.....	43
Figura 15 – <i>Ranking</i> da pesquisa pela expressão "to do"	46
Figura 16 – Subconjuntos de documentos após consulta em um SRI probabilístico	48
Figura 17 – <i>Corpus</i> composto por seis documentos e dez termos.....	52
Figura 18 – Etapa de <i>relevance feedback</i>	53
Figura 19 - Resultado da segunda iteração do modelo Probabilístico.....	53
Figura 20 – Cobertura e precisão.....	59

Figura 21 – Curva de cobertura e precisão	61
Figura 22 – Gráfico de cobertura e precisão com valores de exemplo.....	62
Figura 23 – Cálculo de cobertura e precisão interpolada	66
Figura 24 – Adaptação do diagrama de classes UML modelado	73
Figura 25 – Interface do usuário com o SRI: resultado da renderização de ir.blade.php pelo navegador <i>web</i>	76
Figura 26 – Diagrama de entidade-relacionamento do SRI implementado.....	88
Figura 27 – Exemplo de árvore gerada pela consulta IFMG AND (recuperação OR informação) NOT água.....	100
Figura 28 – Interface web para julgamento de relevância do corpus	116
Figura 29 – Janela modal de classificação dos artigos	117
Código-fonte 1 – Conteúdo da camada de visão, representado pelo documento ir.blade.php.	74
Código-fonte 2 – JavaScript e chamadas AJAX ao SRI: arquivo ir.js	77
Código-fonte 3 – O controlador IRController	81
Código-fonte 4 – Rota de downloads de documentos PDF.....	82
Código-fonte 5 – Classe SRI	83
Código-fonte 6 – Exemplo de <i>migration</i> para criação do banco de dados do SRI.....	90
Código-fonte 7 – Exemplo de <i>seeder</i> para alimentação dos documentos do SRI.....	91
Código-fonte 8 – Exemplo de <i>seeder</i> para alimentação do índice	94
Código-fonte 9 – Exemplo de adição de termo ao índice.....	96
Código-fonte 10 – Exemplo de adição de indexação de termo-documento ao índice	96
Código-fonte 11 – Processo de especificação de consulta no modelo Booleano.....	98
Código-fonte 12 – Montagem da árvore binária de busca no modelo Booleano	99
Código-fonte 13 – Montagem da árvore binária de busca no modelo Booleano	101
Código-fonte 14 – Métodos de manipulação de conjuntos	101
Código-fonte 15 – Processo de especificação de consulta no modelo Vetorial	103
Código-fonte 16 – Montagem da matriz de consulta no modelo Vetorial	104
Código-fonte 17 – Método de cálculo do <i>tf-idf</i>	105
Código-fonte 18 – Normalização das componentes de um vetor: criação de versor	106

Código-fonte 19 – Método de cálculo da proximidade de um vetor documento ao vetor consulta.....	107
Código-fonte 20 – Método de busca vetorial	108
Código-fonte 21 – Processo de especificação de consulta no modelo Probabilístico	109
Código-fonte 22 – Métodos auxiliares da classe Indice	110
Código-fonte 23 – Método de atribuição de pesos BM25.....	110
Código-fonte 24 – Método de busca do modelo Probabilístico	113

LISTA DE TABELAS

Tabela 1 – Esquemas recomendados de atribuição de pesos a termos por <i>tf-idf</i>	33
Tabela 2 – Exemplo de julgamento de relevância	56
Tabela 3 – Resultado do julgamento de relevância pela equipe de especialistas	118
Tabela 4 – Resultado de cobertura e precisão para os três modelos clássicos implementados	119
Tabela 5 – Primeiros dez documentos recuperados pelo modelo Vetorial para a consulta 1.	120
Tabela 6 – Resultado final para o cálculo da precisão média interpolada em 11 pontos	121
Tabela 7 – Dados para a construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos.....	123
Tabela 8 – Dados para construção comparativa entre três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 2.....	126
Tabela 9 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 3.....	127
Tabela 10 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 4.....	128
Tabela 11 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 5.....	129
Tabela 12 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 6.....	131
Tabela 13 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 7.....	132
Tabela 14 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 8.....	133
Tabela 15 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 9.....	134
Tabela 16 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 10.....	135
Tabela 17 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 11.....	137
Tabela 18 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 12 pontos da consulta 11.....	138
Tabela 19 – Resumo comparativo de desempenho da precisão interpolada dos modelos de RI implementados, considerando-se total de posições ocupado	139

LISTA DE SIGLAS E ABREVIATURAS

a.C. – antes de Cristo

AJAX – *Asynchronous Javascript and XML*

API – *Application Programming Interface*

BDBComp – Biblioteca Digital Brasileira de Computação

BM25 – *Best Match 25*

BPM – *Business Process Management*

CB – Computação Brasil

CE – *Community Edition*

CI – Ciência da Informação

CLEF – *Cross Language Evaluation Forum*

COLD/ERM – *Computer Output to Laser Disk/Enterprise Report Management*

DDR3 – *Double Data Rate 3*

DER – Diagrama de Entidade-Relacionamento

df – *document frequency*

DI – *Document Imaging*

DM – *Document Management*

DOM – *Document Object Model*

GB – *Gigabyte*

GED – Gestão Eletrônica de Documentos

GHz – *GigaHertz*

HP – Hewlett Packard

HTML – *HyperText Markup Language*

HTTP – *HyperText Transfer Protocol*

ICR – *Intelligent Character Recognition*

ID – *identifier*

idf – *inverse document frequency*

IFMG – Instituto Federal de Minas Gerais

INFOCOMP – *Journal of Computer Science*

IP – Informática Aplicada

JBCS – *Journal of the Brazilian Computer Society*

JICS – *Journal of Integrated Circuits and Systems*

JSON – *JavaScript Object Notation*
MAP – *Mean Average Precision*
MRR – *Mean Reciprocal Rank*
MSDN – *Microsoft Developer Network*
MVC – *Model-View-Controller*
NDCG – *Normalized Discount Cumulative Gain*
NTCIR – *NII Test Collections for IR Systems*
OCR – *Optical Character Recognition*
ORM – *Object Relational Mapper*
P@5 – *Precision at Five*
P@10 – *Precision at Ten*
PaaS – *Plataform-as-a-Service*
PC – *Personal Computer*
PDF – *Portable Document Format*
PHP – *PHP: Hypertext Preprocessor*
RAM – *Random Access Memory*
RB-RESO – *Revista de Redes de Computadores e Sistemas Distribuídos*
RBIE – *Revista Brasileira de Informática na Educação*
REC – *Conjunto dos documentos recuperados*
REL – *Conjunto dos documentos relevantes*
REST – *Representational State Transfer*
RITA – *Revista de Informática Teórica e Aplicada*
RPM – *Rotations per Minute*
RR – *Conjunto dos documentos recuperados e relevantes*
RI – *Recuperação da Informação*
RIM – *Records and Information Management*
SDK – *Software Development Kit*
SGBD – *Sistema Gerenciador de Bancos de Dados*
SI – *Sistema de Informação*
sim – *similaridade*
SOAP – *Simple Object Access Protocol*
SRI – *Sistema de Recuperação da Informação*
SSL – *Secure Socket Layer*

TB – *Terabyte*

tf – *term frequency*

TREC – *Text Retrieval Conference*

UML – *Unified Modeling Language*

URL – *Uniform Resource Locator*

UNLV – *University of Nevada Las Vegas*

WWW – *World Wide Web*

XML – *eXtensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO	18
2 SISTEMAS DE RECUPERAÇÃO DA INFORMAÇÃO	21
2.1 Gestão eletrônica de documentos	21
2.2 Recuperação da Informação	22
2.3 Sistemas de Recuperação da Informação	24
2.4 Fundamentos de Recuperação da Informação	26
2.4.1 <i>Representação lógica de documento</i>	26
2.4.2 <i>Matriz de frequência termo-documento</i>	28
2.4.3 <i>Atribuição de peso aos termos</i>	29
2.4.4 <i>Índice de termos</i>	33
2.5 Processo de indexação	35
2.6 Processo de especificação de consulta	36
2.7 Modelos de Recuperação da Informação	38
2.7.1 <i>Modelo Booleano</i>	40
2.7.2 <i>Modelo Vetorial</i>	42
2.7.3 <i>Modelo Probabilístico</i>	47
2.8 Avaliação em Recuperação da Informação	55
2.8.1 Medidas para avaliação de SRIs	59
3 MATERIAIS E MÉTODOS	67
3.1 Plataforma de <i>hardware</i> e <i>software</i>	67
3.1.1 <i>Framework Laravel</i>	69
3.2 Máquinas de busca	71
3.2.1 <i>Diagrama de classes</i>	72
3.2.2 <i>Camada de Visão</i>	74
3.2.3 <i>Camada de Controle</i>	81
3.2.4 <i>Camada de Modelo</i>	83
3.2.4.1 O Sistema de Recuperação da Informação implementado	80
3.2.4.2 Construção do índice invertido	83
3.2.4.2.1 <u>Diagrama de entidade-relacionamento</u>	83
3.2.4.2.2 <u>Persistência</u>	85
3.2.4.2.3 <u>Alimentação dos documentos para composição do <i>corpus</i></u>	86
3.2.4.2.4 <u>Reconhecimento de texto em arquivos PDF via OCR</u>	87
3.2.4.2.5 <u>Processo de indexação</u>	89
3.3 Estatísticas de implementação	97
3.4 Implementação dos modelos clássicos	98
3.4.1 <i>Modelo Booleano</i>	98
3.4.2 <i>Modelo Vetorial</i>	102
3.4.3 <i>Modelo Probabilístico</i>	108

3.5 Avaliação experimental	113
<i>3.5.1 Coleção de referência</i>	<i>113</i>
<i>3.5.2 Experimentação</i>	<i>118</i>
4 RESULTADOS E DISCUSSÃO	123
4.1 Consulta 1	123
4.2 Consulta 2	125
4.3 Consulta 3	127
4.4 Consulta 4	128
4.5 Consulta 5	129
4.6 Consulta 6	130
4.7 Consulta 7	132
4.8 Consulta 8	133
4.9 Consulta 9	134
4.10 Consulta 10	135
4.11 Consulta 11	136
4.12 Consulta 12	138
4.13 Resumo comparativo	139
5 CONSIDERAÇÕES FINAIS	141
REFERÊNCIAS	143
APÊNDICE A - Lista de artigos indexados e sua classificação por especialistas	150
APÊNDICE B – Tabela de apoio ao cálculo da precisão média interpolada em 11 pontos	161

1 INTRODUÇÃO

Recuperação da Informação (RI) é um campo de grande relevância e estudo dentro da Ciência da Computação, caracterizado pela representação, busca e manipulação de extensas coleções de texto em formato digital. Sistemas de Recuperação da Informação (SRI) estão atualmente difundidos no mundo, com milhões de pessoas dependendo de seus serviços diariamente no auxílio às atividades de trabalho, educação e entretenimento (BÜTTCHER; CLARKE; CORMACK, 2010, p. 2).

De acordo com Kowalski (2011, p. 1), sendo uma ferramenta de utilização diária, os SRIs têm a atenção de toda uma geração de pessoas que claramente se perguntaram em algum momento: “Será que realmente funciona?”. Não é casual encontrarmos pessoas dizendo que não encontraram aquilo o que buscavam em meio à uma infinidade de resultados, ou seja, sua necessidade informacional não foi suprida em determinada busca. Assim, acima de tudo, um SRI deve ter seus resultados avaliados, fornecendo-se um modo de se quantificar sua precisão e cobertura em comparação à expressão buscada, o que de fato se caracteriza como a relevância dos resultados à necessidade informacional do usuário (BAEZA-YATES; RIBEIRO-NETO, 1999, p. 13).

Recuperação da Informação abrange também a Ciência da Informação (CI), fazendo o intercâmbio entre o tratamento e as representações da informação. SRIs têm sua existência justificada devido ao grande volume de dados e, conseqüentemente, informação gerada em tempos atuais, o que demanda uma gestão de forma eficiente, facilitada e dinâmica.

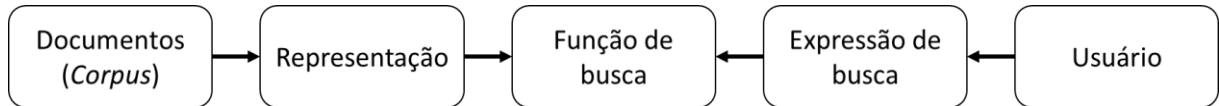
De modo conceitual, não se deve confundir os termos dados e informação, assim como dito por Baeza-Yates e Ribeiro-Neto (1999, p. 2), que diferenciam a Recuperação da Informação da recuperação de dados da seguinte forma:

Recuperação de dados, no contexto de um Sistema de Recuperação da Informação, consiste basicamente na determinação de quais documentos de uma coleção contêm os termos buscados, o que frequentemente não é o bastante para satisfazer as necessidades de informação do usuário. De fato, o usuário de um Sistema de Recuperação da Informação está mais preocupado em obter informações sobre o assunto do que obter tudo aquilo o que tiver relação direta com seus termos da busca [...]. (tradução livre elaborada pelo autor)

No meio acadêmico, Ferneda (2003, p. 15) relata que a Recuperação da Informação tem por objetivo “representar o conteúdo dos documentos do *corpus* (coleção de documentos de

busca) e apresentá-los ao usuário de uma maneira que lhe permita uma rápida seleção dos itens que satisfazem total ou parcialmente a sua necessidade de informação [...]” (FIGURA 1).

Figura 1 – Representação do processo de Recuperação da Informação



Fonte – Adaptada de Ferneda (2003, p. 15).

Por se tratar de um assunto não esgotado e bem difundido na Ciência da Computação, além de valorizado cada dia mais no mercado de trabalho (SINGHAL, 2001, p. 7-8), SRIs são o objetivo primário neste estudo. Utilizando-se de um referencial bibliográfico consolidado, busca-se aqui apresentar detalhes sobre a implementação e avaliação dos modelos clássicos de Recuperação da Informação, mais especificamente os modelos Booleano, Vetorial e Probabilístico (BÜTTCHER; CLARKE; CORMACK, 2010, p. 51-66).

Neste estudo o *corpus* é composto por 200 artigos científicos em formato *Portable Document Format* (PDF) publicados em periódicos na área de Ciência da Computação e julgados por especialistas convidados segundo alguns critérios que serão apresentados ao longo deste trabalho. De acordo com Silva, Santos e Amorim (2013, p. 28) grande parte da literatura trata da implementação de SRIs no ambiente *web*, onde todo o vasto *corpus* de dados é coletado e indexado de forma automática por agentes de *software* chamados *crawlers* ou rastreadores. Neste estudo, entretanto, os documentos que compõem seu *corpus* estão em um contexto local, fora da *web*, exigindo assim sua indexação, julgamento e classificação prévia.

A motivação para o presente projeto deve-se à implementação de um sistema de informação complementar para o sistema de Gestão de Pessoas do Câmpus Formiga do Instituto Federal de Minas Gerais, denominado “Pipou”. Dentre as funcionalidades consideradas para o sistema Pipou está o gerenciamento de documentos em formato PDF relacionados à progressão funcional de funcionários; publicações de portarias, designações, posse e nomeações de servidores no Diário Oficial; dentre outros, cuja solicitação é frequente e manual, dispendendo tempo para sua recuperação. Para agilizar os processos relacionados a tais solicitações é de suma importância que o sistema possua alguma ferramenta de recuperação de informações capaz de prover essa necessidade informacional de maneira ágil e eficaz. Isto posto, justifica-se o desenvolvimento e experimentação de motores de busca clássicos para recuperação de

documentos digitalizados ou originados digitalmente em formato PDF, objeto resultante deste trabalho.

Após contextualizado o problema, seus conceitos chave e sua justificativa, sumariza-se aqui o objetivo primário deste estudo: analisar, de forma experimental, a relevância dos documentos retornados pelo motor de busca com relação aos termos pré-definidos como expressão de busca para três modelos de Recuperação da Informação clássicos: modelo Booleano, modelo Vetorial e modelo Probabilístico.

Por meio de tal estudo, será possível se inferir qualitativamente qual dos métodos clássicos melhor se adéqua ao *corpus* de um conjunto de documentos PDF, expressando assim resultados obtidos pela aplicação de apenas um *dataset* pequeno e autocontido, não sendo aplicável portanto a todo o universo de *datasets* existente. Os frutos deste estudo serão tomados como referência para eventuais escolhas do autor na implementação de um SRI.

São objetivos secundários, e de forma mais específica os seguintes tópicos:

- Pesquisar o tema Sistemas de Recuperação da Informação, assim como a avaliação de desempenho de tais sistemas;
- Implementar um máquina de buscas em PHP contendo os métodos clássicos de RI para um conjunto de arquivos digitais em formato PDF;
- Analisar qual dos métodos clássicos retorna melhores resultados considerando relevância, agilidade, precisão e cobertura, usando para isso da análise experimental de gráficos de cobertura e precisão média interpolada em 11 pontos, originados pela aplicação do SRI a um determinado *corpus*.

As seções a seguir neste trabalho se subdividem da seguinte forma: o capítulo dois apresenta o referencial teórico sobre SRIs, abrangendo seus conceitos e processos, seus métodos clássicos de Recuperação da Informação, bem como os processos de avaliação de SRIs. No capítulo três são apresentados os materiais e métodos envolvidos no estudo e implementação do protótipo da máquina de busca proposta, que consiste principalmente do processo de construção do índice invertido, dos três modelos clássicos de Recuperação da Informação, além de sua posterior avaliação e análise experimental. O capítulo quatro fornece os resultados e discussão sobre todo o desenvolvimento do estudo e, por fim, os resultados obtidos empiricamente pela análise experimental.

2 SISTEMAS DE RECUPERAÇÃO DA INFORMAÇÃO

2.1 Gestão eletrônica de documentos

A prática de arquivamento da informação pode ser rastreada até por volta de 3.000 anos a.C., onde os sumérios designaram áreas especiais para armazenamento de tábuas de barro com inscrições cuneiformes. Até mesmo nessa época entendia-se que a organização e o acesso facilitado aos arquivos era crucial para um eficiente uso da informação (SINGHAL, 2001, p. 1).

A sociedade vive atualmente a Era Digital, na qual a informação é tratada como um bem valioso. De acordo com Lau (2003), o uso de Sistemas de Recuperação da Informação para a gestão da informação é cada vez mais necessário, visto que a informação e o conhecimento assumem um papel estratégico, alavancando novas possibilidades de crescimento em termos de produtos e serviços em empresas, governos e demais instituições. Em um contexto comercial, verifica-se cada vez mais a informatização de processos e digitalização de seus meios e ferramentas. Dentre elas destacam-se os Gestores Eletrônicos de Documentos (GED), sistemas que visam garantir a organização e acesso em formato digital de todo o tipo de informação gerado em meio físico (GED, 2016), sendo constituído pelas seguintes funcionalidades:

- **Capture:** acelera processos de negócio através da captação de documentos e formulários, transformando-os em informações confiáveis e recuperáveis, passíveis de serem integradas a todas as aplicações de negócios.
- **Document Imaging (DI):** propicia a conversão de documentos do meio físico para o digital.
- **Document Management (DM):** permite gerenciar com mais eficácia a criação, revisão, aprovação e descarte de documentos eletrônicos.
- **Workflow / Business Process Management (BPM):** controle e gerência de processos dentro de uma organização, garantindo que as tarefas sejam executadas pelas pessoas certas no tempo previamente definido.
- **COLD/ERM:** tecnologia que trata páginas de relatórios, incluindo a captura, indexação, armazenamento, gerenciamento e recuperação de dados.

- **Forms Processing:** possibilita reconhecer as informações e relacioná-las com campos em bancos de dados, automatizando o processo de digitação. Podem utilizar técnicas de *Intelligent Character Recognition (ICR)* e *Optical Character Recognition (OCR)* para digitalização e reconhecimento de textos inteiros.
- **Records and Information Management (RIM):** gerencia o ciclo de vida de um documento, independente da mídia em que se encontre.

Apesar de se apresentar como um *software* maduro e de evolução tecnológica observável no contexto comercial de médio/grande porte, a base fundamental de funcionamento de GEDs reside justamente nos mecanismos de recuperação eficiente da informação, campo da Ciência da Computação e da Ciência da Informação que é foco de estudo há várias décadas e que tem hoje sua atividade desempenhada especificamente por Sistemas de Recuperação da Informação (SPRAGUE Jr., 1995; FERNEDA, 2003).

2.2 Recuperação da Informação

Durante a Segunda Guerra Mundial (1939-1945) ocorreu uma explosão informacional identificada por Vannevar Bush (CRUZ, 2011, p. 11-13) como “o irreprímível crescimento exponencial da informação e de seus registros, particularmente em ciência e tecnologia” (SARACEVIC, 1996, p. 42). A necessidade de se obter a informação em um curto espaço de tempo endossou o estudo da Recuperação da Informação como possível solução para o problema originado por tal explosão.

Há muitas décadas os modelos de Recuperação da Informação vêm sendo desenvolvidos, sendo anteriores até mesmo à invenção dos computadores e recursos tecnológicos da era moderna. Ainda que pensadas há tanto tempo, várias das teorias e ideias primárias relacionadas à Recuperação da Informação são até hoje utilizadas como base no desenvolvimento de novos sistemas computacionais, conforme Silva, Santos e Ferneda (2013, p. 29).

O termo Recuperação da Informação (por máquinas) foi criado por Mooers (1951, p. 25) que o definiu formalmente como:

[...] nome do processo onde um possível usuário de informação é capaz de converter a sua necessidade de informação em uma lista real de citações de documentos armazenados que contenham informações úteis a ele. Isto é o processo de busca ou descoberta sob informação armazenada. (tradução livre elaborada pelo autor)

Em contraste com esta definição, Manning (2009, p. 1) cita que o simples fato de se observar o número de um cartão de crédito é caracterizado como Recuperação da Informação. Contudo, ressalta que no contexto acadêmico a Recuperação da Informação denota a busca por material, normalmente documentos de natureza não estruturada que satisfaça a uma necessidade informacional dentro de uma grande coleção de dados, normalmente armazenada em computadores.

Segundo conceitua Mooers (1951, p. 25), na visão do usuário esse processo de Recuperação da Informação parte do pressuposto de uma necessidade informacional do usuário, ou seja, este deve fornecer a um SRI uma consulta baseada em sua necessidade informacional. O objetivo de um SRI consiste em comparar a consulta fornecida pelo usuário com os documentos armazenados na coleção, retornando os que melhor satisfazem a necessidade do usuário (POLTROCK, 2003, p. 246). Contudo, na visão de um SRI, o processo de Recuperação da Informação inicia-se ao receber uma consulta do usuário.

Além do fato de se recuperar a informação de forma a suprir a necessidade informacional do usuário, mesmo que os resultados obtidos não sejam todos aqueles que existem, mas sim os mais relevantes segundo os critérios do SRI, ainda há a questão do tempo de resposta que na perspectiva do usuário é um fator importante usado para discernir a efetividade de um SRI (KOWALSKI, 2011, p. 6).

Em todas as definições de RI supracitadas ressalta-se a subjetividade na questão da relevância da informação apresentada, afinal “informações úteis ao usuário” e “material que satisfaça uma necessidade informacional” são frases inerentemente subjetivas – o que é relevante para um usuário, pode não ser para outro. Neste contexto, Baeza-Yates e Ribeiro-Neto (1999, p. 2) destacam a importância do estudo da recuperação de informações alavancada principalmente pelo advento da rede mundial de computadores – *World Wide Web* (WWW). Ainda segundo os autores, anteriormente à década de 90, apesar da maturidade do referido campo de pesquisa, tal segmento era objeto de estudo basicamente de bibliotecários e especialistas da Ciência da Informação, sem muita aplicação de mercado.

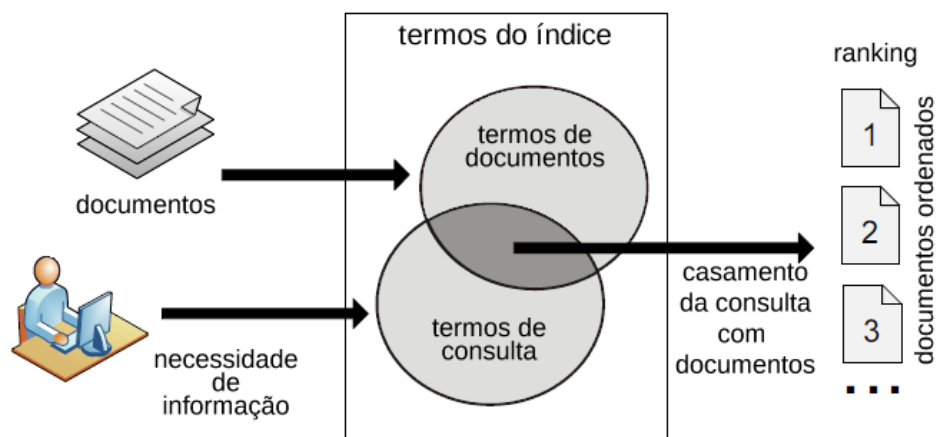
Na Ciência da Computação os SRIs se destacam devido à grande aplicabilidade de conceitos naturalmente inerentes à computação em seu funcionamento. A utilização de bancos de dados (DAS, 2005), engenharia de software (MARCUS *et al.*, 2008), estruturas de dados

para manipulação em memória e métodos de busca (FRAKES; BAEZA-YATES, 1992, p. 18), inteligência artificial (GOKER; DAVIES, 2009, p. 135), álgebra de conjuntos, álgebra booleana e álgebra linear (DOMINICH, 2008), são alguns dos conhecimentos necessários para a implementação de um Sistema de Recuperação da Informação.

2.3 Sistemas de Recuperação da Informação

Baeza-Yates e Ribeiro-Neto (1999, p. 21) caracterizam o processo de Recuperação da Informação composto por dois componentes: um índice¹, responsável por receber como entrada documentos para indexação² e um Sistema de Recuperação da Informação, responsável por receber expressões de busca do usuário. A Recuperação da Informação tem por objetivo encontrar um conjunto ordenado de documentos (etapa conhecida por *matching* e aqui traduzida por casamento) que sejam relevantes para a consulta feita pelo usuário (etapa denominada *ranking* ou classificação/ordenação). A Figura 2 ilustra este processo, onde usuários com necessidades informacionais realizam consultas ao índice, que realiza o casamento entre os termos existentes nos documentos com os termos de consulta. Após determinar o resultado do casamento usando alguma técnica de Recuperação da Informação, os documentos são ordenados por relevância e devolvidos ao usuário do SRI.

Figura 2 – Processo de Recuperação da Informação



Fonte – Adaptada de Baeza-Yates e Ribeiro-Neto (2012, p. 4).

¹ Índice: similar ao índice de um livro, é uma lista ordenada de forma a garantir acesso rápido aos seus termos de indexação, seja por ordenação alfabética, ou numérica (DICIONÁRIO INFORMAL, 2016).

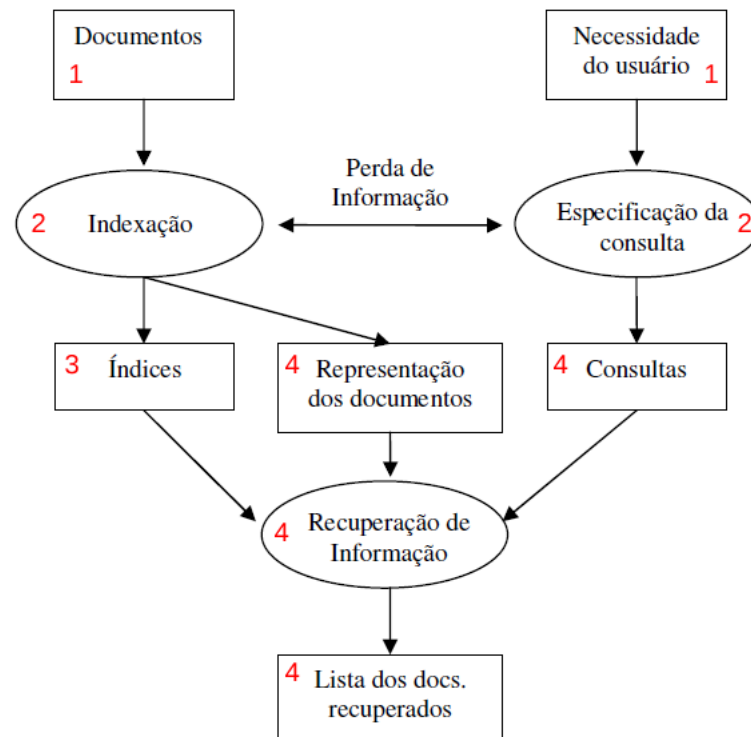
² Indexação: processo de se adicionar termos ao índice, tornando o documento um item recuperável de acordo com seu conteúdo, ou seja, pelos seus termos. (DICIONÁRIO INFORMAL, 2016).

Portanto, um SRI pode ser visto como parte de um Sistema de Informação (SI) com a atribuição de armazenar os documentos em um banco de dados, possibilitando sua posterior recuperação a fim de responder à necessidade informacional do usuário (BÜTTCHER; CLARKE; CORMACK, 2010, p. 5-7).

Segundo Rowley (2002, p. 399) e Baeza-Yates e Ribeiro-Neto (1999, p. 3-8), as principais etapas envolvidas na construção de um SRI são:

1. **AQUISIÇÃO:** seleção e obtenção de forma textual de documentos digitais e expressões de consulta;
2. **INDEXAÇÃO:** aquisição e representação da informação na forma de índice;
3. **ARMAZENAMENTO:** identificação e representação do conteúdo do documento;
4. **RECUPERAÇÃO:** especificação da função de comparação que seleciona os documentos relevantes baseada nas representações (especificação de consulta).

Figura 3 – Componentes de um Sistema de Recuperação da Informação



Fonte – Adaptada de Gey (1992 *apud* SCHREIBER *et al.*, 2008, p. 3).

Cada uma destas etapas possui subprocessos que serão abordados nas seções seguintes apoiadas pelo fluxograma disposto na Figura 3, o qual demonstra que a etapa de aquisição (1) não é unilateral, afinal recebe como entrada tanto documentos quanto expressões de busca

fornecidas pelo usuário. Apesar de ser uma atividade naturalmente prévia à consulta, a indexação (2) também deve ser realizada sob a expressão de consulta, pois para que seja possível a comparação de cada documento do *corpus* com a consulta realizada é necessário que esta consulta seja especificada, ou representada, de forma semelhante aos documentos.

A Figura 3 ainda demonstra que o processo de indexação tipicamente é finalizado pela associação do documento armazenado (e representado no sistema por um identificador numérico, por exemplo) aos seus termos anteriormente adicionados ao índice (3). Logo após estas etapas, fica a cargo do SRI a responsabilidade da representação lógica dos documentos e consultas que, junto a um modelo de Recuperação da Informação, vão garantir a entrega de resultados relevantes à consulta feita pelo usuário (4).

2.4 Fundamentos de Recuperação da Informação

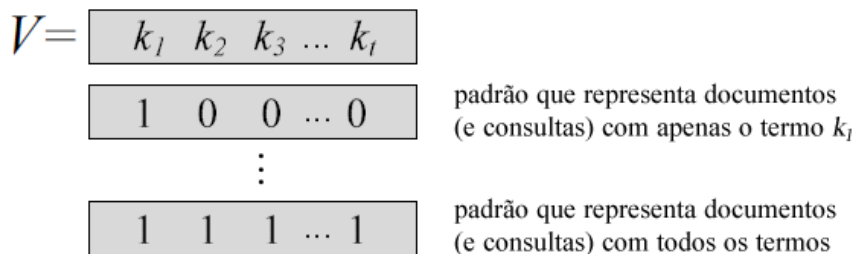
Esta seção apresenta conceitos fundamentais sobre representação de documentos, frequência termo-documento, atribuição de pesos e índice de termos. Tais conceitos são fundamentais à Recuperação da Informação e serão retomados adiante durante a discussão sobre o processo de indexação e sobre os modelos clássicos usados neste trabalho.

2.4.1 Representação lógica de documento

Segundo Baeza-Yates e Ribeiro-Neto (1999, p. 24-25), cada documento é representado dentro de um SRI como um conjunto de palavras-chave representativas, ou seja, termos de um índice. Um termo de indexação é geralmente uma palavra ou um agrupamento de palavras que representa um conceito ou significado presente no documento (FERNEDA, 2003, p. 20). De forma a minimizar custos de performance e armazenamento, um conjunto pré-selecionado de termos de indexação pode ser usado para representar o conteúdo de um documento, porém o mais comum é a representação de todo o conteúdo no índice, chamada de representação em *full text*.

A ideia de representação lógica de um documento ou consulta se baseia na seguinte definição: seja t o número de termos de indexação em uma coleção de documentos e k_i o i -ésimo termo desse índice. O vocabulário $V = \{k_1, k_2, k_3, \dots, k_t\}$ é o conjunto de todos os termos de indexação distintos presentes na coleção (BAEZA-YATES; RIBEIRO-NETO, 1999, p. 25). Cada padrão de ocorrência de termos é chamado de **componente conjuntiva do termo** (FIGURA 4), a qual é associada unicamente a cada documento como $c(d_j)$ ou consulta $c(q)$. Como demonstra a Figura 4, a componente conjuntiva para o documento com apenas o primeiro termo k_1 indexado é $c(d_j) = \{1, 0, 0, 0, \dots, 0\}$, enquanto que para uma consulta com os termos k_2 e k_3 , seria $c(q) = \{0, 1, 1, 0, \dots, 0\}$.

Figura 4 – Representação lógica de documentos e consultas



Fonte – Adaptada de Baeza-Yates e Ribeiro-Neto (2012, p. 12).

Segundo Manning, Raghavan e Schütze (2009, p. 3-4) o resultado dessa representação pode ser compreendido como uma matriz de incidência binária de termo-documento (FIGURA 5), onde os termos são as unidades de indexação representadas pelas linhas da matriz, e os documentos são as colunas da matriz. O cruzamento de um termo com um documento na matriz é usado para representar os valores **contém** (representado na matriz por 1) ou **não contém** (representado na matriz por 0), como podemos ver na Figura 5. Pode-se usar a transposta da matriz, de forma que as linhas sejam formadas por documentos e as colunas por termos.

Figura 5 – Matriz de incidência termo-documento

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
Antony	1	1	0	0	0	1	
Brutus	1	1	0	1	0	0	
Caesar	1	1	0	1	1	1	
Calpurnia	0	1	0	0	0	0	
Cleopatra	1	0	0	0	0	0	
mercy	1	0	1	1	1	1	
worser	1	0	1	1	1	0	
...							

Fonte – Manning, Raghavan e Schütze (2009, p. 4).

Entretanto, Manning, Raghavan e Schütze (2009, p. 4-6) afirmam que esse tipo de representação é inviável em sua análise assintótica. Considerando-se, por exemplo, $N = 1$ milhão de documentos, cada um com uma média de mil palavras. Considerando-se ainda uma média de seis *bytes* por palavra incluindo espaços e pontuação, tudo isso resultaria aproximadamente 6 GB de dados armazenados. Digamos que $M = 500$ mil termos distintos entre estes, o resultado de $(500 \text{ mil}) \times (1 \text{ milhão})$ é aproximadamente 500 bilhões de 0's e 1's, cujo consumo de memória pode ser reduzido pela utilização de matrizes esparsas.

2.4.2 Matriz de frequência termo-documento

A ocorrência de um termo k_i em um documento d_j estabelece uma relação entre k_i e d_j . Essa relação pode ser quantificada pela frequência do termo presente no documento, fato este representado pela **matriz de frequência termo-documento** ilustrada na Figura 6. Nesta representação cada elemento $f_{i,j}$ representa a frequência do termo k_i no documento d_j , também chamado de *term frequency* ou simplesmente *tf* (BAEZA-YATES; RIBEIRO-NETO, 1999).

Figura 6 – Matriz de frequência termo-documento

$$\begin{array}{cc}
 & d_1 & d_2 \\
 k_1 & \left[\begin{array}{cc} f_{1,1} & f_{1,2} \end{array} \right. \\
 k_2 & \left[\begin{array}{cc} f_{2,1} & f_{2,2} \end{array} \right. \\
 k_3 & \left[\begin{array}{cc} f_{3,1} & f_{3,2} \end{array} \right.
 \end{array}$$

Fonte – Baeza-Yates e Ribeiro-Neto (2012, p. 13).

Embora padeça do mesmo problema apontado nas matrizes de incidência, as matrizes de frequência são usadas em pequenas instâncias do problema em modelos clássicos, como será visto na seção 2.7 deste documento.

2.4.3 Atribuição de peso aos termos

Para a correta classificação de resultados em um SRI, pesos numéricos são atribuídos a documentos e consultas. Para Greengrass (2000, p. 13), um peso $w_{ij} > 0$ é atribuído a um dado termo k_i em um determinado documento d_j , sendo este valor diferente para cada documento distinto em que o termo aparece. Isto se torna uma medida de efetividade ou importância da presença de um termo em cada documento da coleção.

Termos de um documento não são igualmente úteis para descrever seu conteúdo, afinal existem termos de indexação que são mais vagos que outros. Para exemplificar, seja uma palavra que aparece em todos os documentos da coleção. Ela é completamente inútil para as atividades de classificação do SRI, pois sua relevância é ínfima, visto que toda a coleção a possui (BAEZA-YATES; RIBEIRO-NETO, 1999, p. 24-25).

De acordo com Baeza-Yates e Ribeiro-Neto (2012, p. 26-27), pesos podem ser calculados usando-se a frequência de ocorrência de um termo nos documentos. O valor total da frequência de ocorrência (F_i) do termo k_i na coleção é definido por:

$$F_i = \sum_{j=1}^N f_{i,j} \quad (1)$$

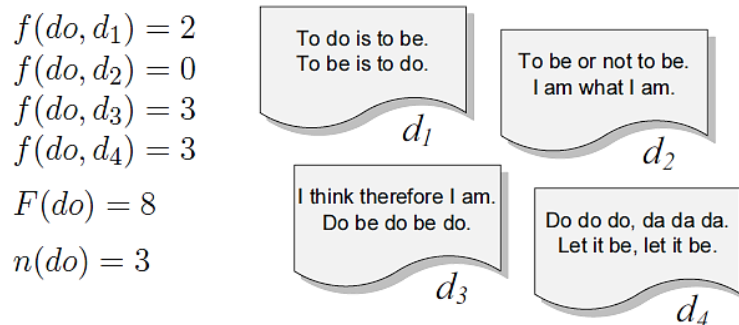
onde

- F_i : frequência total de ocorrência de um dado termo k_i em toda a coleção.
- N : o número de documentos na coleção.
- $f_{i,j}$: frequência de ocorrência do termo de indexação k_i no documento d_j .

A frequência de documento n_i (também denominada df) de um termo k_i é o número de documentos nos quais este termo aparece, logo $n_i \leq F_i$.

A Figura 7 exemplifica a obtenção da frequência do termo tf para o termo “do” contido em quatro documentos de exemplo d_1 , d_2 , d_3 e d_4 , bem como a frequência total $F(do)$ e a frequência do documento $df(do)$ (neste caso, usando a notação $n(do)$) do mesmo termo nesta coleção.

Figura 7 – Atribuição de peso ao termo "do"



Fonte – Baeza-Yates e Ribeiro-Neto (2012, p. 27).

Conforme Hiemstra (2009, p. 9 *apud* SALTON, 1971), Salton e Yang (1973), experiências demonstram que a atribuição de pesos a termos não é uma tarefa trivial. Os autores sugeriram o uso da técnica chamada de **tf-idf**, que é a combinação de *term frequency* **tf**, ou seja, o número de ocorrências de um termo em um documento, e *inverse document frequency* **idf**, que é o valor inversamente proporcional ao *document frequency* **df**. Detalhes sobre o funcionamento desta técnica podem ser consultados diretamente na fonte e têm a explicação de seu cálculo a seguir.

A Figura 8 apresenta uma coleção de exemplo constituída por quatro documentos d_1 , d_2 , d_3 e d_4 , junto ao vocabulário do índice composto por 14 termos de indexação e aos cálculos de **tf**, **idf** e **tf-idf**, que serão abordados em detalhes até o final desta seção. Nesta Figura 8 podemos ver o cálculo do $tf_{i,j}$, onde i é o índice do termo de indexação k_i e j o índice do documento d_j . O n_i representa a frequência **df** de documentos com o termo k_i , usado no cálculo do **idf**. Por fim,

temos os resultados do cálculo do *tf-idf* para cada termo em cada documento, conforme a Equação 4, logo a seguir nesta seção.

Figura 8 – Atribuição de pesos por *tf*, *idf* e *tf-idf*

	Vocabulary		<i>tf</i>				<i>idf</i>		<i>tf-idf</i>			
			$tf_{i,1}$	$tf_{i,2}$	$tf_{i,3}$	$tf_{i,4}$	n_i	idf_i	d_1	d_2	d_3	d_4
To do is to be. To be is to do. <i>d₁</i>	1	to	3	2	-	-	2	1	3	2	-	-
	2	do	2	-	2.585	2.585	3	0.415	0.830	-	1.073	1.073
	3	is	2	-	-	-	1	2	4	-	-	-
To be or not to be. I am what I am. <i>d₂</i>	4	be	2	2	2	2	4	0	-	-	-	-
	5	or	-	1	-	-	1	2	-	2	-	-
	6	not	-	1	-	-	1	2	-	2	-	-
I think therefore I am. Do be do be do. <i>d₃</i>	7	I	-	2	2	-	2	1	-	2	2	-
	8	am	-	2	1	-	2	1	-	2	1	-
	9	what	-	1	-	-	1	2	-	2	-	-
Do do do, da da da. Let it be, let it be. <i>d₄</i>	10	think	-	-	1	-	1	2	-	-	2	-
	11	therefore	-	-	1	-	1	2	-	-	2	-
	12	da	-	-	-	2.585	1	2	-	-	-	5.170
			13	let	-	-	-	2	1	2	-	4
			14	it	-	-	-	2	1	2	-	4

Fonte – Adaptada de Baeza-Yates e Ribeiro-Neto (2012, p. 35-43).

Pela suposição de Luhn (KOCABAS; DINÇER; KARAOGLAN, 2011, p. 993-994), o valor do peso atribuído a um termo em um documento (w_{ij}) é proporcional ao f_{ij} (*tf*), ou seja, quanto mais frequente é um termo no texto de um documento, maior é seu peso e consequentemente sua importância ao representar um documento. Isto leva à fórmula apresentada na Equação 2 e exemplo de seu uso pela Figura 8 (BAEZA-YATES; RIBEIRO-NETO, 2012, p. 34).

$$tf_{ij} = \begin{cases} 1 + \lg f_{ij} & \text{se } f_{ij} > 0 \\ 0 & \text{caso contrário} \end{cases} \quad (2)$$

onde

- tf_{ij} : frequência do termo k_i para o documento d_j .
- $\lg f_{ij}$: logaritmo na base 2 da frequência do termo k_i para o documento d_j . O log é um método de normalização para uso junto ao *idf* (BAEZA-YATES; RIBEIRO-NETO, 2012, p. 44).

A operação de logaritmo na base 2 é a forma de normalização mais abordada para o cálculo de *tf* na literatura, por fazer uma comparação diretamente relativa aos pesos por *idf*. Pelo uso do *tf*, se muitos termos de indexação estiverem atribuídos a um documento, ele será

recuperado por consultas para as quais nem mesmo é relevante, considerando-o assim como uma estatística local de cada documento, variando seu valor de um documento para outro. Por outro lado, o *idf* é considerado como uma estatística global, medindo o quão distribuído está um termo em toda a coleção, o que determina o quão provável é a aparição de um dado termo em um documento (GREENGRASS, 2000, p. 19).

A fórmula apresentada pela Equação 3 para o cálculo do *idf* é o fundamento da atribuição de pesos moderna e é usada para *ranking* em praticamente todos os modelos atuais de Recuperação da Informação (BAEZA-YATES; RIBEIRO-NETO, 1999, p. 29):

$$idf_i = \log \frac{N}{n_i} \quad (3)$$

onde

- *idf_i* : valor da frequência do documento inversa (do inglês: *inverse document frequency*).
- *N* : o número de documentos na coleção.
- *n_i* : *document frequency df*. Número de documentos da coleção que possuem um determinado termo.

Termos estão distribuídos em um texto de acordo com a lei de Zipf, que diz empiricamente a dimensão, importância ou frequência de elementos em uma lista ordenada (PIANTADOSI, 2014, p. 1).

Segundo Baeza-Yates e Ribeiro-Neto (1999, p. 29), o melhor esquema de atribuição de pesos conhecido é o *tf-idf*, caracterizado pela mescla entre os dois esquemas apresentados até então, *tf* e *idf*. Sua fórmula é apresentada pela Equação 4.

$$w_{i,j} = \begin{cases} (1 + \lg f_{i,j}) \times \log \frac{N}{n_i} & \text{se } f_{i,j} > 0 \\ 0 & \text{caso contrário} \end{cases} \quad (4)$$

onde

- *w_{i,j}* : peso numérico atribuído a um dado termo *k_i* em um documento *d_j* por *tf-idf*.
- $(1 + \lg f_{i,j})$: cálculo normalizado do *term frequency tf* (ver Equação 2).
- $\log \frac{N}{n_i}$: cálculo normalizado do *inverse document frequency idf* (ver Equação 3).

Logo, a Equação 4 apresenta o cálculo do peso atribuído a cada termo k_i em um documento d_j , que nada mais é que a multiplicação entre os valores de tf e idf , apresentados anteriormente.

Existem variantes dos esquemas de atribuição apresentados, tanto para tf , quanto idf e $tf-idf$. Contudo, Baeza-Yates e Ribeiro-Neto (2012, p. 46) recomendam o uso de três esquemas $tf-idf$, conforme podemos ver na Tabela 1.

Tabela 1 – Esquemas recomendados de atribuição de pesos a termos por $tf-idf$

Esquema de atribuição de pesos	Atribuição de pesos aos termos do documento	Atribuição de pesos aos termos da consulta
1	$f_{i,j} \times \log \frac{N}{n_i}$	$\left(0.5 + 0.5 \times \frac{f_{i,q}}{\max_i f_{i,q}}\right) \times \log \frac{N}{n_i}$
2	$1 + \log f_{i,j}$	$\log \left(1 + \frac{N}{n_i}\right)$
3	$(1 + \log f_{i,j}) \times \log \frac{N}{n_i}$	$(1 + \log f_{i,q}) \times \log \frac{N}{n_i}$

Fonte – Adaptada de Baeza-Yates e Ribeiro-Neto (2012, p. 46).

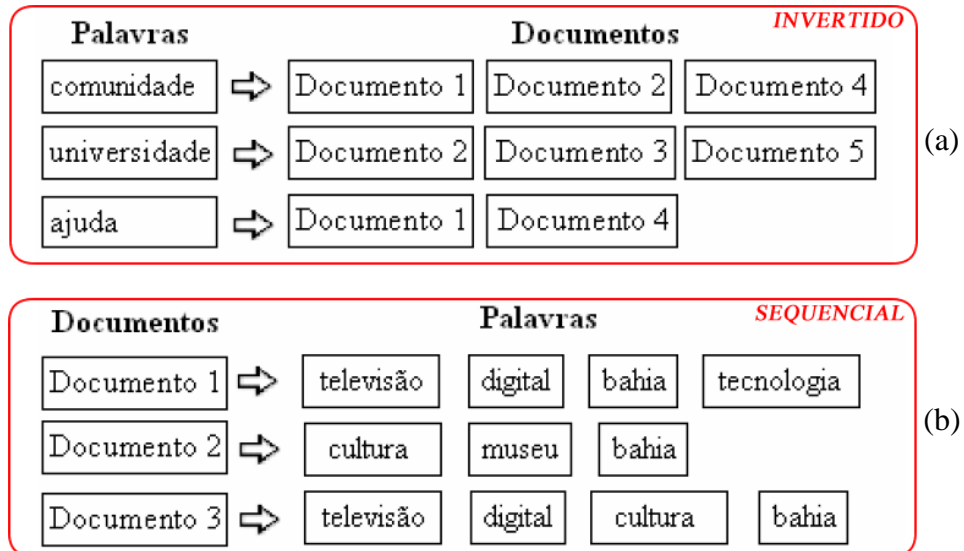
Na Tabela 1 verificam-se normalizações do valor por diferentes meios, por exemplo: no primeiro esquema, verifica-se que para a atribuição de peso ao termo em um determinado documento utiliza-se a multiplicação do tf pelo seu valor de idf . Na atribuição do peso ao termo da consulta, existe uma normalização diferente, somando-se 0,5 à multiplicação de 0,5 pela razão entre o tf e o tf do último termo da consulta ($\max_i f_{i,q}$). Nos outros esquemas são apresentadas outras sugestões de normalização pelos autores, o que deve ter seus resultados analisados de forma empírica pelo desenvolvedor de um SRI.

2.4.4 Índice de termos

Segundo Frakes e Baeza-Yates (1992, p. 38), três das estruturas mais comumente utilizadas na Recuperação da Informação são os índices lexicográficos, estruturas de arquivos por agrupamento, denominadas *clusters* (CIFERRI, 2013, p. 30) e índices baseados em *hashing* (SILBERSCHATZ; KORTH; SUDARSHAN, 2001, p. 446-490). Na categoria de índices

lexicográficos, as implementações mais comuns encontradas são de índices invertidos e índices sequenciais, sendo que Amazonas *et al.* (2008, p. 201) os diferencia de acordo com a Figura 9.

Figura 9 – Tipos comuns de índice



Fonte – Adaptada de Amazonas *et al.* (2008, p. 201).

O índice invertido apresentado pela Figura 9.a consiste no mapeamento de cada termo de indexação para uma lista de documentos aos quais ele pertence, contendo ainda dados sobre características dos termos na coleção de documentos, tais como a frequência de cada termo em um documento, como visto, denominado *term frequency* (CARDOSO, 2000, p. 2). O índice disposto na Figura 9.b apresenta um exemplo de índice sequencial, onde o mapeamento é feito de cada documento para uma lista de seus termos indexados.

De acordo com Frakes e Baeza-Yates (1992, p. 40-41) para a implementação de um índice podem ser usadas estruturas de vetores ordenados (*sorted arrays*), árvores B (*B-trees*) (CORMEN, 2002, p. 349) e até mesmo árvores digitais, denominadas em inglês como *tries* (SEDGEWICK, R.; WAYNE, K., 2011, p. 730), que utilizam a decomposição digital do conjunto de palavras para representar os termos, por isso também conhecidas como árvores de prefixos. Para Cormen (2002 *apud* AMAZONAS *et al.*, 2008), entretanto, a utilização de índices dispostos como árvores binárias aumenta a necessidade de alocação de espaço em memória para o armazenamento, afinal as atividades de criação e atualização podem gerar um aumento de tempo considerável no processo de indexação.

2.5 Processo de indexação

Segundo Zobel e Moffat (2006, p. 3) diversas fontes podem ser usadas em um processo de indexação, desde arquivos de texto até arquivos de vídeo, áudio e imagem. Apesar de ser necessário o registro de diferentes dados na indexação dos tipos não textuais, o processo de indexação segue basicamente o mesmo princípio.

Segundo Amazonas *et al.* (2008, p. 200) o processo de indexação de dados incorpora diversos conceitos multidisciplinares, como: Linguística, Matemática, Psicologia Cognitiva, Ciência da Computação e, principalmente, as técnicas de recuperação e interpretação de informações. Ainda segundo os autores, a construção de um índice parte basicamente de três etapas: (i) *tokenize*, onde é realizada a análise léxica do texto; (ii) *analysis* em que é realizada a retirada das palavras sem significado do texto em linguagem natural; e (iii) *stemming* onde é feita a radicalização dos termos, no final do processo.

Conforme demonstram Manning, Raghavan e Schütze (2009, p. 6-7), os passos iniciais de processamento do documento seguem a seguinte ordem, exemplificados pelo diagrama da Figura 10, cujas anotações numéricas devem ser observadas como orientação aos passos:

1. **Coletar os documentos e/ou trechos de conteúdo que serão indexados**, que depende do modelo de SRI adotado: *full text* ou por amostragem. Amazonas *et al.* (2008, p. 200) cita que nesta etapa podem ser retirados acentos, caracteres especiais e substituídas letras maiúsculas por minúsculas, possibilitando uma pesquisa mais abrangente nos documentos;
2. **Transformar o texto em *tokens*³, tornando cada documento uma lista de *tokens***. Posteriormente a esta etapa pode-se remover palavras indesejadas, conhecidas na literatura de Recuperação da Informação por *stop-words* ou *blacklist*;
3. **Realizar processamento linguístico**, produzindo uma lista de *tokens* normalizados, que para o índice serão os termos⁴ de indexação. Neste passo, é opcional a radicalização/normalização de palavras, tornando assim o índice menos denso. Sua

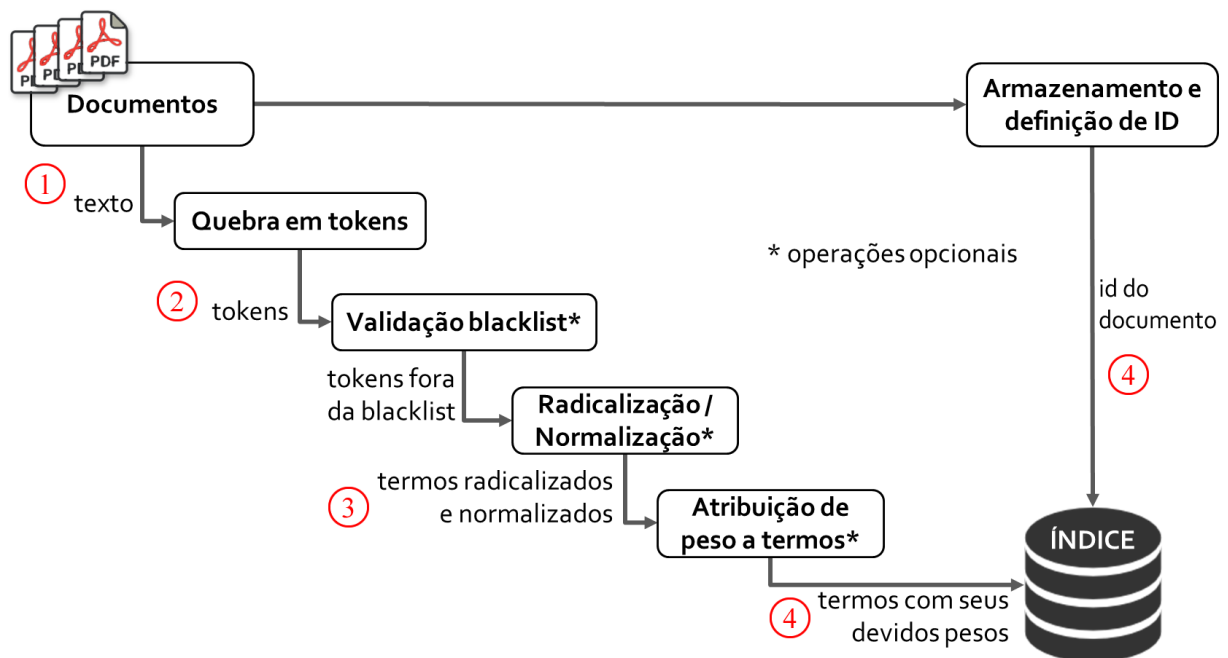
³ *token*: uma unidade de um texto quebrado segundo algum critério (MAUSAM, 2012, p. 5).

⁴ termo de indexação: ou simplesmente termo, é a chave de indexação de um índice, ou seja, é o *token* depois de radicalizado, normalizado e manipulado para inserção no índice (MAUSAM, 2012, p. 5).

radicalização resulta na palavra-base, sem prefixos, sufixos, plural ou flexões verbais, dependendo da formalidades da língua (PENG, 2007; GOSPODNETIC, 2005);

4. **Indexar os documentos com relação à aparição de seus termos**, criando assim um índice invertido (ver Figura 9), que consiste em um dicionário de termos e suas ocorrências. A atribuição de pesos aos termos no índice é realizada pelo armazenamento do *tf* junto à relação termo-documento, sendo que os pesos atribuídos a cada termo participante de uma expressão de consulta são dinâmicos e, portanto, serão abordados na seção 2.7 deste estudo: Modelos de Recuperação da Informação.

Figura 10 – Processo de indexação



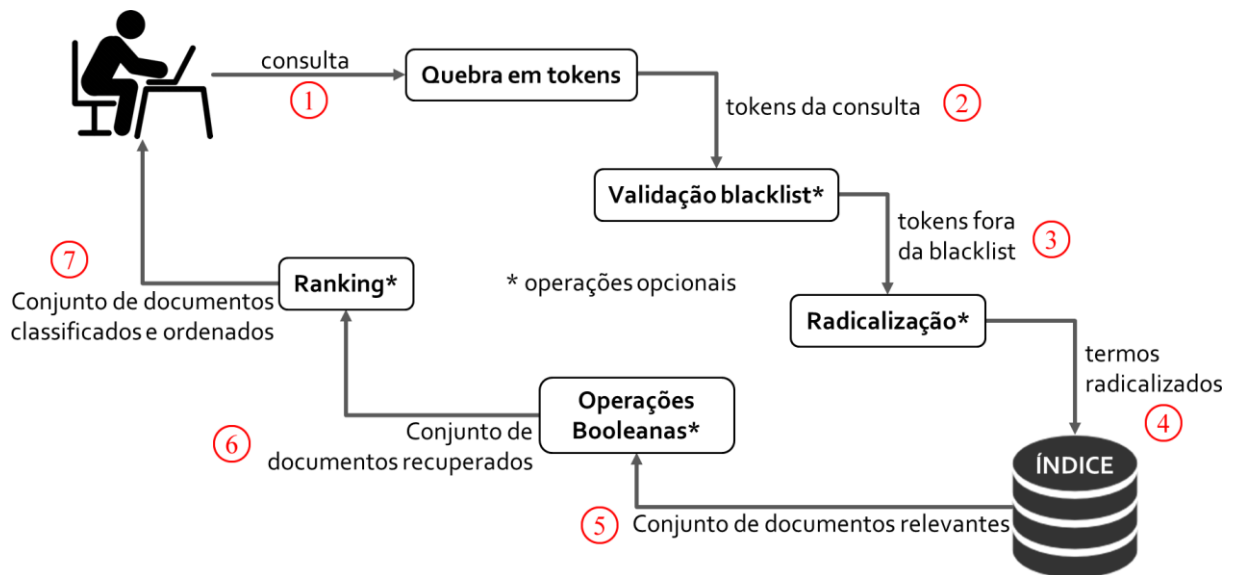
Fonte – Adaptada de Mausam (2012, p. 3).

2.6 Processo de especificação de consulta

Segundo Cardoso (2000, p. 2), o processo de especificação da consulta geralmente é uma tarefa difícil. Frequentemente há uma distância semântica entre a real necessidade do usuário e o que ele expressa na consulta formulada. Essa distância é gerada pelo limitado conhecimento do usuário sobre o universo de pesquisa e pelo formalismo da linguagem consultada.

A especificação de consulta é a parte interativa entre o usuário e um Sistema de Recuperação da Informação. É a etapa pela qual o SRI interpreta a necessidade de informação do usuário e a contrasta com seu *corpus* indexado, resultando uma lista ordenada de documentos classificados segundo alguma função de *ranking*, como se observa pela Figura 11 (BAEZA-YATES; RIBEIRO-NETO, p. 1999. 9-10).

Figura 11 – Processo de especificação de consulta



Fonte – Adaptada de Mausam (2012, p. 4).

A Figura 11 demonstra o processo inverso e posterior ao processo de indexação abordado na seção 2.5. A especificação da consulta passa pelos mesmos passos abordados no processo de indexação. Após as quatro primeiras etapas abordadas no processo de indexação, o próximo passo consiste em três etapas ainda não abordadas neste estudo, numeradas de acordo com o disposto na Figura 11, itens 5, 6 e 7:

- 1. Coletar a expressão de busca do usuário:** etapa semelhante à etapa 1 do processo de indexação, porém aplicada à expressão de busca.
- 2. Transformar o texto da expressão de busca em *tokens*:** assim como a etapa 2 do processo de indexação, também transforma a expressão de busca em uma lista de *tokens*.
- 3. Validação:** passo opcional e similar à verificação de *blacklist* abordada no processo de indexação.
- 4. Radicalização:** passo opcional e similar à etapa de radicalização e obtenção de termos abordada no processo de indexação.

5. **Recuperação de documentos relevantes:** esta etapa consiste na seleção de documentos que têm o casamento exato com o termo buscado (*matching*), respeitando-se as regras da expressão de busca, como é o caso dos operadores lógicos no modelo booleano, pode acontecer de forma filtrada ou não. Filtros podem ser aplicados de acordo com o usuário que requer a informação e dependem da regra de negócio aplicada ao SRI (BAEZA-YATES; RIBEIRO-NETO, 2012, p. 8-9). Para este estudo optou-se pela recuperação sem filtros.
6. **Função de classificação:** etapa principal do processo de Recuperação da Informação, onde são aplicadas as funções de *ranking* ou ordenação do modelo implementado no SRI (ver seção 2.7).
7. **Entrega de resultados:** apresentação dos resultados ao usuário na forma ordenada decrescente por relevância da expressão de busca ao documento (BAEZA-YATES; RIBEIRO-NETO, 1999, p. 10).

2.7 Modelos de Recuperação da Informação

De acordo com Ferneda (2003, p. 29), a eficiência de um SRI está diretamente ligada ao modelo de Recuperação da Informação que o mesmo utiliza, o que influencia diretamente no modo de operação do sistema. Mesmo que os modelos clássicos tenham sido criados entre os anos 60 e 70 e aperfeiçoados nos anos 80, suas principais ideias ainda estão presentes na maioria dos sistemas de recuperação e nos mecanismos de busca *web* atuais.

Para Baeza-Yates e Ribeiro-Neto (1999, p. 23), um modelo de Recuperação da Informação (FIGURA 12) é definido formalmente pela quádrupla

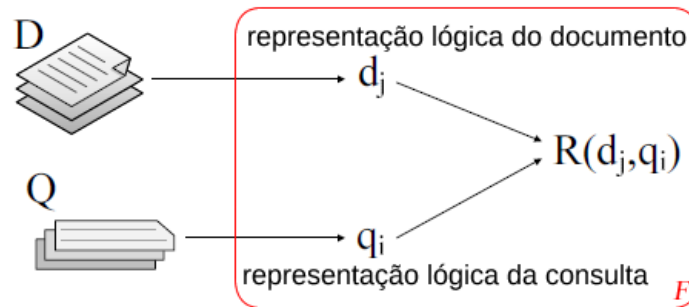
$$[D, Q, F, R(q_i, d_j)] \quad (5)$$

onde

- D : conjunto de visão lógica para os documentos na coleção;
- Q : conjunto de visão lógica para consultas do usuários;

- F : *framework*⁵ de modelagem para documentos e consultas;
- $R(q_i, d_j)$: função de classificação (*ranking*) ou ordenação.

Figura 12 – Definição formal de um modelo de Recuperação da Informação



Fonte – Adaptada de Baeza-Yates e Ribeiro-Neto (2012, p. 6).

Ao desenvolver um modelo de Recuperação da Informação, as formas de representação dos documentos e das necessidades de informação do usuário são as primeiras entidades a serem definidas. Com base nestas entidades, o *framework* do modelo pode ser definido, fornecendo os princípios para a construção da função de classificação (do inglês: *ranking*) (BARTH, 2013, p. 250).

Elencados por Amazonas *et al.* (2008, p. 198), os modelos clássicos utilizados no processo de RI, a saber: Booleano, Vetorial e Probabilístico, apresentam estratégias de busca de documentos relevantes para uma consulta. Segundo Baeza-Yates, Ribeiro-Neto (1999, p. 24-34) e Manning, Raghavan e Schütze (2009), no modelo Booleano, o *framework* é composto por conjuntos de documentos e operações clássicas da teoria de conjuntos. No modelo Vetorial, documentos e consultas são representados como vetores em um espaço n -dimensional, sendo assim o *framework* composto por um espaço n -dimensional e operações de geometria analítica sob vetores. No modelo Probabilístico, o *framework* é definido por operações da teoria das probabilidades. Além destes, modelos muito mais avançados de RI têm sido propostos ao longo dos anos, destacando-se modelos baseados em bases de conhecimento, lógica difusa e redes neurais (CARDOSO, 2000, p. 2-3).

Segundo exemplifica Mausam (2012, p. 6), os modelos são categorizados em grande parte pela sua função de *ranking*, como demonstra o Quadro 1.

⁵ *framework*: arcabouço ou conjunto de ferramentas para resolver problemas de domínio específico (DICIONÁRIO INFORMAL, 2016).

Quadro 1 – Função de *ranking* por modelo de RI

MODELO	FUNÇÃO DE RANKING
Booleano	por casamento exato de consulta
Vetorial	por similaridade à consulta
Probabilístico	por similaridade à consulta
<i>PageRank</i>	por importância dos documentos (<i>hits</i>)
Métodos combinados (híbridos)	função utilizada pela maioria dos mecanismos de busca na <i>web</i> como Bing, Google, Yahoo, dentre outros.

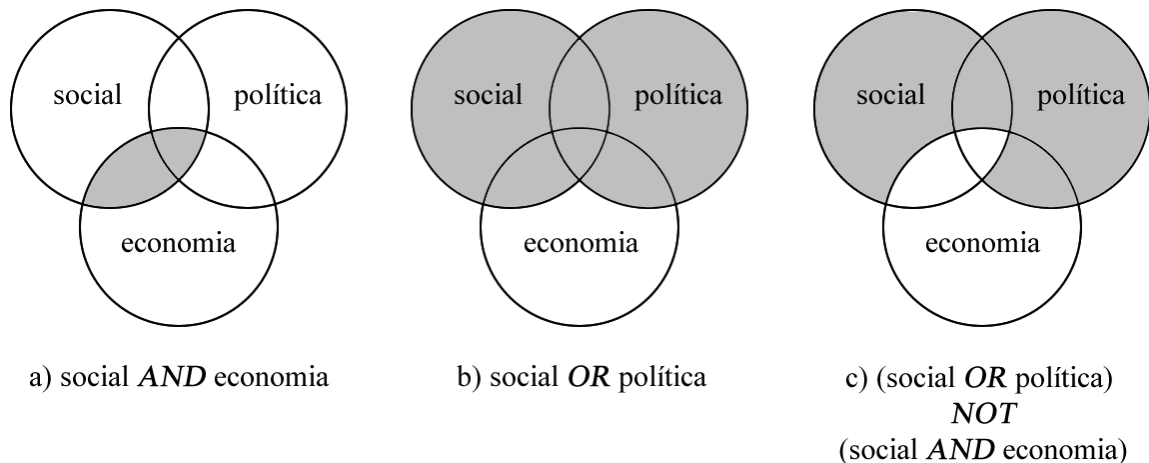
Fonte – Adaptado de Mausam (2012, p. 6).

2.7.1 Modelo Booleano

Considerado como o primeiro modelo de Recuperação da Informação, o modelo Booleano é baseado na teoria dos conjuntos e na álgebra booleana. Nele não há a atribuição de pesos aos termos de indexação. Apenas dois valores são atribuídos, 0 (termo não está presente) e 1 (termo está presente) para cada documento da coleção de busca ou *corpus* (HIEMSTRA, 2009, p. 4; GREENGRASS, 2000, p. 14). Assim, com o intuito de aperfeiçoar a busca, é possível que sejam utilizados os operadores **OR**, **AND** e **NOT**, que correspondem respectivamente aos operadores lógicos de disjunção, conjunção e negação da lógica proposicional, equivalentes à teoria de conjuntos, sendo o último deles um operador unário de complemento de um conjunto. Existe ainda a possibilidade da atribuição de prioridades às operações, fazendo para tanto o uso de parêntesis para correlacionar os termos da pesquisa. Para Amazonas *et al.* (2008, p. 198) neste modelo não há a possibilidade de se medir o grau de relevância dos documentos retornados uma vez que não existe um peso atribuído capaz de orientar a construção de um *ranking*.

Como exemplo, Hiemstra (2009, p. 4) sugere a consulta **social AND economia** que produz como resultado um conjunto de documentos que tem indexado os termos “social” e “economia” em um mesmo documento, ou seja, a interseção entre os dois conjuntos, como demonstrado pela Figura 13.a. Outros exemplos também sugeridos pelo autor são a consulta **social OR política** com resultado ilustrado na Figura 13.b, e a consulta (**social OR política**) **NOT (social AND economia)**, cujo resultado da busca é destacado em cinza na Figura 13.c.

Figura 13 – Combinações booleanas de conjuntos visualizadas como diagramas de Venn⁶



Fonte – Adaptada de Hiemstra (2009, p. 5).

Por se tratar de um modelo simples baseado em um critério de decisão binário, Baeza-Yates e Ribeiro-Neto (1999, p. 26) afirmam que a ausência da informação de relevância prejudica a performance de recuperação, tornando este modelo mais um mecanismo de recuperação de dados do que informação. Entretanto, existem modelos refinados e estendidos do modelo Booleano que garantem um melhor desempenho e até mesmo a relevância, como é o caso por exemplo do uso de lógica difusa (GREENGRASS, 2000, p. 13; KOWALSKI, 2011, p. 18-19; GROSSMAN; FRIEDER, 1998, p. 58-60).

Há ainda neste modelo a vantagem apontada por Hiemstra (2009, p. 4) que é uma sensação de controle do sistema por um usuário avançado que tenha entendimento pleno do uso da ferramenta, tornando imediatamente claro o porquê de determinados documentos estarem presentes como resultado de uma consulta. Apesar de ser um modelo básico, Dominich (2008, p. 23) ressalta que o modelo Booleano é muito importante, afinal é usado extensivamente na aplicação de Sistemas Gerenciadores de Bancos de Dados (SGBD), além de mecanismos de busca por toda a *web*.

O Quadro 2 apresenta de forma resumida a definição das representações lógicas de documentos, de consultas e da função de classificação no modelo Booleano, bem como traz exemplos das mesmas. Seja d_1 um documento que contém os termos k_1 , k_2 e k_4 , obtidos de um *corpus* que contém k_1 , k_2 , k_3 , k_4 , k_5 e k_6 . Sejam as consultas $(k_1 \text{ AND } k_2)$, $(k_3 \text{ OR } k_4)$ e $(k_1 \text{ NOT } k_3)$, apresentadas como exemplo de construção de expressões de consulta e, por fim, o exemplo

⁶ Diagramas de Venn: introduzidos por John Venn em 1881, são a ilustração padrão baseada na teoria dos conjuntos para ilustrar relações matemáticas e lógicas entre diferentes grupos de conjuntos (VENN, 1880 *apud* PIROOZNI; NAGARAJAN; DENG, 2007).

de função de busca (*matching*) (k_1 AND k_2 AND k_3), o qual retorna os documentos do *corpus* que possuem os três termos em seu conteúdo, k_1 , k_2 e k_3 (ver Quadro 2).

Quadro 2 – Representações no modelo Booleano

	REPRESENTAÇÃO	EXEMPLO
Documento	conjunto de termos de indexação representado como vetor de pesos binário	$d_1 = \{1, 1, 0, 1, 0, 0\}$ d_1 contém os termos k_1 , k_2 e k_4
Consulta	termos de indexação (na forma binária) conectados por operadores booleanos	k_1 AND k_2 k_3 OR k_4 k_1 NOT k_3
Função de busca	o documento é considerado relevante se e somente se atender à consulta booleana	k_1 AND k_2 AND k_3 resulta os documentos que têm os três termos consultados (k_1 , k_2 e k_3)

Fonte – Adaptado de Baeza-Yates e Ribeiro-Neto (2012), Ferneda (2003) e Amazonas *et al.* (2008).

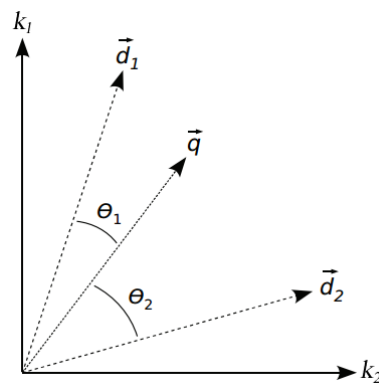
2.7.2 Modelo Vetorial

O modelo Vetorial, também chamado de modelo de espaço vetorial, oferece um ambiente no qual é possível se obter documentos que respondam parcialmente a uma expressão de busca. Por meio da atribuição de pesos aos termos de indexação e termos de consulta, é calculado o grau de similaridade entre a expressão buscada e cada um dos documentos, o que resulta uma lista ordenada decrescente de documentos segundo o grau de similaridade (FERNEDA, 2003, p. 27-28).

Baeza-Yates e Ribeiro-Neto (1999, p. 27-30) formalizam que neste modelo o peso w_{ij} atribuído ao par (k_i, d_j) , termo k_i e documento d_j , é positivo e não binário. Os termos de uma consulta também são associados a um peso. Assim, o vetor de uma consulta é definido no espaço n -dimensional como $\vec{q} = (w_{1,q}, w_{2,q}, \dots, w_{t,q})$, onde $w_{i,q} \geq 0$ e t é o número total de termos no sistema. No caso dos documentos, um vetor no espaço vetorial é definido como $\vec{d}_j = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$. Observa-se que cada termo da expressão de consulta se torna um eixo do referido espaço n -dimensional.

O modelo vetorial propõe o cálculo do grau de similaridade entre o documento \vec{d}_j e a consulta \vec{q} usando a correlação entre os vetores formados por suas componentes. Esta correlação pode ser quantificada utilizando-se a distância euclidiana (QIAN *et al.*, 2004, p. 1232) ou o cosseno do ângulo formado entre vetor consulta \vec{q} e o vetor documento \vec{d}_j , chamada de medida de similaridade por cosseno.

Figura 14 – Medida de similaridade por cosseno no modelo Vetorial



Fonte – Adaptado de Büttcher, Clarke e Cormack (2010, p. 55).

O exemplo apresentado na Figura 14 demonstra o cálculo dos ângulos entre o vetor consulta \vec{q} e os dois vetores documento \vec{d}_1 e \vec{d}_2 . Por razão de $\theta_1 < \theta_2$, \vec{d}_1 será classificado com uma posição de *ranking* melhor que \vec{d}_2 . Outra observação está no fato que neste exemplo apresentado pela Figura 14 a expressão de busca é constituída por dois termos, formando assim um espaço bidimensional.

Visto que o cálculo do ângulo entre dois vetores é realizado pela razão entre o produto interno destes vetores e o produto de suas normas (BOULOS; OLIVEIRA, 1987, p. 208), temos:

$$\cos \theta = \frac{|\vec{q} \cdot \vec{d}|}{\|\vec{q}\| \times \|\vec{d}\|} \quad (6)$$

onde

- $\cos \theta$: cosseno do ângulo formado entre o vetor consulta \vec{q} e o vetor documento \vec{d} .
- $|\vec{q} \cdot \vec{d}|$: módulo do produto interno entre o vetor consulta \vec{q} e o vetor documento \vec{d} , o qual resulta um número real (STEINBRUCH; WINTERLE, 1987, p. 39).

- $\|\vec{q}\| \times \|\vec{d}\|$: produto da norma⁷ do vetor consulta \vec{q} pela norma do vetor documento \vec{d} , que resulta um número real (STEINBRUCH; WINTERLE, 1987, p. 40).

De acordo com Baeza-Yates e Ribeiro-Neto (2012, p. 57), a forma de se calcular a similaridade entre cada documento e a consulta realizada, portanto, equivale a

$$\cos \theta = \mathit{sim}(\vec{d}_j, \vec{q}) = \frac{\sum_{i=1}^t w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^t w_{i,j}^2} \times \sqrt{\sum_{j=1}^t w_{i,q}^2}} \quad (7)$$

onde

- $\mathit{sim}(\vec{d}_j, \vec{q})$: similaridade entre o vetor consulta \vec{q} e o documento \vec{d}_j .
- $w_{i,j}$: peso do i -ésimo elemento do vetor documento \vec{d}_j .
- $w_{i,q}$: peso do i -ésimo elemento do vetor consulta \vec{q} .

Logo, sabendo-se que $w_{i,j} > 0$ e $w_{i,q} > 0$, infere-se que $0 \leq \mathit{sim}(d_j, q) \leq 1$.

O uso de uma mesma representação, tanto para os documentos como para as expressões de busca, permite o cálculo do grau de similaridade entre dois documentos ou entre uma expressão e cada um dos documentos do *corpus* (FERNEDA, 2003, p. 30). A ordenação criada por este modelo permite restringir a quantidade de documentos recuperados, bastando restringir a um número máximo de resultados, ou por um valor mínimo de grau de similaridade.

Uma grande vantagem inicialmente atribuída a este modelo por Salton (1971 *apud* BAEZA-YATES; RIBEIRO-NETO, 1999, p. 28) se apresenta no uso do esquema de atribuição de pesos *tf-idf*. Assim, o modelo vetorial tenta balancear o número de ocorrências do termo no documento com o número de documentos onde ele aparece, ou seja, ele é proporcional ao número de ocorrências no documento e inversamente proporcional ao número de ocorrências em documentos de toda a coleção (MANNING; RAGHAVAN; SCHÜTZE, 2009, p. 125; BARTH, 2013, p. 251-252).

Por razão de os termos de indexação serem independentes, isto é, não terem relacionamentos existentes entre si, isto pode ser apontado como uma desvantagem

⁷ norma de um vetor: representado por $\|\vec{v}\|$ ou $|\vec{v}|$, é o número real não negativo calculado por $\sqrt{\sum x_i^2}$ (STEINBRUCH; WINTERLE, 1987, p. 40).

(FERNEDA, 2003, p. 31). Entretanto, Baeza-Yates e Ribeiro-Neto (1999, p. 30) declaram que não há evidências conclusivas que apontem que tais dependências afetam significativamente o desempenho de um SRI. Outra limitação ainda apontada neste modelo é não permitir a formulação de buscas booleanas, o que para Ferneda (2003, p. 31) se mostra como uma considerável restrição à flexibilidade do modelo.

Existe ainda uma prática abordada por Baeza-Yates e Ribeiro-Neto (2012, p. 49) que é a normalização de documentos. Ela se encarrega de resolver o problema que o tamanho (em conteúdo) de documentos é variável, tornando assim mais provável a recuperação de grandes arquivos do que de pequenos. O método para se normalizar o tamanho desses documentos depende da representação adotada que, no caso da representação vetorial em questão, é realizado por meio da norma de vetores.

O modelo Vetorial carrega consigo o grande mérito da definição de um dos componentes essenciais de qualquer teoria científica: um modelo conceitual. Este modelo serviu como base para o desenvolvimento de uma teoria que alimentou variadas pesquisas e resultou na implementação do sistema SMART, trabalho da vida de pesquisa de Gerald Salton, tendo um papel significativo no desenvolvimento de toda a área da Recuperação da Informação (SALTON, 1971 *apud* FERNEDA, 2003, p. 31-32). Por permitir o desenvolvimento de soluções simples e rápidas, o modelo vetorial é até hoje utilizado amplamente em soluções de indexação e pesquisa de documentos, como é o caso do SRI Lucene⁸ da Fundação Apache.

A Figura 15 utiliza os pesos relacionados pela Figura 8 e apresenta um exemplo de *ranking* da expressão “*to do*” realizado pelo método vetorial em um *corpus* contendo quatro documentos d_1 , d_2 , d_3 e d_4 .

⁸ Mais detalhes sobre o *software* Lucene podem ser obtidos diretamente na Fonte – <http://lucene.apache.org>.

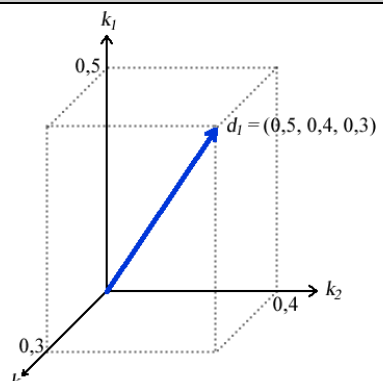
Figura 15 – Ranking da pesquisa pela expressão "to do"

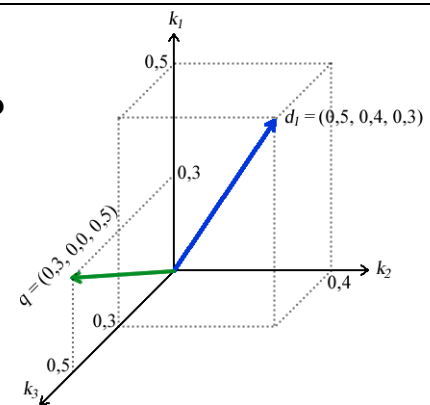
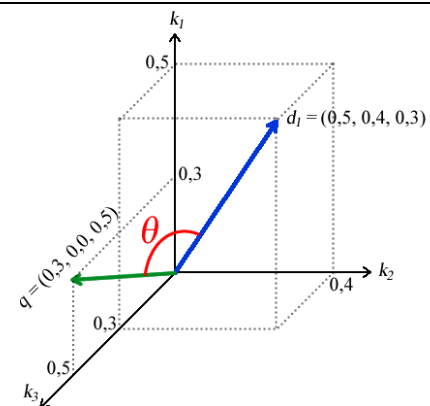
doc	cálculo do ranking	ranking
d_1	$\frac{1*3+0.415*0.830}{5.068}$	0.660
d_2	$\frac{1*2+0.415*0}{4.899}$	0.408
d_3	$\frac{1*0+0.415*1.073}{3.762}$	0.118
d_4	$\frac{1*0+0.415*1.073}{7.738}$	0.058

Fonte – Adaptado de Baeza-Yates e Ribeiro-Neto (2012, p. 59).

O Quadro 3 oferece de maneira resumida as representações e exemplos para documentos, consultas e função de classificação no modelo Vetorial. Nesta tabela podemos ver a representação do documento como um vetor $\vec{d}_1 = (0,5, 0,4, 0,3)$ com suas componentes do espaço tridimensional calculadas pelo esquema de atribuição de pesos *tf* ou *tf-idf*, como podemos ver no Quadro 3.a. No mesmo espaço vetorial podemos ver a representação lógica do vetor consulta $\vec{q} = (0,3, 0,0, 0,5)$, apresentado no Quadro 3.b. Por fim, o cálculo de similaridade entre a consulta e cada documento do espaço vetorial é calculado no Quadro 3.c, como demonstrado pelo cálculo da similaridade de $d_1 = 0,4242$ à consulta q .

Quadro 3 – Representações no modelo Vetorial

	REPRESENTAÇÃO	EXEMPLO
Documento	vetor em um espaço n -dimensional com cada componente sendo o peso atribuído por <i>tf</i> ou <i>tf-idf</i> .	 <p>a) $\vec{d}_1 = (0,5, 0,4, 0,3)$</p>

Consulta	<p>vetor em um espaço n-dimensional com cada componente sendo o peso atribuído por <i>tf</i> ou <i>tf-idf</i>. Cada termo da expressão de consulta é um eixo de um espaço vetorial.</p>  <p>b) $\vec{q} = (0,3, 0,0, 0,5)$</p>
Função de busca	<p>cálculo da distância euclidiana ou do valor do cosseno do ângulo formado entre cada vetor documento e o vetor consulta.</p> $sim(q, d_1) = cos \theta = \frac{ \vec{q} \cdot \vec{d}_1 }{\ \vec{q}\ \times \ \vec{d}_1\ }$  <p>c) $sim(\vec{q}, \vec{d}_1) = \frac{(0,15+0,0+0,15)}{(\sqrt{0,3^2+0,0^2+0,5^2} \times \sqrt{0,5^2+0,4^2+0,3^2})} = \frac{0,3}{\sqrt{0,34} \times \sqrt{0,5}} = \frac{0,3}{0,7071} = 0,4242$</p>

Fonte – Adaptado de Baeza-Yates e Ribeiro-Neto (2012), Ferneda (2003), Amazonas *et al.* (2008), Mausam (2012) e Dominich (2008).

2.7.3 Modelo Probabilístico

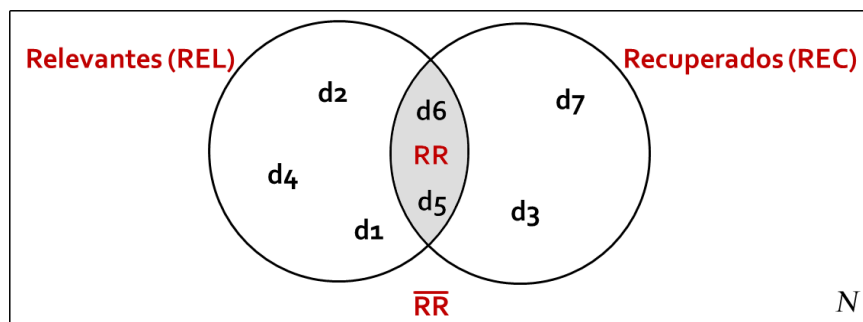
A Recuperação da Informação Probabilística é baseada na ideia de que a probabilidade da relevância de um documento relativo à consulta é maior que sua probabilidade de irrelevância (DOMINICH, 2008, p. 218). Segundo Maron e Juhns (1960 *apud* DOMINICH 2008, p. 2018), desde que um sistema de recuperação não possa dizer com certeza qual documento é relevante, deveríamos portanto lidar com probabilidades. Contrários à suposição de Luhn⁹, Maron e Juhns (1960) apresentaram seu modelo de indexação probabilístico

⁹ Luhn supôs que cada termo pudesse ter seu peso atribuído pela frequência de ocorrência em um documento, calculada automaticamente pelo SRI (KOCABAS; DINÇER; KARAOGLAN, 2011, p. 993-994).

sugerindo que a indexação humana usando a teoria das probabilidades seria mais eficaz (HIEMSTRA, 2009, p. 10). Baseado nos estudos de Maron e Juhns, o modelo Probabilístico que realmente se destacou no campo da Recuperação da Informação foi o modelo proposto por Robertson e Jones (1976, p. 129-146), posteriormente conhecido como *Binary Independence Retrieval* (HIEMSTRA, 2009, p. 11).

Como observado na Figura 16, dada uma expressão de busca podemos dividir o *corpus* em quatro subconjuntos distintos: o conjunto dos documentos relevantes (**REL**), o conjunto dos documentos recuperados (**REC**), o conjunto dos documentos relevantes que foram recuperados (**RR**) e o conjunto dos documentos não relevantes e não recuperados (\overline{RR}). O conjunto dos documentos relevantes e recuperados (**RR**) é resultante da interseção dos conjuntos **REL** e **REC** (FERNEDA, 2003, p. 38-39), isto é, o resultado ideal de uma busca é o conjunto que contenha todos e apenas os documentos relevantes para o usuário (**RR**), buscando-se portanto fazer com que **REC** seja igual a **REL**.

Figura 16 – Subconjuntos de documentos após consulta em um SRI probabilístico



Fonte – Adaptada de Baeza-Yates e Ribeiro-Neto (2012, p. 65).

Para Amazonas *et al.* (2008, p. 199), o modelo Probabilístico parte da suposição de que exista um conjunto ótimo de documentos para cada pesquisa do usuário, sendo este passível de recuperação. O modelo busca obter tal conjunto ótimo de documentos utilizando inicialmente outro método de recuperação a fim de obter uma lista inicial de documentos relevantes e, a partir dela, realizando interações sucessivas com os usuários, fazer análises de relevância dos documentos a serem retornados. A cada iteração o usuário verifica os documentos recuperados e decide quais deles são relevantes e quais não são, processo denominado *relevance feedback* (DOMINICH, 2008, p. 222). Na prática é necessário examinar apenas os primeiros 10 ou 20 documentos retornados, segundo Baeza-Yates e Robeiro-Neto (2012, p. 63). O modelo então

utiliza as marcações do usuário para refinar a descrição do conjunto ideal de resposta (BAEZA-YATES; RIBEIRO-NETO, 1999, p. 31).

Segundo Baeza-Yates e Ribeiro-Neto (2012, p. 65), seja **REL** o conjunto de documentos relevantes e \overline{REL} o complemento de **REL**, ou seja, o conjunto dos documentos não relevantes, a similaridade de um documento **d** em relação à expressão de busca **q** é definida como:

$$sim(d, q) = \frac{P(REL | d)}{P(\overline{REL} | d)} \quad (8)$$

onde

- $sim(d, q)$: grau de similaridade entre o documento **d** e a consulta **q**, representado por um valor numérico real.
- $P(REL | d)$: probabilidade de um documento **d** ser relevante em relação à expressão de busca **q**.
- $P(\overline{REL} | d)$: probabilidade de um documento **d** ser considerado não relevante à expressão de busca **q**.

Aplicando-se a regra de Bayes, da Teoria das Probabilidades (MITCHEL, 1997, p. 156; BAEZA-YATES; RIBEIRO-NETO, 2012, p. 66), tem-se:

$$sim(d, q) = \frac{P(d | REL) \times P(REL)}{P(d | \overline{REL}) \times P(\overline{REL})} \quad (9)$$

onde

- $sim(d, q)$: grau de similaridade entre o documento **d** e a consulta **q**, representado por um valor numérico real.
- $P(d | REL)$: probabilidade de se selecionar o documento **d** do conjunto de documentos relevantes **REL**.
- $P(d | \overline{REL})$: probabilidade de se selecionar o documento **d** do conjunto dos documentos não relevantes.
- $P(REL)$: probabilidade de um documento selecionado aleatoriamente ser relevante.
- $P(\overline{REL})$: probabilidade de um documento não ser relevante.

Conforme observa Ferneda (2003, p. 40), considerando-se que $P(REL)$ e $P(\overline{REL})$ são iguais para todos os documentos do *corpus*, a fórmula da similaridade apresentada na Equação 9 é reduzida e apresentada de acordo com a Equação 10.

$$sim(d, q) = \frac{P(d | REL)}{P(d | \overline{REL})} \quad (10)$$

Barth (2013, p. 253) ainda ressalta que o peso atribuído para os termos em um modelo probabilístico são todos binários, por exemplo, $w_{i,j} \in \{0, 1\}$ e $w_{i,q} \in \{0, 1\}$, onde $w_{i,j}$ representa o peso do termo i no documento j e $w_{i,q}$ representa o peso do termo i na consulta q .

Por fim, a fórmula de *ranking* do modelo probabilístico é dada na Equação 11 por:

$$sim(d, q) = \sum_{i=1}^t \left(\log \frac{P(k_i | REL) \times P(\bar{k}_i | \overline{REL})}{P(k_i | \overline{REL}) \times P(\bar{k}_i | REL)} \right) \quad (11)$$

onde

- $sim(d, q)$: grau de similaridade entre o documento d e a consulta q , representado por um valor numérico real.
- t : número total de termos da expressão de consulta.
- $P(k_i | REL)$: probabilidade de um termo k_i estar presente em um documento selecionado do conjunto de relevantes REL .
- $P(\bar{k}_i | REL)$: probabilidade do termo k_i não estar presente em um documento selecionado do conjunto de relevantes REL .
- $P(k_i | \overline{REL})$: probabilidade de um termo k_i estar presente em um documento selecionado do conjunto de não relevantes \overline{REL} .
- $P(\bar{k}_i | \overline{REL})$: probabilidade do termo k_i não estar presente em um documento selecionado do conjunto de não relevantes \overline{REL} .

Observa-se ainda que $P(k_i | REL) + P(\bar{k}_i | REL) = 1$. A Equação 11 ignora fatores que são constantes para todos os documentos no contexto de uma mesma busca (FERNEDA, 2003, p. 40).

Segundo Manning, Raghavan e Schütze (2009, p. 224), Baeza-Yates e Ribeiro-Neto (2012, p. 73), cálculos de probabilidade resumem-se a um problema de contagem. Portanto, para uma determinada expressão de busca, pode-se representar os documentos da coleção por meio das expressões numéricas dadas no Quadro 4. Considerando-se um *corpus* com N documentos e um determinado termo k_i , existe no *corpus* um total de n_i documentos indexados por k_i . Desses n documentos apenas r são relevantes (FERNEDA, 2003, p. 40-41).

Quadro 4 – Tabela auxiliar de incidência de termos

	RELEVANTE	NÃO RELEVANTE	TODOS OS DOCs
Documentos que contêm k_i	r_i	$n_i - r_i$	n_i
Documentos que não contêm k_i	$R - r_i$	$N - n_i - (R - r_i)$	$N - n_i$
Todos os documentos	R	$N - R$	N

Fonte – Adaptado de Manning, Raghavan e Schütze (2009, p. 224) e, Baeza-Yates e Ribeiro-Neto (2012, p. 73).

A Equação 11 pode ser reescrita usando a mesma notação apresentada na Tabela 5 pela fórmula:

$$sim(d, q) = \sum_{i=1}^t \left(\log \frac{r_i \times (N - R - n_i + r_i)}{(n_i - r_i) \times (R - r_i)} \right) \quad (12)$$

onde

- $sim(d, q)$: grau de similaridade entre o documento d e a consulta q , representado por um valor numérico real.
- t : número total de termos da expressão de consulta q .
- N : conjunto de todos os documentos no *corpus*.
- n_i : número de documentos que contém o termo k_i .
- R : conjunto de todos os documentos relevantes à consulta q .
- $r_i = P(k_i | REL)$: número de documentos relevantes que contém o termo k_i .

Para exemplificar, Ferneda (2003, p. 41) apresenta uma instância do processo de *ranking* do modelo Probabilístico por meio de um *corpus* contendo seis documentos, **DOC₁** a **DOC₆**, dez termos de indexação, k_1 a k_{10} , e uma consulta q , reproduzido na Figura 17.

Figura 17 – *Corpus* composto por seis documentos e dez termos

	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9	k_{10}
DOC₁	1	0	0	1	0	0	0	1	1	0
DOC₂	0	0	0	0	0	0	0	1	1	1
DOC₃	0	1	0	0	0	1	1	0	0	0
DOC₄	1	0	0	1	0	0	0	0	0	1
DOC₅	0	0	0	0	0	0	0	1	1	0
DOC₆	0	0	1	0	1	0	0	0	0	0
q	0	0	0	1	0	0	0	0	0	1

Fonte – Adaptada de Ferneda (2003, p. 41).

Pela Figura 17, a expressão de busca q é composta apenas pelos termos k_4 e k_{10} . Como na primeira iteração do processo não se sabe qual o conjunto de documentos relevantes R , algumas simplificações são aplicadas na Equação 12 (BAEZA-YATES; RIBEIRO-NETO, 1999, p. 33), a saber:

- Assumir que $P(k_i | REL)$ é constante e igual a 0,5 para todos os termos k_i .
- Assumir que a distribuição dos termos de indexação dos documentos (relevantes ou não) é uniforme.

Levando-se em consideração as modificações propostas, obtém-se portanto a primeira lista de documentos a partir da simplificação da Equação 12 (ver Equação 13). O resultado desta etapa de *relevance feedback* sobre a instância apresentada é ilustrada na Figura 18.

$$sim(d, q) \approx \sum_{i=1}^t \left(\log \frac{N-n}{n} \right) \quad (13)$$

onde

- $sim(d, q)$: grau de similaridade entre o documento d e a consulta q , representado por um valor numérico real.
- t : número total de termos da expressão de consulta q .
- N : conjunto de todos os documentos no *corpus*.
- n : número de documentos que contém o termo k_i .

Figura 18 – Etapa de *relevance feedback*

		k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9	k_{10}	$sim(\text{DOC}_b, q)$
<input checked="" type="checkbox"/>	DOC₄	1	0	0	1	0	0	0	0	0	1	0.51
	DOC₁	1	0	0	1	0	0	0	1	1	0	0.26
<input checked="" type="checkbox"/>	DOC₂	0	0	0	0	0	0	0	1	1	1	0.26

Fonte – Adaptada de Ferneda (2003, p. 42).

A Figura 18 apresenta o primeiro resultado da busca com apenas três documentos recuperados, d_4 , d_1 e d_2 , onde se verifica que, apesar do sistema atribuir um grau de similaridade igual para os documentos d_1 e d_2 , o usuário não selecionou o documento d_1 como relevante. Assim, após a submissão de nova consulta com a mesma expressão de busca, o modelo de recuperação de informações probabilístico recalcula o valor de similaridade utilizando a Equação 12, obtendo-se como resultado o *ranking* apresentado na Figura 19.

Figura 19 – Resultado da segunda iteração do modelo Probabilístico

		k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9	k_{10}	$sim(\text{DOC}_b, q)$
	DOC₄	1	0	0	1	0	0	0	0	0	1	2.02
	DOC₂	0	0	0	0	0	0	0	1	1	1	1.65
	DOC₁	1	0	0	1	0	0	0	1	1	0	0.37

Fonte – Adaptada de Ferneda (2003, p. 42).

Observando a Figura 19 entende-se que a iteração anterior apresentava a ordem dos identificadores numéricos dos documentos como 4, 1, 2. Para a iteração seguinte, e considerando-se a etapa de *relevance feedback* realizada anteriormente, a ordem de classificação é alterada, invertendo-se os documentos 1 e 2, resultando um *ranking* de documentos em: 4, 2, 1.

O processo de *relevance feedback* agregado à uma nova consulta deve ser repetido até que o usuário fique satisfeito com os resultados obtidos (MANNING; RAGHAVAN; SCHÜTZE, 2009, p. 229).

A principal vantagem do modelo Probabilístico é a ordenação decrescente dos documentos por suas probabilidades de serem relevantes. Algumas evidências indicam que este modelo tem um desempenho melhor que o modelo vetorial (COOPER, 1994; BAEZA-YATES; RIBEIRO-NETO, 1999, p. 34; JONES; WALKER; ROBERTSON, 2000). Dentre as desvantagens destacam-se: (i) o modelo não faz uso da frequência dos termos no documento, e; (ii) assume-se a premissa de independência entre termos. De qualquer forma, Baeza-Yates e Ribeiro-Neto (1999 p. 30) discutem que não há consenso se a premissa de independência entre termos é ruim em situações práticas.

Um exemplo concreto da aplicação do modelo Probabilístico é a função de ordenação *Best Match 25* (BM25) (BAEZA-YATES; RIBEIRO-NETO, 2012, p. 165; MANNING, RAGHAVAN; SCHÜTZE, 2009, p. 232-234). Esta função de ordenação faz uso da frequência dos termos no documento e também leva em consideração o tamanho dos documentos. No BM25 o cálculo da similaridade entre documentos e a consulta é dada pela Equação 14:

$$\mathit{sim}(d, q) \approx \sum_{k \in q} \log \frac{N}{n_i} \times \frac{(c + 1) \times f_{k,d}}{c \left((1 - b) + b \times \left(\frac{T_d}{T_{med}} \right) \right) + f_{k,d}} \quad (14)$$

onde

- $\mathit{sim}(d, q)$: grau de similaridade entre o documento d e a consulta q , representado por um valor numérico real.
- N : conjunto de todos os documentos no *corpus*.
- n : número de documentos que contém o termo k_i .
- k : termo da expressão de consulta q .
- $f_{k,d}$: frequência do termo k no documento d .
- T_d : tamanho do documento d .
- T_{med} : tamanho médio dos documentos na coleção.
- c : parâmetro de valor positivo que calibra a escala do $f_{k,d}$. Se c igual a 0 então o retorno de $\mathit{sim}(q,d)$ é similar ao resultado do modelo Booleano.

- b : parâmetro ($0 \leq b \leq 1$) que determina a influência do tamanho dos documentos no cálculo da $sim(q, d)$. Quando $b = 0$ o valor de $sim(d, q)$ não é normalizado considerando T_d e T_{med} .

Segundo Baeza-Yates e Ribeiro-Neto (2012, p. 165), o BM25 foi criado como o resultado de uma série de experimentos em variações do modelo Probabilístico utilizando-se o sistema OKAPI, definido por HjØrland (2006) como uma fórmula constituída de meia dúzia de variáveis buscando estimar a probabilidade de um determinado documento ser relevante a uma determinada consulta, usando para isso o esquema de atribuição de pesos *term frequency*. Para maiores detalhes sobre o sistema OKAPI, consultar diretamente na fonte HjØrland (2006).

Assim, ao contrário do modelo clássico Probabilístico, a fórmula do BM25 pode ser calculada sem o uso da informação de relevância do usuário (*relevance feedback*). É um consenso que o BM25 tem performance melhor que o modelo clássico vetorial para a maioria das coleções onde foi aplicado. Logo, esta tem sido a função padrão para a avaliação de novas funções de *ranking*, em substituição ao modelo clássico vetorial (BAEZA-YATES; RIBEIRO-NETO, 2012, p. 176).

2.8 Avaliação em Recuperação da Informação

As medidas mais comuns para avaliar o desempenho de um sistema computacional são tempo e espaço (KNUTH, 1997, p. 107). Quanto menor o tempo de resposta de um sistema e quanto menor o espaço em memória utilizado, melhor o sistema é considerado. No entanto, para sistemas onde o objetivo é recuperar informações, outras medidas devem ser utilizadas. Segundo Baeza-Yates e Ribeiro-Neto (1999, p. 444):

[...] em termos de qualidade dos resultados obtidos no que diz respeito a um conjunto de consultas de teste, uma avaliação de um Sistema de Recuperação da Informação mede a qualidade por meio da comparação de documentos presentes nos resultados com aqueles em um conjunto (apontado por especialistas) conhecidamente rotulado como relevante para a consulta em teste. (tradução e adaptação elaborada pelo autor)

Não existe uma resposta exata à consulta realizada pelo usuário. O SRI é responsável pela recuperação e ordenação dos documentos de acordo com a sua relevância à consulta. As

medidas utilizadas para avaliar um SRI devem focar em medir o quão relevante é o resultado obtido para o usuário (BAEZA-YATES; RIBEIRO-NETO, 1999, p. 74-82).

Existem coleções de teste padronizadas para comparação de eficiência de SRIs, as quais devem possuir os seguintes elementos (MANNING; RAGHAVAN; SCHÜTZE, 2009, p. 153; LUGO, 2004, p. 24-25):

- **Um conjunto de documentos**, que pode conter somente alguns dados como título, autor e resumo ou então o texto completo. Podem ser utilizadas informações adicionais, tais como um conjunto de termos usado como vocabulário de controle, descritores designados por autor e informação sobre citações (ver Tabela 2, segunda coluna).
- **Um conjunto de consultas**, exemplos de requisição de informações, constituídas por consultas reais submetidas por usuários, seja usando linguagem natural ou alguma linguagem formal de consulta. Ocasionalmente pode-se utilizar consultas artificialmente construídas para recuperar documentos conhecidos ou o texto de um documento usado como amostra, por exemplo (exemplificado pela Tabela 2 como o número identificador de tópicos).
- **Um conjunto de julgamentos de relevância**: para cada consulta existente no conjunto de consultas são fornecidos documentos, pertencentes à coleção de documentos, considerados relevantes pelos usuários que submetem a consulta ou por especialistas do domínio. Para coleções pequenas, isto pode ser obtido revisando-se todos os documentos. Para coleções grandes, geralmente são combinados os resultados de diferentes representações da consulta construída por diferentes usuários (ver Tabela 2, terceira coluna).

A Tabela 2 a seguir apresenta um exemplo de julgamento de relevância, com identificador do tópico, um documento de texto e seu grau de relevância ao termo julgado por um especialista no domínio de abordagem do teste.

Tabela 2 – Exemplo de julgamento de relevância

Identificador do tópico	Documento	Relevância
1	CSIRO135-03599247	2
1	CSIRO141-07897607	1
...
50	CSIRO265-01044334	0
50	CSIRO265-01351359	1

Tabela 2	(continuação)	
50	CSIRO266-04184084	0

Fonte – Barth (2010, p. 15).

Assim, a Tabela 2 apresenta o *benchmark* resultado do julgamento de relevância por um especialista. Para tal julgamento, se faz necessário o uso de um *dataset*, como apresentado pelo Quadro 5, com dois tópicos de exemplo do TREC 2007 *Enterprise Track*¹⁰.

Quadro 5 – Exemplo do *benchmark* TREC 2007 Enterprise Track

<pre> <top> <num>CE-001</num> <query>genetic modification</query> <narr> Over arching information on gene technology / biotechnology. Specific pages on certain GM (e.g. cotton). </narr> <page>CSIRO135-03599247</page> <page>CSIRO141-08973435</page> <page>CSIRO141-07897607</page> </top> <top> <num>CE-002</num> <query>hairpin RNAi / gene silencing</query> <narr> Information to help scientists find out more about hairpin RNAi technology. Specific contacts to obtain vectors. </narr> <page>CSIRO197-05231046</page> <page>CSIRO139-13111797</page> <page>CSIRO145-13752815</page> </top> </pre>
--

Fonte – Trec (2007) *Enterprise topics*.

Observa-se que por uma estrutura de rótulos, denominados *tags* em uma linguagem de marcação, os dados são mapeados para cada tópico, sendo a *tag* **<num>** para armazenar o identificador do tópico, a *tag* **<query>** para referenciar a consulta a qual se submeteu, **<narr>** a narrativa ou contexto para se julgar documentos como relevantes à consulta e por fim **<page>** como a lista de documentos considerados relevantes¹¹ (TREC, 2007). Ressalta-se que a coluna

¹⁰ TREC 2007 Enterprise topics (CE001-CE050). Mais informações em: http://trec.nist.gov/data/t16_enterprise.html.

¹¹ TREC data: mais informações disponíveis em: http://trec.nist.gov/data/topics_eng/topics.501-550.txt

Relevância existente na Tabela 2 é o resultado do processo de julgamento por um especialista, processo este subjetivo e com valor atribuído, neste exemplo, numérico.

Manning, Raghavan e Schütze (2009, p. 153-154) listam ainda outras coleções ou entidades que oferecem coleções padrão mais importantes no teste e avaliação de SRIs, cujos quais se destacam:

- **Cranfield:** coleção com 1400 documentos e 225 consultas. Mais informações em: http://ir.dcs.gla.ac.uk/resources/test_collections/cran/;
- **Text REtrieval Conference (TREC):** conferência que encoraja a pesquisa em recuperação da informação usando coleções gigantes. Oferece uma vasta quantidade de coleções para uso de testes. Mais informações em: <http://trec.nist.gov/data.html>;
- **GOV2:** é uma das coleções disponibilizadas para teste pela TREC. Consiste de vários arquivos de sites do governo, somando um total de 25.205.179 documentos que formam 426 Gigabytes de dados. Mais informações em: http://ir.dcs.gla.ac.uk/test_collections/gov2-summary.htm;
- **NII Test Collections for IR Systems (NTCIR):** série de *workshops* para a promoção da pesquisa em Recuperação da Informação. Oferece 11 conjuntos numerados de NTCIR-1 até NTCIR-11, cada um com particularidades de uso ou área de informação;
- **Cross Language Evaluation Forum (CLEF):** associação independente sem fins lucrativos com objetivos científicos, culturais e educacionais na área de sistemas de acesso à informação. Para mais informações consultar a coleção em: <http://www.clef-initiative.eu/dataset/test-collection>.

Baeza-Yates e Ribeiro-Neto (2012, p. 2) resumem a avaliação de SRIs à medida de quão bem os sistemas atendem à necessidade informacional dos usuários. Esta não é uma medida trivial de se conseguir, afinal resultados podem ser relevantes para alguns usuários e para outros não. Ávila (2014, p. 5) discute que do ponto de vista do usuário a relevância é **subjetiva**, pois depende de um julgamento específico do usuário; **dependente do contexto**, pois relaciona-se às necessidades atuais do usuário; **cognitiva**, pois depende da percepção e comportamento humano; e **dinâmica**, pois muda com o decorrer do tempo. Isto posto, identifica-se a dificuldade de realizar uma avaliação sob o ponto de vista informacional.

Dado um algoritmo de recuperação de informações, as medidas de avaliação devem quantificar a similaridade entre o conjunto de documentos recuperados e o conjunto de

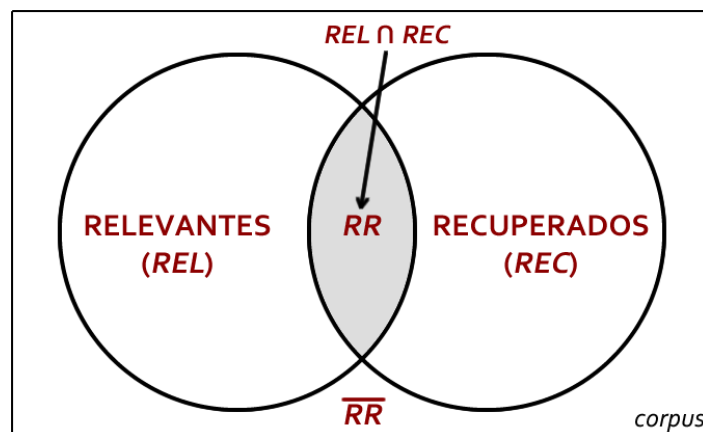
documentos considerados relevantes pelos especialistas, fornecendo assim uma estimativa da qualidade do algoritmo de recuperação da informação avaliado (BARTH, 2013, p. 262). A avaliação de um Sistema de Recuperação da Informação é um componente crítico e deve integrar toda implementação moderna de um Sistema de Recuperação de Informação (BAEZA-YATES; RIBEIRO-NETO, 2012, p. 2).

2.8.1 Medidas para avaliação de SRIs

As duas medidas mais frequentemente usadas na avaliação do desempenho de SRIs são: **precisão** (do inglês: *precision*) e **cobertura** (do inglês: *recall*), também abordada na literatura como revocação (MANNING; RAGHAVAN; SCHÜTZE, 2009, p. 155). Para explicar cada uma delas serão usados os elementos apresentados na Figura 20. Sejam:

- q : uma expressão de busca;
- *corpus* : conjunto de todos os documentos da coleção.
- **REL** : conjunto dos documentos relevantes a q .
- **REC** : conjunto de documentos recuperados, gerado por um SRI.
- **RR** : interseção entre os conjuntos de documentos relevantes e recuperados, formando o conjunto dos documentos relevantes recuperados.

Figura 20 – Cobertura e precisão



Fonte – Elaborada pelo autor.

A precisão é definida como a fração dos documentos recuperados (**REC**) que é relevante (BAEZA-YATES; RIBEIRO-NETO, 2012, p. 12). Logo, quanto mais próximo seja o conjunto **REC** do conjunto **REL**, mais próximo do valor 1 será o valor calculado da precisão conforme Equação 15:

$$precisão = \frac{|relevantes \cap recuperados|}{|recuperados|} \quad (15)$$

A cobertura é a fração dos documentos relevantes (**REL**) que foi recuperada (BAEZA-YATES; RIBEIRO-NETO, 2012, p. 12). Da mesma forma que a precisão, quanto mais próximo seja o conjunto **REC** do conjunto **REL**, mais próximo do valor 1 será o valor calculado da cobertura. O cálculo da cobertura é simples, e pode ser obtido pela razão apresentada na Equação 16:

$$cobertura = \frac{|relevantes \cap recuperados|}{|relevantes|} \quad (16)$$

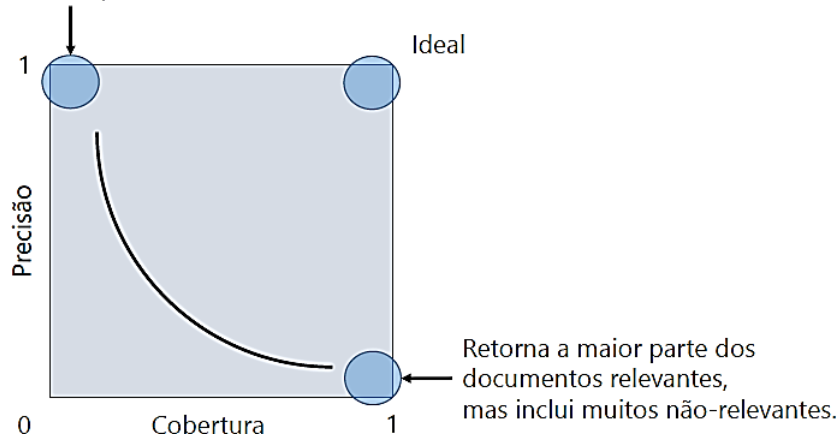
Ávila (2014, p. 7) conclui que a precisão se resume à habilidade do SRI recuperar **somente** itens relevantes e a cobertura se resume à habilidade do SRI recuperar **todos** os itens relevantes. Observamos que no caso extremo, um sistema que retorna todos os documentos do *corpus* como seu conjunto de resultados tem a garantia de uma cobertura igual a 100%, mas terá baixa precisão. Por outro lado, um sistema pode retornar um único documento e ter um baixo índice de cobertura, mas teria uma chance razoável de 100% de precisão (RUSSEL; NORVIG, 2003, p. 843).

Para Baeza-Yates e Ribeiro-Neto (1999, p. 75-76) as definições de precisão e cobertura assumem que todos os documentos do conjunto **REL** foram examinados ou vistos pelo usuário. Entretanto, ao usuário não são apresentados todos os documentos do conjunto de relevantes de uma só vez, sendo necessária a inspeção visual, denominada *browsing*. No *browsing* o resultado entregue é visto de cima para baixo, em ordem decrescente de classificação. Esta etapa de inspeção dentre os resultados apresentados impacta diretamente na qualidade da precisão e cobertura, pois elas variam de acordo com a inspeção do conjunto de documentos relevantes feito pelo usuário. Baeza-Yates e Ribeiro-Neto (2012, p. 13) sugerem ser apropriado o uso da

curva de cobertura e precisão, que tem um comportamento teórico descrito conforme o modelo dado pela Figura 21. A curva de cobertura e precisão e seu significado para a recuperação de informações serão comentados nos parágrafos que seguem.

Figura 21 – Curva de cobertura e precisão

Retorna documentos que são relevantes, mas esquece muitos outros relevantes.

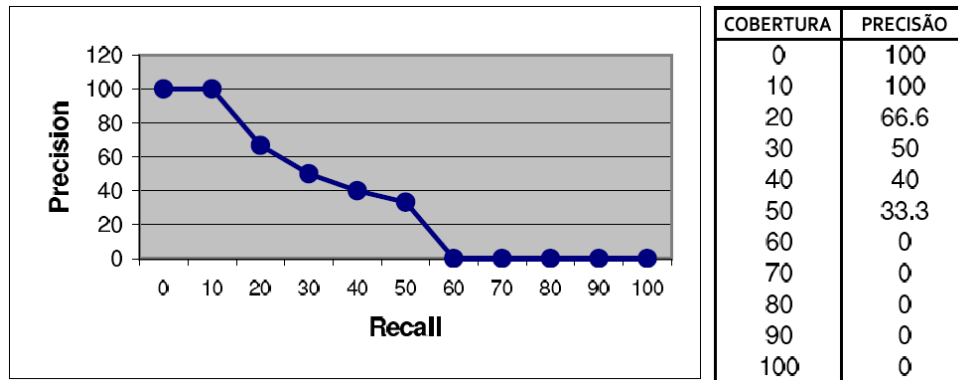


Fonte – Ávila (2014, p. 14).

A Figura 21 apresenta um modelo conceitual, teórico e ideal da curva de cobertura e precisão. A dimensão cobertura é marcada no eixo das abscissas, enquanto que a precisão é marcada no eixo das ordenadas. A curva cobertura e precisão mostra o comportamento do SRI pela sua precisão a cada nível de cobertura; em teoria é esperado que conforme a cobertura da pesquisa aumenta, reduz-se a precisão dos resultados, afinal alguns documentos não relevantes são esperados em meio aos resultados.

Observa-se ainda que tanto a precisão quanto a cobertura atingem valores máximos de 1, o que equivale a um casamento de 100%. A medida de comparação entre dois SRIs consiste na análise de ambas as medidas cobertura e precisão: o SRI cuja curva mais se aproximar do canto superior direito é o SRI com melhor desempenho, afinal objetiva-se por esta análise atingir o máximo de cobertura e o máximo de precisão possível. A Figura 22 apresenta um exemplo com valores para cobertura e precisão do resultado ordenado e classificado de uma busca hipotética e, junto a estes valores, a curva de cobertura e precisão correspondente.

Figura 22 – Gráfico de cobertura e precisão com valores de exemplo

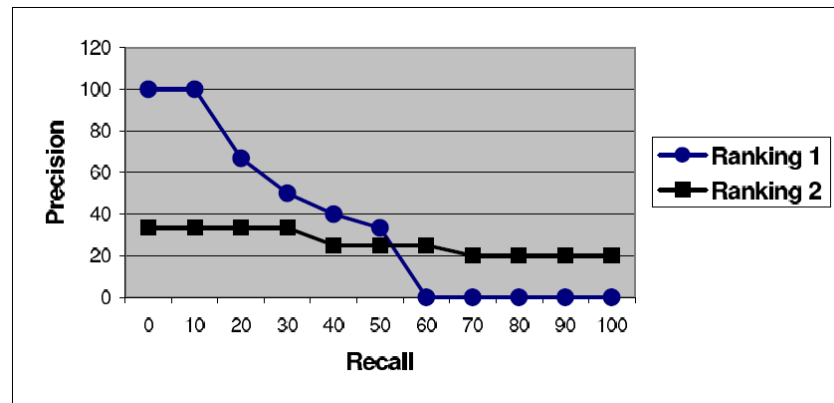


Fonte – Adaptada de Baeza-Yates e Ribeiro-Neto (2012, p. 19).

Observando o exemplo da Figura 22, a curva de cobertura e precisão deve ser interpretada da seguinte forma: até os primeiros 10% de cobertura dos documentos relevantes a precisão do SRI analisado foi de 100%, ou seja, estes 10% de documentos relevantes recuperados até então participam do conjunto dos documentos relevantes (*REL*) e recuperados (*REC*), nenhum não relevante foi recuperado até este momento. Considerando-se o nível de cobertura em 20%, a precisão do referido SRI caiu a 66,6%, o que mostra que alguns dos documentos recuperados ao se atingir a cobertura de 20% (dos documentos julgados relevantes) não estão entre os documentos relevantes. A análise do gráfico se segue desta mesma forma até que, por fim, o resultado apresentado ao final, quando a cobertura atinge 100% traz uma precisão tão pequena que se aproxima do zero, tornando a curva do gráfico de cobertura e precisão praticamente metade em precisão igual a zero, o que mostra que o SRI sob análise não teve um bom desempenho no quesito precisão, causado por não conseguir recuperar todos os documentos julgados relevantes previamente junto ao conjunto de recuperados e apresentado ao usuário.

Para comparação, o Gráfico 1 traz duas curvas de cobertura e precisão representando duas funções de *ranking* de diferentes métodos hipotéticos, ilustrando assim a comparação entre eles.

Gráfico 1 – Curvas de cobertura e precisão representando duas funções de ranking



Fonte – Adaptada de Baeza-Yates e Ribeiro-Neto (2012, p. 19).

Observando-se o Gráfico 1, percebe-se que a função de *Ranking 1* apresenta uma precisão maior até o valor de cobertura de 55%. Entretanto, esta curva atinge a precisão igual a zero ao chegar aos 60% de cobertura dos documentos relevantes. Isto nos quer dizer que a precisão desta função de *ranking* é, no mínimo, 40% não precisa. Por outro lado, a função de *Ranking 2* apresenta uma precisão baixa por todos os níveis de cobertura, porém quase constante. Isto significa que sua cobertura atinge os 100% mantendo quase que constantemente baixa sua precisão, o que também não é um ótimo resultado. Assim, a análise de gráficos como este tem a aplicação prática subjetiva ao desenvolvedor, afinal em alguns sistemas pode-se valorizar mais a precisão dos documentos obtidos, ou o mínimo de documentos não relevantes, em outros, pode-se valorizar mais a cobertura, por exemplo: desde que se atinja a cobertura de 100%, ou seja, todos os documentos relevantes sejam recuperados, não importa a precisão. Com relação ao canto superior direito, no caso do Gráfico 1 metade da cobertura foi de melhor desempenho do *Ranking 1* e a segunda metade de melhor desempenho do *Ranking 2*.

As medidas de precisão e cobertura foram definidas quando as pesquisas de Recuperação da Informação eram feitas principalmente por bibliotecários interessados em resultados completos. Atualmente, a maioria das consultas (centenas de milhões por dia) são feitas por usuários que estão menos interessados em perfeição e mais interessados em encontrar uma resposta imediata, uma resposta que apareça no topo da lista de resultados (RUSSEL; NORVIG, 2003, p. 844).

Segundo Manning, Raghavan e Schütze (2009, p. 156) existe uma forma de se definir uma média harmônica entre precisão e cobertura, usando a *medida F* (do inglês: *F-measure*) cujo resultado é dado pela Equação 17:

$$F = \frac{2 \times (\textit{precisão} \times \textit{cobertura})}{\textit{precisão} + \textit{cobertura}} \quad (17)$$

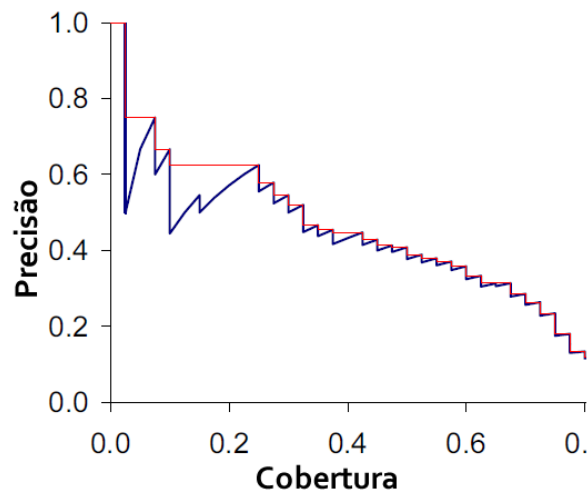
Precisão, cobertura e medida F são medidas baseadas em conjuntos, computadas usando um conjunto de documentos não ordenados. Estas medidas não são suficientes para medir o desempenho da maioria dos SRI atuais, que fornecem um resultado ordenado segundo algum critério (MANNING; RAGHAVAN; SCHÜTZE, 2009, p. 158).

Atualmente existem medidas de avaliação de SRIs mais aprimoradas e que, segundo Baeza-Yates, Ribeiro-Neto (2012, p. 25-41) e Barth (2013, p. 264-268), fornecem resultados ordenados. Os autores citam algumas delas: precisão em n: P@5, P@10 (para máquinas de busca na *web*), *Mean Average Precision* (MAP), *R-Precision*, *Mean Reciprocal Rank* (MRR) e *Normalized Discount Cumulative Gain* (NDCG). Maiores detalhes sobre cada uma dessas medidas podem ser obtidas diretamente nas fontes supracitadas.

Visto que as medidas de precisão, cobertura e medida F são utilizadas para conjuntos não ordenados de resultados, existe uma forma de tornar tais medidas listas ordenadas, podendo ser utilizadas em resultados com *ranking*. Para isso, segundo Bernardi (2012, p. 17-20), basta computar para cada resultado as medidas de acordo com o maior valor de precisão encontrado. Segundo a autora, se na ordem de recuperação o documento abaixo do documento analisado não é relevante, então a cobertura permanece inalterada, o maior valor encontrado até então permanece inalterado, mas o valor da precisão é reduzido. Caso contrário, se o documento é relevante, então tanto a cobertura quanto a precisão aumentam, levando a curva de cobertura e precisão para a direita e para cima, respectivamente, nos eixos das abcissas e ordenadas.

De acordo com Manning, Raghavan e Schütze (2009, p. 158), é bom se remover tais variações pelo uso da precisão interpolada, ilustrada pela linha em vermelho da curva de cobertura e precisão no Gráfico 2. Esta nova curva foi denominada pelos autores de **gráfico de cobertura e precisão média interpolada em 11 pontos**, sendo os 11 pontos níveis de cobertura de 0.0 a 1.0, representando de 0 a 100% de cobertura do resultado em questão (BERNARDI, 2012, p. 21).

Gráfico 2 – Exemplo de gráfico de cobertura e precisão média interpolada



Fonte – Adaptada de Manning, Raghavan e Schütze (2009, p. 158).

A construção dessa curva é feita pelos dados apresentados no Quadro 6, que serão melhor detalhados adiante.

Quadro 6 – Exemplo dos cálculos para construção da curva de cobertura e precisão média interpolada em 11 pontos

Documentos Relevantes (1)	Ranking (2)	ID documento (3)	Cobertura (4)	Precisão no nível de cobertura (5)	Nível de cobertura (6)	Precisão interpolada (7)
0123	1	0234	0		0%	0.5
0132	2	0132	0.111	0.5	10%	0.5
0241	3	0115	0.111		20%	0.4
0256	4	0193	0.111		30%	0.4
0299	5	0123	0.222	0.4	40%	0.4
0311	6	0345	0.222		50%	0
0324	7	0387	0.222		60%	0
0357	8	0256	0.333	0.375	70%	0
0399	9	0078	0.333		80%	0
	10	0311	0.444	0.4	90%	0
	11	0231	0.444		100%	0
	12	0177	0.444			

Fonte – Adaptada de Bernardi (2012, p. 23).

O Quadro 6 apresenta o passo-a-passo para construção de gráficos de cobertura e precisão média interpolada em 11 pontos como aquele apresentado no Gráfico 2. Nele podemos observar a apresentação das seguintes colunas: (1) conjunto dos identificadores dos documentos pertencentes ao conjunto dos relevantes, definido previamente por especialistas; (2)

identificação da posição do documento no resultado apresentado, na forma numérica sequencial; (3) o identificador do documento recuperado, sendo marcados aqueles que pertencem ao conjunto dos relevantes; (4) cálculo da medida de cobertura; (5) cálculo da medida de precisão. As duas últimas colunas apresentam o nível de cobertura entre 0 e 100% (6) e a precisão interpolada calculada (7). O cálculo de cobertura e precisão é realizado conforme demonstra a Figura 23.

Figura 23 – Cálculo de cobertura e precisão interpolada

Ranking	ID documento	Relevante
1	588	X
2	589	X
3	576	
4	590	X
5	986	
6	592	X
7	984	
8	988	
9	578	
10	985	
11	103	
12	591	
13	772	X
14	990	

Seja o número de documentos relevantes igual a 6. Calcula-se a cobertura e precisão a cada documento casado com os relevantes

COBERTURA: $1/6 = 0,167$
PRECISÃO: $1/1 = 1$

COBERTURA: $2/6 = 0,333$
PRECISÃO: $2/2 = 1$

COBERTURA: $3/6 = 0,5$
PRECISÃO: $3/4 = 0,75$

COBERTURA: $4/6 = 0,667$
PRECISÃO: $3/4 = 0,667$

Caso seja perdido pelo menos um documento relevante na recuperação, nunca será atingido 100% de cobertura.

COBERTURA: $5/6 = 0,833$
PRECISÃO: $5/13 = 0,38$

Fonte – Adaptada de Bernardi (2012, p. 18).

Segundo Bernardi (2012, p. 25) ainda é possível com os dados obtidos pelo Quadro 6 se fazer o cálculo médio da precisão interpolada, bastando para tanto se realizar a razão da soma das precisões interpoladas (ver coluna 7 do Quadro 6) por 11, número de níveis de cobertura. De acordo com a autora, este é o valor da precisão média de 11 pontos.

Usando um *software* de manipulação de planilhas eletrônicas, percebe-se na Figura 23 que o cálculo da cobertura é realizado dividindo-se a quantidade de documentos relevantes até a linha analisada pelo número total de documentos relevantes da coleção. O cálculo da precisão, por sua vez, é a razão entre a quantidade de documentos relevantes até a linha analisada pela posição da linha analisada no *ranking* (coluna *Ranking*). A Figura 23 ainda demonstra que, caso algum documento relevante não esteja entre os resultados obtidos, a cobertura nunca alcançará o valor de 100%.

3 MATERIAIS E MÉTODOS

Dentre as funcionalidades que compõem um GED, a principal delas podemos destacar como a possibilidade de recuperação de informações em um *corpus* de documentos digitais. Por ser um estudo não apenas teórico, o presente trabalho aborda algumas das características tecnológicas fundamentais de GED, dentre as quais podemos destacar: DI, COLD/ERM e *Forms Processing* (ver seção 2.1). Tais funcionalidades demandam a explanação de inúmeras tecnologias e métodos abordados nesta seção e necessários previamente à correta utilização de um SRI.

Esta seção provê uma visão geral das atividades desempenhadas para a construção de um SRI com implementação dos três modelos clássicos de Recuperação da Informação. Serão detalhados os equipamentos, as tecnologias e as linguagens utilizadas, além dos passos para a construção de um índice invertido juntamente com a coleta dos arquivos que compuseram o *corpus* base para a análise experimental realizada.

3.1 Plataforma de *hardware* e *software*

O ambiente de desenvolvimento e análise experimental de todo este estudo consistiu no uso das plataformas de *hardware* e *software* apresentadas no Quadro 7.

Quadro 7 – Materiais e métodos para o desenvolvimento do estudo

<i>HARDWARE</i>	
Computador PC, tipo <i>notebook</i>	
Processador Core I7-4500U 8 núcleos @1.80GHz - @2.40GHz	
8GB memória RAM DDR3	
1TB de disco rígido 7200RPM	
<i>SOFTWARE</i>	
ITEM	DESCRIÇÃO
Sistema Operacional Microsoft Windows 8.1 <i>Single Language</i> x64	Sistema Operacional comercial oferecido pela aliança acadêmica MSDN criada entre IFMG e Microsoft. Mais informações: https://www.microsoft.com/pt-br/windows .

Servidor <i>web</i> Apache versão 2.4.9	projeto de servidor <i>HyperText Transfer Protocol</i> (HTTP) <i>open source</i> desenvolvido e mantido pela fundação Apache. Mais informações: https://httpd.apache.org/ .
SGBD MySQL versão 5.6.17 <i>Community Server Edition</i>	sistema gerenciador de bancos de dados <i>open source</i> mais usado no mundo. Mais informações: http://dev.mysql.com .
Serviço OCR <i>online</i> ABBYY Cloud OCR SDK	serviço OCR de uso <i>online</i> via <i>webservice</i> ¹² com licença estudantil para uso de até 5000 páginas digitalizadas/mês. Mais informações: http://ocrsdk.com/ .
OpenShift by Red Hat	<i>Plataform-as-a-Service</i> (PaaS) mantido pela Red Hat que permite a desenvolvedores rapidamente desenvolver, hospedar, e escalar em um ambiente na nuvem. Mais informações: https://www.openshift.com/ .
Editor de textos Sublime Text <i>build</i> 3083	sofisticado editor de texto voltado para programadores. Mais informações: http://www.sublimetext.com/ .
Serviço de versionamento <i>online</i> de código-fonte CloudForge	hospedagem de código-fonte por versionamento via Subversion ou GIT. Mais informações: http://www.cloudforge.com/ .
Ferramenta de modelagem de Diagrama de Entidade-Relacionamento (DER) MySQL Workbench versão 6.3.5 <i>build</i> 201 CE (64 bits)	ferramenta <i>front-end</i> para modelagem de DER e uso do SGBD MySQL. Mais informações: http://www.mysql.com/products/workbench/ .
Ferramenta de modelagem de diagrama de classes <i>Unified Modeling Language</i> (UML) Microsoft Office Visio 2010 <i>Premium</i>	aplicativo comercial de criação e desenvolvimento de diagramas oferecido pela aliança acadêmica MSDN IFMG - Microsoft. Mais informações: https://products.office.com/pt-br/visio/flowchart-software .
TECNOLOGIAS WEB DE DESENVOLVIMENTO	
Linguagem de desenvolvimento <i>server-side</i> : PHP versão 5.5.12	linguagem de programação <i>server-side</i> com base em pré-processamento de <i>scripts</i> . Mais informações: https://www.php.net/ .
<i>Framework</i> de desenvolvimento PHP: Laravel versão 4.2	poderoso <i>framework</i> de desenvolvimento PHP focado em manter os últimos recursos da linguagem . Mais informações: https://laravel.com/ .
<i>HyperText Markup Language</i> 5 (HTML5)	linguagem de marcação padrão de desenvolvimento <i>client-side</i> para <i>web</i> . Mais informações: https://www.w3.org/TR/html5/ .
<i>Cascading Style Sheets</i> 3 (CSS3)	mecanismo simples de estilização de documentos web. Mais informações: https://www.w3.org/Style/CSS/ .
JQuery versão 2.1.3	biblioteca JavaScript para construção de aplicações ricas na <i>web</i> . Mais informações: https://jquery.com/ .
<i>Asynchronous JavaScript and XML</i> (AJAX)	uso metodológico de tecnologias como JavaScript e XML providas por navegadores, para tornar a navegação de páginas <i>web</i> mais interativas, por meio de requisições assíncronas ao servidor. Mais informações: http://www.w3schools.com/Ajax/ajax_intro.asp .

¹² Webservice: método que pode ser acessado por outros programas utilizando a infraestrutura da *web*. (W3C, 2001)

<i>Framework</i> de desenvolvimento <i>front-end</i> Bootstrap versão 3.3.6	<i>framework</i> HTML, CSS e JavaScript gratuito de desenvolvimento responsivo, <i>mobile first</i> para a <i>web</i> . Mais informações: http://getbootstrap.com/ .
<i>Template bootstrap Admin LTE Control Panel</i> versão 2.3.2	modelo de interface de painel de controle baseado em Bootstrap. Mais informações em: https://almsaeedstudio.com/ .
<i>FontAwesome</i> 3.2.1	Coleção icônica de caracteres vetoriais de uso livre para desenvolvimento de aplicações responsivas na <i>web</i> . Mais informações: https://fontawesome.github.io/Font-Awesome/ .
<i>JavaScript Object Notation</i> (JSON)	notação em formato JavaScript para a troca de objetos entre máquinas na <i>web</i> . Mais informações em: http://www.json.org/ .

Fonte – Elaborado pelo autor.

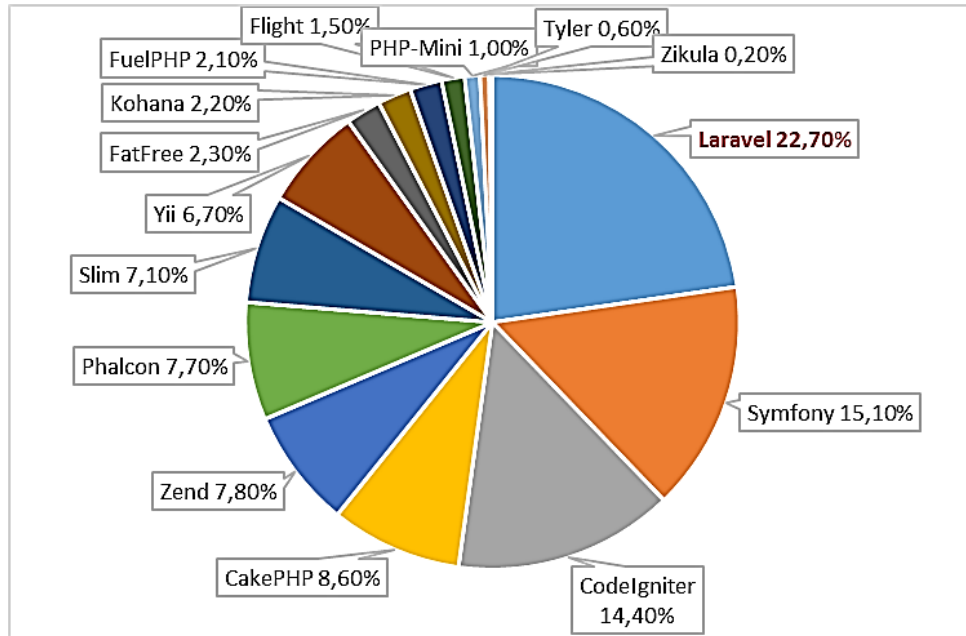
3.1.1 *Framework Laravel*

De acordo com Saunier (2014, p. 7-8), o uso de um arcabouço de técnicas, funcionalidades e tecnologias para o desenvolvimento de aplicações, independente da linguagem de programação adotada, que retire a preocupação do desenvolvedor da parte de infraestrutura está entre os conceitos mais bem aceitos pela comunidade de TI. Ainda segundo o autor, tais ferramentas são denominadas *frameworks* de desenvolvimento e auxiliam muito na agilização do processo de desenvolvimento de forma organizada, garantindo o uso de padrões de projeto e boas práticas apenas pelo seu uso.

O *framework* escolhido para este estudo é denominado Laravel, arcabouço de desenvolvimento *web* criado por Taylor Otwell baseado no padrão de projeto *Model, View, Controller* (MVC) e distribuído livremente. O MVC é um padrão de arquitetura de projeto de *software*, do inglês *design pattern*, que divide a aplicação em três camadas: o **modelo**, que consiste nos dados da aplicação, regras de negócio, lógica e funções; o **controle**, que realiza a mediação da entrada pelo usuário, convertendo-a em comando para a visão ou o modelo; e **visão**: que representa a saída de dados ao usuário (REENSKAUG; COPLIEN, 2009).

O Laravel foi criado para maximizar a qualidade de *software*, reduzindo tanto o custo inicial de desenvolvimento quanto de manutenção durante sua vida útil. Provê uma sintaxe limpa e um conjunto chave de funcionalidades que evitam horas de desenvolvimento repetitivo e redundante (MCCOOL, 2012, p. 3).

Gráfico 3 – Os mais populares frameworks PHP até fevereiro de 2015, segundo o GitHub



Fonte – Fauzi (2015).

A razão da escolha de tal *framework* está em uma pesquisa feita anteriormente a este trabalho sobre as melhores opções gratuitas do mercado, como podemos ver no Gráfico 3. De acordo com Saunier (2014, p. 11), as principais características oferecidas pelo Laravel e utilizadas no desenvolvimento deste trabalho são:

- **modularidade:** permite a customização do ambiente e bibliotecas do *framework* da forma que melhor convier ao desenvolvedor. Utiliza *Composer Dependency Manager*¹³ para gerenciamento de dependências e pacotes de forma ágil e facilitada.
- **ambiente de testes:** oferece recursos para testes do ambiente de desenvolvimento e produção, agilizando assim o processo de criação e entrega do produto.
- **roteamento:** oferece flexibilidade e segurança no acesso à aplicação por meio de verbos HTTP, como GET, POST, PUT e DELETE associados a rotas de destino, fazendo com que nada que não seja previamente autorizado possa ser acessado. Inspirado no Sinatra (Ruby).
- **ORM Eloquent:** apresenta nativamente integrado o *Object Relational Mapper* (ORM) chamado Eloquent, que permite acesso objeto-relacional ao dados da aplicação,

¹³ Mais informações sobre o Composer, gerenciador de dependências para PHP, em: <https://getcomposer.org/>.

abstraindo do desenvolvedor o acesso em baixo nível ao banco de dados e suas particularidades.

- ***query builder***: construtor de consultas com linguagem próximo à do PHP, facilitando o uso de consultas complexas sem o uso do ORM Eloquent.
- ***schema builder***: construtor do banco de dados diretamente pelo código PHP. Permite a construção e versionamento da aplicação sem interação alguma diretamente com o SGBD. Inspirado em Ruby on Rails.
- ***migrations***: junto ao uso do *schema builder*, permite o controle de versões do banco de dados para trabalho em equipe.
- ***seeding***: para efeitos de teste e alimentação inicial da aplicação em produção, a funcionalidade de *seeding* permite preencher o banco de dados em tempo de desenvolvimento e/ou produção.
- ***blade template engine***: inspirado em Razor para ASP, Blade é uma linguagem de modelos leve para criação hierárquica de blocos de *layouts* predefinidos com uso de injeção dinâmica de conteúdo. Esta funcionalidade garante agilidade e simplicidade na geração das *views* da aplicação.
- **autenticação**: oferece um padrão seguro e eficaz para o processo de autenticação de usuários na aplicação.

São também funcionalidades de destaque oferecidas pelo Laravel e não utilizadas neste trabalho: ambientes configuráveis, recursos de *e-mailing*, Redis e processamento de filas de atividades em lote. Mais informações sobre o Laravel podem ser adquiridas diretamente em sua documentação oficial, disponível em: <https://laravel.com/docs/4.2>.

3.2 Máquinas de busca

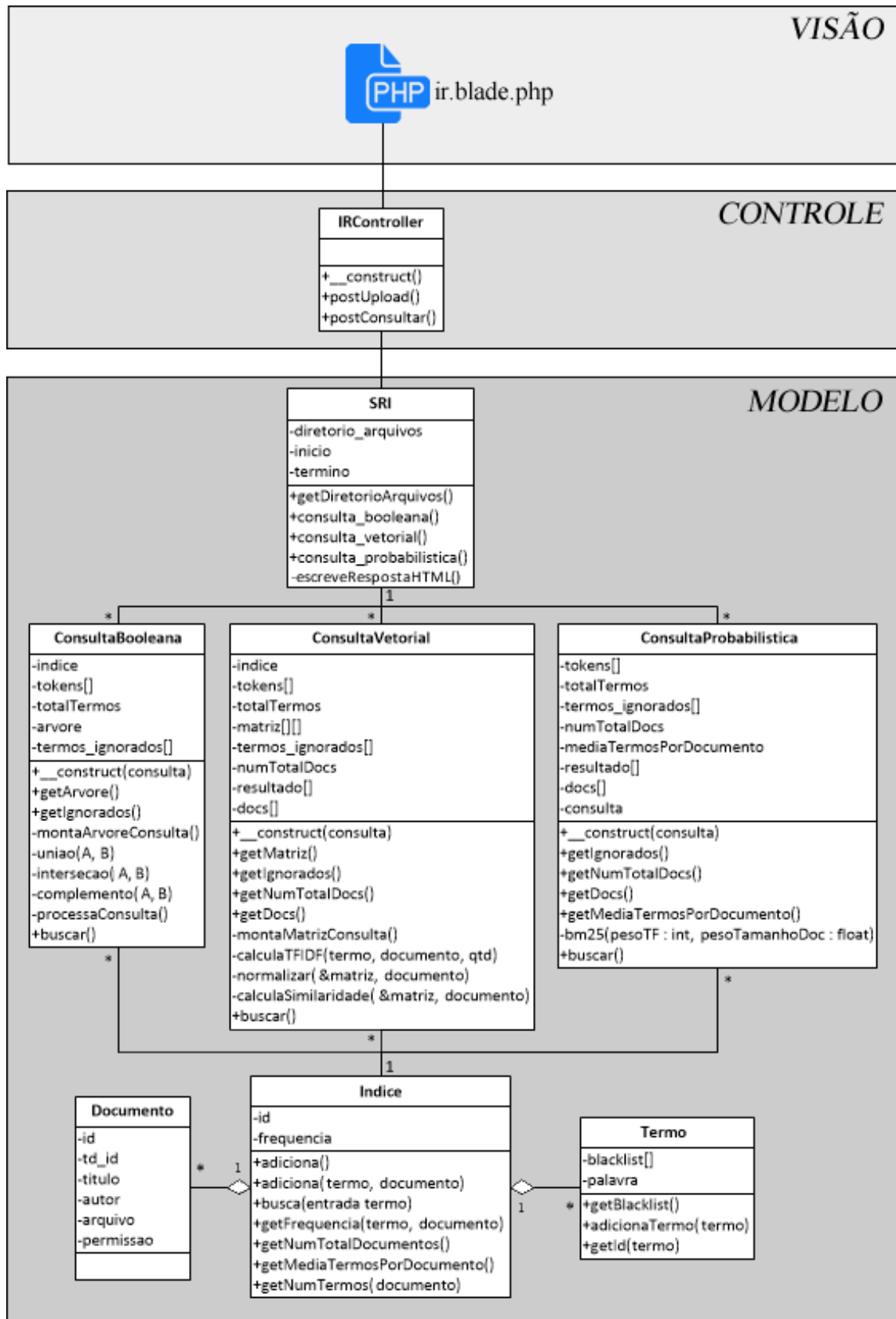
Vários conceitos estão envolvidos na construção de máquinas de busca e Recuperação da Informação em documentos digitais. Esta seção apresenta sistematicamente os conceitos aplicado à construção do SRI proposto neste trabalho, utilizando como subseções a própria divisão sugerida pelo uso do *framework* Laravel, o MVC.

3.2.1 Diagrama de classes

Segundo Fowler e Scott (1999, p. 58), o diagrama de classes se tornou uma grande verdade central no desenvolvimento por orientação a objetos. Apesar de utilizado e adotado largamente, seus elementos básicos podem suprir grande parte dos modelos conceituais. Existem variantes e extensões ao modelo proposto inicialmente por Booch, Jacobson e Rumbaugh entre 1994 e 1995 (BOOCH; RUMBAUGH; JACOBSON, 2005).

A Figura 24 apresenta o diagrama de classes UML modelado e implementado para o Sistema de Recuperação da Informação proposto, contemplando a opção ao usuário por um dos três modelos clássicos de Recuperação da Informação.

Figura 24 – Adaptação do diagrama de classes UML modelado



Fonte – Elaborada pelo autor.

Percebe-se pela Figura 24 que o modelo de classes UML implementado não utiliza recursos avançados da linguagem, pois sua lógica é simples e grande parte do modelo consiste na camada de modelo. Representado como o documento *ir.blade.php*, apenas um elemento simbólico é usado como visão, que nada mais é que a interface entre o usuário do sistema e o SRI implementado.

Pela interação com esta interface, o usuário dispara métodos da classe de controle *IRController* utilizando requisições AJAX, a qual é responsável por encaminhar sua requisição à camada de modelo para efetivamente realizar a consulta desejada. Neste passo, após definido o modelo de pesquisa, Booleano, Vetorial ou Probabilístico, o SRI executa a busca segundo o comportamento de cada uma destas implementações, sendo que todos utilizam em comum o índice previamente indexado do SRI, com 200 artigos em PDF publicados em periódicos científicos da área de computação e afins. O índice é composto por uma classe que intermedeia a relação entre termos e documentos, como se pode ver na Figura 24.

3.2.2 Camada de Visão

Conforme anteriormente abordado, o documento que representa a interface entre o usuário e o SRI implementado é tão somente o *ir.blade.php*, cujo conteúdo é apresentado pelo Código-fonte 1.

Código-fonte 1 – Conteúdo da camada de visão, representado pelo documento *ir.blade.php*

```

1  @extends('layout.main')
2
3  @section('cabecalho_html') @parent @stop
4  @section('cabecalho_pagina') @parent @stop
5  @section('barra_lateral') @parent @stop
6  @section('titulo') SRI @stop
7  @section('subtitulo') (Sistema de Recuperação da Informação) @stop
8  @section('breadcrumb') @parent @stop
9
10 {{-- CONTEÚDO PRINCIPAL DA PÁGINA --}}
11 @section('conteudo')
12     <div class="box box-primary">
13         <div class="box-header">
14             <h3 class="box-title"><i class="fa fa-search"></i> Buscar arquivos</h3>
15         </div><!-- /.box-header -->
16

```

```

17 <div class="box-body">
18 <div class="row margin">
19 <div class="col-md-8 col-md-offset-2">
20 <div class="input-group">
21 <input type="text" class="form-control" placeholder="Pesquisar..." id="edt_consulta">
22 <span class="input-group-btn">
23 <button class="btn btn-primary" type="button" id="btn_pesquisar">
24 <i class="fa fa-search"></i></button>
25 </span>
26 </div><!-- /input-group -->
27 </div><!-- /.col-md-8 -->
28 </div>
29 <div class="row margin">
30 <div class="form-group">
31 <div class="col-lg-4 col-xs-12 col-lg-offset-5">
32 <div class="radio">
33 <label>
34 <input type="radio" name="radio" id="radioBooleano" value="booleano"
35 checked="checked">
36 Modelo Booleano
37 </label>
38 </div>
39 <div class="radio">
40 <label>
41 <input type="radio" name="radio" id="radioVetorial" value="vetorial">
42 Modelo Vetorial
43 </label>
44 </div>
45 <div class="radio">
46 <label>
47 <input type="radio" name="radio" id="radioProbabilistico" value="probabilistico">
48 Modelo Probabilístico
49 </label>
50 </div>
51 </div>
52 </div>
53 </div>
54 </div><!-- /.box-body -->
55 </div><!-- /.box -->
56
57 <div class="box box-default">
58 <div class="box-header">
59 <h3 class="box-title">
60 <i class="fa fa-search"></i>
61 Resultados da busca por <span id="titulo_termo_busca">""</span>
62 </h3>
63 <div class="box-tools pull-right">
64 <span class="label label-default" id="total_resultados">Nenhuma consulta realizada</span>
65 <span class="label label-success" id="tempo_consulta"></span>
66 <button class="btn btn-box-tool" data-widget="collapse"><i class="fa fa-minus"></i></button>
67 </div>
68 </div><!-- /.box-header -->
69 <div class="box-body">
70 <div class="table-responsive">
71 <table class="table no-margin table-striped">

```

```

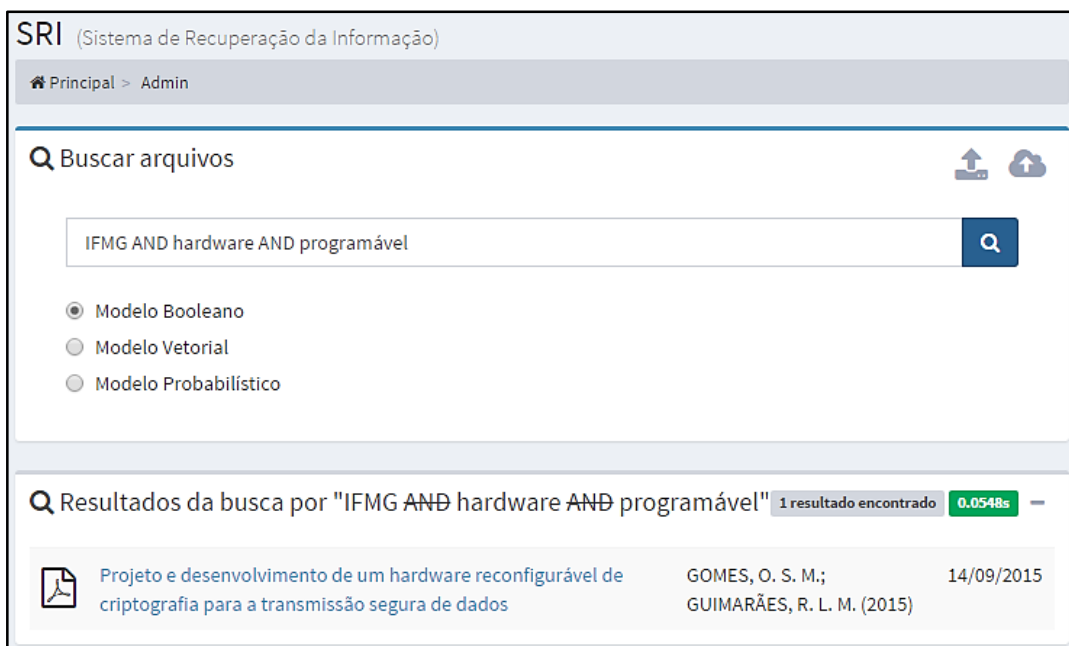
72     <tbody id="conteudo_resultado">
73         <!--AQUI VÃO OS RESULTADOS DA BUSCA -->
74     </tbody>
75 </table>
76 </div>
77 </div>
78 </div>
79 @stop
80
81 @section('rodape_pagina') @parent @stop
82 @section('rodape_html') @parent
83     {{ HTML::script('js/ir.js') }}
84 @stop

```

Fonte – Elaborado pelo autor.

No Código-fonte 1 podemos perceber o uso da hierarquia de extensões de *layout* proporcionada pelo uso da funcionalidade nativa do Laravel Blade. Por meio da anotação **@section** (linha 3, por exemplo) podemos estender conteúdo definido de *layout* em documentos externos, tornando assim o desenvolvimento de conteúdo modularizado. Para as *@sections* sem conteúdo, simplesmente a *view* atual não adiciona conteúdo extra ao definido na classe *layout.main* (linha 1) e carregado no documento *ir.blade.php* pela anotação **@parent** (linha 3, por exemplo). A Figura 25 apresenta o resultado da renderização pelo navegador *web* do documento da camada de visão *ir.blade.php*.

Figura 25 – Interface do usuário com o SRI: resultado da renderização de *ir.blade.php* pelo navegador *web*



Fonte – Elaborada pelo autor.

Como se observa pela Figura 25, as classes e estilos visuais renderizados são resultado da aplicação adaptada de Bootstrap¹⁴, *Admin LTE Control Panel Template*¹⁵ e de arquivos CSS próprios da aplicação.

De acordo com o Código-fonte 1 e a Figura 25, em sua seção conteúdo `@section('conteudo')`, linha 11, o arquivo *ir.blade.php* é composto por dois grandes blocos, denominados *box*, sendo o primeiro destes para o campo de pesquisa junto às opções de escolha do método de consulta e, o segundo, para exibição do resultado do método de classificação/ordenação. Todos os outros elementos em torno destes dois são elementos estruturais para formar o conteúdo de apresentação, ou seja, sua visão ao usuário. Exemplo disto é o `` (linha 65), responsável pela exibição do tempo decorrido após uma consulta realizada.

Ao final do Código-fonte 1 podemos perceber a inclusão na `@section('rodape_html')` do arquivo de JavaScript (linha 82) responsável por toda a interação do usuário na *view ir.blade.php*, cujo código fonte é apresentado no Código-fonte 2.

Código-fonte 2 – JavaScript e chamadas AJAX ao SRI: arquivo ir.js

```

1  $(function () {
2      // ao pressionar o botão pesquisar
3      $('#btn_pesquisar').on('click', function() {
4
5          //captura a opção de consulta selecionada
6          var metodo = $("input[name='radio']:checked").val();
7          //captura a expressão de busca inserida
8          var consulta = $('#edt_consulta').val();
9          // elemento onde é apresentado o título da busca realizado
10         var titulo_termo_busca = $('#titulo_termo_busca');
11         // elemento onde é apresentado o tempo da consulta
12         var tempo = $('#tempo_consulta');
13
14         // dispara requisição AJAX ao servidor
15         $.ajax({
16             url    : '/admin/ir/consultar',
17             type   : 'post',
18             async  : true,
19             data   : {'metodo': metodo,
20                     consulta: consulta},
21             dataType : 'json',
22             beforeSend: function(){
23                 titulo_termo_busca.html(''+consulta+'');

```

¹⁴ Bootstrap é o *framework* de desenvolvimento *front-end web* mais popular do mundo para criação de conteúdo responsivo e adaptável a dispositivos móveis. Mais informações podem ser adquiridas em: <http://getbootstrap.com/>.

¹⁵ Mais informações sobre o *template* Bootstrap utilizado em: <https://almsaeedstudio.com/>.

```

24     exhibeLoader('#conteudo_resultado', 'top');
25     },
26     success : function(resposta) {
27         console.log(resposta); // exibe no console o resultado obtido para DEBUG
28
29         //alimenta titulo da pesquisa
30         titulo_termo_busca.html(alimentaTituloPesquisa(resposta.ignorados, consulta));
31         // alimenta resultado da pesquisa
32         $('#conteudo_resultado').html(resposta.dados);
33
34         // se o resultado for igual a 0 documentos
35         if (resposta.total == 0) {
36             $('#conteudo_resultado').html("");
37             $('#total_resultados').html("Nenhum resultado encontrado");
38             tempo.html("");
39         } else if (resposta.total == 1) { // controle par ao caso de apenas 1 resultado (plural/singular)
40             $('#total_resultados').html("1 resultado encontrado");
41             tempo.html(resposta.tempo.toFixed(4)+'s');
42         } else {
43             $('#total_resultados').html(resposta.total+' resultados encontrados');
44             tempo.html(resposta.tempo.toFixed(4)+'s');
45         }
46     },
47     error : function(erro){
48         console.log(erro);
49     },
50     complete : function() {
51         ocultaLoader();
52     }
53 });
54 });
55
56 // método para tachar termos ignorados da pesquisa
57 function alimentaTituloPesquisa(ignorados, consulta) {
58     if (consulta.length == 0) {
59         return "";
60     }
61
62     var i;
63     var resp = "";
64     consulta = consulta.split(" ");
65     for (i = 0; i < consulta.length; ++i) {
66         // se não encontrar o termo na lista de ignorados, so adiciona
67         if (ignorados.indexOf(consulta[i].toLowerCase()) == -1) {
68             if (i == consulta.length-1) {
69                 resp += consulta[i]+'';
70             } else {
71                 resp += consulta[i]+' ';
72             }
73         } else { // se encontrar adiciona tachado
74             if (i == consulta.length-1) {
75                 resp += '<span style="text-decoration: line-through;">'+consulta[i]+'</span>';
76             } else {
77                 resp += '<span style="text-decoration: line-through;">'+consulta[i]+'</span> ';
78             }

```

```

79     }
80     }
81     return resp;
82     }
83
84     // adiciona o manipulador de eventos ao campo de consulta, para o caso de se pressionar ENTER
85     $("#edt_consulta").keypress(function(event) {
86         if (event.which == 13 ) {
87             event.preventDefault();
88             $('#btn_pesquisar').click();
89         }
90     });
91 });

```

Fonte – Elaborado pelo autor.

Inicialmente o *script* apresentado no Código-fonte 2 apresenta a função anônima¹⁶ $\$(function ()$ que é primordial no uso da biblioteca JQuery em JavaScript. Em JQuery essa função significa executar o *script* quando o documento estiver carregado, ou seja, utiliza o *status* de documento *onload*, nativo do JavaScript, para executar o *script* somente após o navegador ter carregado todo o conteúdo HTML a ser renderizado (DEERING, 2011).

Assim, o Código-fonte 2 inicia pela atribuição de um manipulador de eventos ao botão de consulta da interface $\$('#btn_pesquisar').on('click')$, linha 3. Pelo uso de uma função anônima, ao se clicar neste botão os seguintes passos são executados: (1) captura de elementos e valores do *Document Object Model* (DOM) (FRANKLIN, 2011) que são chave para o envio e recebimento de informações entre as camadas de visão e modelo, como por exemplo a expressão de busca e o elemento onde será apresentado o tempo de consulta (linhas 6 a 12); e (2) dispara uma requisição AJAX pelo método $\$.ajax()$ da biblioteca JQuery, requisitando assim ao servidor que inicialize o método de busca selecionado e que alimente a página atual de forma assíncrona e sem a necessidade de se recarregar todo o seu conteúdo, apenas são adicionados novos elementos ao DOM quando o processo de busca é terminado no servidor (linha 15).

Uma requisição AJAX em JQuery é constituída dos seguintes parâmetros: *url* define qual será a rota¹⁷ destino da requisição; *type* seleciona qual o verbo HTTP a ser utilizado, como POST, GET, PUT, DELETE; *data* representa os dados passados para o servidor, que no caso do Código-fonte 2 é o método escolhido para a busca e a expressão de busca inserida pelo

¹⁶ Mais informações sobre funções anônimas no JQuery: <http://jquerybrasil.org/como-funciona-o-jquery/>

¹⁷ Laravel trabalha com o conceito de rotas para definir quais URL podem ser acessadas na aplicação. Mais informações em: <https://laravel.com/docs/4.2/routing>

usuário; *dataType* é o tipo de dados da comunicação de dados de envio/recebimento. Existem funções definidas dentro da função *\$.ajax()*, a saber: a) *beforeSend()*, b) *success()*, c) *error()* e d) *complete()* que respectivamente são responsáveis por: a) executar ações antes da requisição ser realizada, como é o caso de se exibir alguma animação de progresso; b) o que fazer ao se receber a requisição com sucesso; c) o que fazer em caso de erro na requisição, por exemplo caso o servidor esteja inacessível e, por fim; d) qual atitude tomar ao se completar uma requisição, que neste optou-se apenas por ocultar a animação de progresso utilizada. Mais detalhes sobre requisições AJAX com JQuery e todas as suas parametrizações podem ser consultadas diretamente na documentação oficial, disponível em: <http://api.jquery.com/jquery.ajax/>.

O interessante para este estudo é o resultado obtido no caso de sucesso da requisição AJAX (linha 26). Nesta situação optou-se primeiramente por lançar todo o pacote de resposta no console do navegador apenas como forma de auxílio no desenvolvimento. Logo após isso, simplesmente é alimentada a expressão de busca no título do segundo bloco da *view*, bem como seus resultados classificados e ordenados são inseridos no elemento do DOM de identificador **conteudo_resultado**, definido no *script* como *\$('#conteudo_resultado')* (Código-fonte 1, linha 74).

A estrutura da resposta do servidor em formato JSON é definida de acordo com o Quadro 8.

Quadro 8 – Estruturada resposta do servidor à camada de visão do SRI

```
Object {
  dados : <string de resultado em HTML da função de classificação/ordenação do SRI>,
  erro : <caso ocorra algum erro do lado servidor, armazena a mensagem de erro>,
  ids : <armazena os números identificadores do resultado obtido, separado por vírgulas>,
  ignorados : <vetor de termos ignorados da expressão de busca passada>,
  tempo : <tempo em segundos da execução da consulta no servidor>,
  total : <número total de documentos recuperado>
}
```

Fonte – Elaborado pelo autor.

Ao final do Código-fonte 2 podemos observar duas funções auxiliares, sendo a primeira delas responsável pela alimentação do título do resultado no documento HTML, garantindo o estilo de fonte tachado para os elementos ignorados da consulta e, a segunda função, que apenas atribui um manipulador de eventos à ação de pressionar a tecla ENTER *\$('#edt_consulta').keypress()* (linha 85). Logo, estando o usuário com o foco no campo de

pesquisa, esta função garante que ao se pressionar ENTER, o botão de consulta terá seu método associado disparado, oferecendo assim usabilidade ao protótipo implementado.

Toda a interação entre a camada de visão, representando justamente a interação do usuário com o sistema, não interage diretamente com o modelo do negócio. Essa comunicação ocorre com um intermediador, a camada de controle, que é abordada na próxima seção.

3.2.3 Camada de Controle

Apesar de ter algumas responsabilidades internas à segurança e navegabilidade do sistema em Laravel, como alimentação de elementos padrão da visão, a camada de controle deste estudo definida pela classe **IRController** se resume a um método denominado **consultar()**, como demonstra o Código-fonte 3. Observação: no Laravel, os métodos públicos de um controlador são inicializados pelo seu verbo HTTP¹⁸ de chamada, ficando assim o método consultar como: **postConsultar()**. Dessa forma, ao se disparar uma requisição para **<url do controlador>/consultar**, como exemplificado pela requisição AJAX do Código-fonte 2, o método consultar do controlador **IRController** será disparado.

Código-fonte 3 – O controlador IRController

```

1  public function postConsultar() {
2      $metodo = Input::get('metodo'); // captura o modelo selecionado de RI
3      $consulta = Input::get('consulta'); // captura a expressão de busca inserida pelo usuário
4
5      if (empty($consulta)) {
6          return Response::json(['total' => 0]); //se expressão vazia, retorna resposta vazia
7      }
8
9      switch ($metodo) { // seleção do método de busca
10         case 'booleano':
11             $resultado = SRI::consulta_booleana($consulta);
12             break;
13         case 'vetorial':
14             $resultado = SRI::consulta_vetorial($consulta);
15             break;
16         case 'probabilistico':
17             $resultado = SRI::consulta_probabilistica($consulta);
18             break;

```

¹⁸ Mais informações sobre os métodos de controladores e seus verbos HTTP em: <https://laravel.com/docs/4.2/controllers>.

```

19     }
20     return Response::json($resultado); // retorno da resposta em formato JSON
21 }

```

Fonte – Elaborado pelo autor

O Código-fonte 3 realiza a mediação entre a camada de modelo e a camada de visão, justamente a função de um controlador definida no modelo MVC (REENSKAUGH; COPIEN, 2009). Nela observamos que basicamente o controlador recupera o modelo selecionado de Recuperação da Informação e a expressão de busca inserida pelo usuário e encaminha ao devido mecanismo de busca, seja ele o Booleano, o Vetorial ou o Probabilístico. Antes de tudo, existe a validação de uma consulta realizada sem expressão de busca (linha 5), o que faz com que o controlador retorne uma resposta vazia.

Ao fim do Código-fonte 3, a tarefa do controlador é retornar os dados em formato legível à camada de visão, neste trabalho optado pelo formato JSON. Utilizando o Laravel, o simples método estático *json()*, da classe *Response* se encarrega da formatação necessária nesta resposta.

Uma outra pequena classe de controle ainda existe no protótipo implementado, a classe de rotas da aplicação. Apesar de ser inerentemente uma classe de uso do Laravel, a qual se responsabiliza pela segurança de rotas acessíveis na aplicação, um pequeno método definido como uma rota foi utilizado para permitir *downloads* de documentos PDF, conforme é apresentado pelo Código-fonte 4. Esta classe não foi documentada no diagrama de classes por não ter relação direta com o SRI, mas sim como a usabilidade da interface do usuário.

Código-fonte 4 – Rota de downloads de documentos PDF

```

1 Route::get('download/{filename}', function($filename) {
2     // verifica se o arquivo existe na pasta app/storage/files
3     $file_path = storage_path() . '/files/' . $filename;
4
5     if (file_exists($file_path)) {
6         return Response::download($file_path, $filename, ['Content-Length: ' . filesize($file_path)]);
7     } else {
8         exit('O arquivo requerido não existe no servidor!');
9     }
10 })->where('filename', '[A-Za-z0-9\-\_\.\.]+');

```

Fonte – Elaborado pelo autor.

O Código-fonte 4 demonstra a rota criada com o prefixo **/download/<nome do arquivo>** para permitir que os documentos PDF indexados pelo índice possam ser recuperados

na camada de visão. Assim, o Código-fonte 4 verifica a existência do arquivo antes de responder com uma resposta padrão HTTP para *download* de arquivos em navegadores *web*, tarefa realizada pelo método estático *download()* da classe **Response** do Laravel. Um detalhe da rota criada é que a mesma somente permite nomes de arquivos segundo a expressão regular `[A-Za-z0-9\-_\.\+]` define (linha 10), ou seja, nomes de documentos que contenham letras, números, hifens, sublinhados e pontos.

3.2.4 Camada de Modelo

Na camada de modelo se concentram todas as regras de negócio e lógica da aplicação. Nela podemos encontrar toda a estrutura dos modelos de Recuperação da Informação implementados, como as classes de ConsultaBooleana, ConsultaVetorial e ConsultaProbabilistica. Junto a elas, e de suma importância, está a implementação do índice invertido, cujo processo de construção e indexação necessariamente é prévio à utilização do SRI, seguido do processo de especificação de consulta para a utilização de fato do mecanismo de busca implementado. Todos estes assuntos são abordados nas seções a seguir.

3.2.4.1 O Sistema de Recuperação da Informação implementado

De acordo com a Figura 24, que apresenta o modelo adaptado do diagrama de classes, percebe-se que a camada de controle do protótipo desenvolvido não interage diretamente com cada modelo de RI implementado, mas sim com uma classe denominada **SRI**. Esta classe **SRI** é responsável por toda a manipulação de requisições ao protótipo de SRI construído, como demonstra o Código-fonte 5, que apresenta as principais funções da máquina de busca implementada.

Código-fonte 5 – Classe SRI

1	<code><?php</code>
2	
3	<code>class SRI {</code>

```

4  private static $diretorio_arquivos = '/files/';
5  private static $inicio;
6  private static $termino;
7
8  // getter para uso externo
9  public static function getDiretorioArquivos() {
10     return storage_path() . self::$diretorio_arquivos;
11 }
12
13 // método para realizar a consulta utilizando o modelo booleano
14 public static function consulta_booleana($consulta) {
15     self::$inicio = microtime(true); // captura o tempo de início da consulta
16
17     $resp = new ConsultaBooleana($consulta); // cria um objeto de consulta booleana
18     $res = $resp->buscar(); // dispara o método de consulta booleana
19     $resultado = self::escreveRespostaHTML($res, $resp);
20
21     self::$termino = microtime(true); // captura o tempo de término da consulta
22     $resultado['tempo'] = self::$termino - self::$inicio; // calcula o tempo total gasto na consulta
23
24     return $resultado;
25 }
26
27 // método para realizar a consulta utilizando o modelo vetorial
28 public static function consulta_vetorial($consulta) {
29     self::$inicio = microtime(true); // captura o tempo de início da consulta
30
31     $resp = new ConsultaVetorial($consulta); // cria um objeto de consulta vetorial
32     $res = $resp->buscar(); // dispara o método de consulta vetorial
33     $resultado = self::escreveRespostaHTML($res, $resp);
34
35     self::$termino = microtime(true); // captura o tempo de término da consulta
36     $resultado['tempo'] = self::$termino - self::$inicio; // calcula o tempo total gasto na consulta
37
38     return $resultado;
39 }
40
41 // método para realizar a consulta utilizando o modelo probabilístico
42 public static function consulta_probabilistica($consulta) {
43     self::$inicio = microtime(true);
44
45     $resp = new ConsultaProbabilistica($consulta); // cria um objeto de consulta probabilística
46     $res = $resp->buscar(); // dispara o método de consulta probabilística
47     $resultado = self::escreveRespostaHTML($res, $resp);
48
49     self::$termino = microtime(true); // captura o tempo de término da consulta
50     $resultado['tempo'] = self::$termino - self::$inicio; // calcula o tempo total gasto na consulta
51
52     return $resultado;
53 }
54
55 // método para escrever o resultado da consulta na formatação HTML necessária à camada de visão.
56 private static function escreveRespostaHTML(&$res, &$resp) {
57     if (sizeof($res)) {
58         $total = sizeof($res);

```

```

59     } else {
60         $total = 0;
61     }
62
63     $resultado['total'] = $total;
64     $resultado['erro'] = "";
65     $resultado['dados'] = "";
66     $resultado['ids'] = ""; // para uso na análise experimental
67     $resultado['ignorados'] = $resp->getIgnorados();
68
69     //escreve em HTML o resultado a ser inserido no DOM
70     foreach ($res as $r) {
71         try {
72             $doc = Documento::find($r);
73             $resultado['ids'] .= ','.$doc->id; // para uso na análise experimental
74             $resultado['dados'] .=
75                 '<tr>
76                     <td width="38" style="vertical-align: middle; text-align: center;">
77                         <i class="fa fa-file-pdf-o fa-2x"></i>
78                     </td>
79                     <td>
80                         <div class="titulo_resultado">
81                             <a href="/download/' . $doc->arquivo . "'>'. (($doc->titulo) ? $doc->titulo : ') . '</a>
82                         </div>
83                         <div class="descricao_resultado">'. (($doc->descricao) ? $doc->descricao : ') . '</div>
84                     </td>
85                     <td width="180">'. $doc->autor . '</td>
86                     <td width="40">'. (($doc->created_at) ? $doc->created_at->format('d/m/Y') : ') . '</td>
87                 </tr>';
88         } catch (Exception $e) {
89             $resultado['erro'] = $e;
90         }
91     }
92     return resultado;
93 }
94 }

```

Fonte – Elaborado pelo autor.

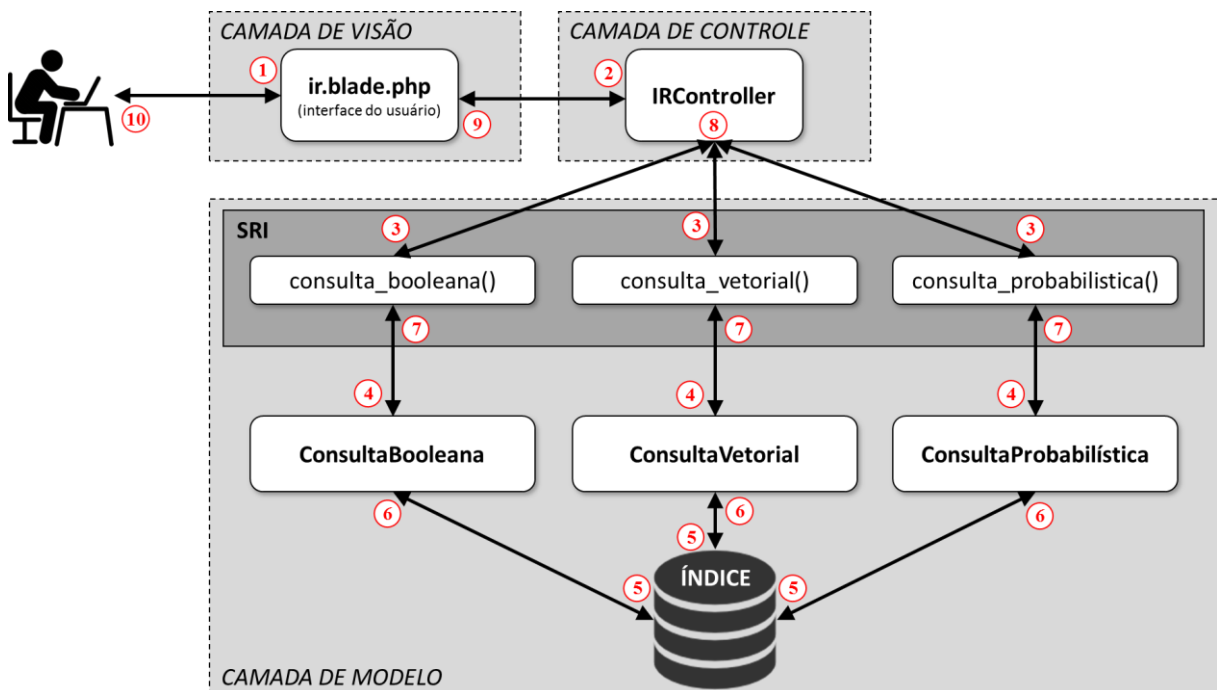
Analisando-se o Código-fonte 5, percebe-se que a classe **SRI** consiste em três atributos, sendo o primeiro para definição do diretório dentro da variável de ambiente **storage_path()** onde estão ou serão alocados os documentos PDF que compõem o *corpus* de busca do SRI. Os outros dois atributos apenas são marcadores de intervalo para cálculo do tempo dispendido pelos modelos de RI do sistema implementado.

Além do método *getter* do atributo diretório, o Código-fonte 5 possui três métodos relacionados aos modelos de RI e um último método responsável pela construção do HTML de resposta para a camada de visão, utilizado por cada um dos métodos de RI. Como abordado, tal resultado em HTML é passado primeiramente à camada de controle, que por sua vez o trata e encaminha à camada de visão.

Cada método, concernente a um modelo de RI presente no Código-fonte 5 tem, nessa classe, um comportamento similar. Todos os três métodos marcam o tempo de início e término pelo uso da função *microtime(true)*¹⁹, nativa do PHP. Em cada método ainda há a criação de um objeto da classe do modelo de RI selecionado pelo usuário e após construído tal objeto, é realizada sua função de busca. Por fim, os resultados obtidos do SRI são passados à função *escreveRespostaHTML()* (linha 56) para produção em HTML da formatação para o *layout* necessário à correta visualização na camada de visão da aplicação.

A Figura 26 apresenta um resumo demonstrando os passos seguidos por todo o SRI implementado, sendo necessário seguir a ordem numérica atribuídas aos passos para se entender seu processo.

Figura 26 – Resumo de todo o processo de busca definido para o protótipo do SRI implementado



Fonte – Elaborada pelo autor.

De acordo com a Figura 26, os seguintes passos são definidos:

1. O usuário insere a expressão de busca e escolhe o modelo de RI a ser usado pelo SRI por meio do uso da interface gráfica de usuário implementada. Utilizando uma

¹⁹ Segundo o manual do PHP o uso do valor *true* como atributo da função *microtime()* retorna um valor *float* formatado em segundos. Mais informações: http://php.net/manual/pt_BR/function.microtime.php.

requisição AJAX a camada de visão dispara assincronamente o método *consultar()* existente no controlador *IRController*, exibindo ao usuário uma animação indicativa de processamento (*loader*).

2. O controlador *IRController* verifica se a consulta é nula, cancelando assim o processo de consulta. Caso contrário são executados métodos estáticos da classe *SRI* respectivamente de acordo com o modelo de RI selecionado pelo usuário no passo 1.
3. Neste passo, cada método da classe *SRI* é responsável por instanciar a classe do modelo de RI alvo.
4. O construtor de cada classe do modelos definidos de RI possui as etapas necessárias à representação lógica de documentos e consulta, ficando a cargo da etapa 3 anterior disparar o método *buscar()* de tais classes.
5. As classes dos modelos de RI utilizam de forma frequente o acesso ao índice lexicográfico montado previamente à consulta. O uso da função *buscar()* citada na etapa anterior efetivamente executa o processo de recuperação do resultado o que é feito por um casamento exato de todos os documentos que contêm algum dos termos da expressão de busca.
6. Ao montar o resultado composto pelos documentos considerados relevantes pelo índice da aplicação e a função de recuperação do modelo de RI adotado, a classe de RI em questão utiliza sua função de similaridade juntos aos pesos associados às representações para criar a ordenação de apresentação dos documentos recuperados (*ranking*). Por fim, retorna à classe *SRI* uma lista ordenada de ID's de documentos recuperada.
7. Neste passo, a classe *SRI* monta toda a estrutura de informações necessária à apresentação dos resultados ao usuário, gerando por exemplo *hiperlinks* de acesso aos títulos dos documentos, informações de tempo da busca e formatação da expressão de busca com os termos ignorados pelo *blacklist*.
8. Todo o pacote de informações citado no passo anterior é retornado ao *IRController*, o qual reencaminha o pacote de resultados para a camada de visão formatando-o de acordo com a especificação JSON.
9. A camada de visão, por sua vez, atualiza os conteúdos gerados pelo servidor na tela de forma assíncrona, encerrando assim a exibição da animação definida no passo 1.
10. Por fim, o usuário tem uma lista de resultados considerada relevante à sua consulta pelo modelo de RI escolhido.

3.2.4.2 Construção do índice invertido

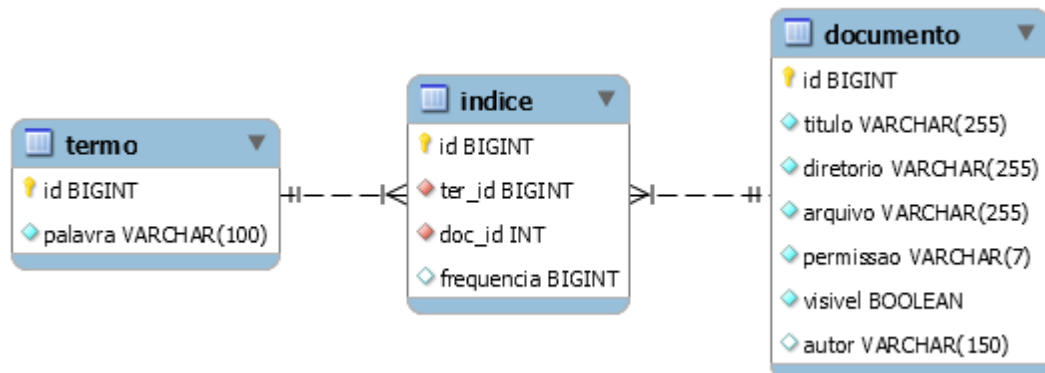
Parte integrante do modelo do SRI implementado, o processo de construção do índice invertido é fundamental para o seu funcionamento adequado, além de ser um quesito que rege a forma de trabalho dos modelos de RI. A Figura 24 o apresenta como parte ligada diretamente aos modelos clássicos de RI e esta seção aborda as particularidades de sua construção.

3.2.4.2.1 Diagrama de entidade-relacionamento

De acordo com Davis e Yen (1999), diagramas de entidade-relacionamento, conhecidos como DER, têm o propósito principal de modelagem ágil com o mínimo de custo, fornecendo assim uma boa ideia da estrutura de um banco de dados. São usados para planejar e desenvolver um banco de dados por meio de um modelo de persistência de dados, disponibilizando ao analista uma visão clara e de alto nível sobre os dados da aplicação e, se usados em conjunto com diagramas de fluxo da informação, possibilitando ao analista uma visão lógica de todo o sistema a ser planejado. Portanto, caso o interesse na etapa de desenvolvimento de *software* seja sobre a persistência, sua segurança e integridade, importando-se menos com o processo, um DER é um excelente ponto de partida para a modelagem de sistemas de informação.

Como a ideia proposta para este estudo é de um SRI modular, com baixa acoplagem a um Sistema de Informação, para os efeitos da implementação e análise experimental, seu DER simplificado pode ser visto na Figura 27.

Figura 27 – Diagrama de entidade-relacionamento do SRI implementado



Fonte – Elaborada pelo autor.

O cerne para a construção da etapa de persistência do índice consiste na relação N:N entre as entidades termo e documento, onde pode existir ainda o armazenamento da frequência de um termo no documento, citado anteriormente como *term frequency tf*. Utilizando o tipo *unsigned BIGINT* para os identificadores do índice, disponível na maioria dos SGBD, é possível se obter um valor máximo de 18.446.744.073.709.551.615²⁰ identificadores diferentes que, neste caso, seria o valor máximo de termos que poderiam ser indexados por este índice.

Para o propósito dos testes e análise experimental deste estudo optou-se por manter o autor do documento como um simples atributo textual, ciente de que em uma aplicação em produção isto seria contraproducente, o que de fato ocasionaria redundância desnecessária no banco de dados.

Outras estatísticas como o *document frequency df* e o *inverse document frequency idf* não podem ser armazenadas e manipuladas diretamente pela persistência da aplicação por razão de serem estatísticas diretamente relacionadas à consulta, e não ao termo/documento. Assim, seus valores devem ser calculados em tempo de execução da consulta, por meio da contagem de documentos relacionados ao termo, no caso do *df*, e pela Equação 3, para o caso do *idf*.

3.2.4.2.2 Persistência

Sendo de total liberdade ao desenvolvedor escolher alternativas de SGBDs como PostgreSQL, SQLServer, SQLite e OracleDB, o Laravel utiliza sua classe de *Schema Builder* junto ao recurso de *migrations* para a criação efetiva do banco de dados. Neste estudo e implementação se definiu como SGBD o MySQL.

Utilizando *migrations*, o Laravel permite ao desenvolvedor ou equipe de desenvolvimento o versionamento do banco de dados, garantindo assim o trabalho em equipe. Assim como no versionamento de código-fonte, onde cada integrante necessita atualizar seu código antes de iniciar um novo subprojeto, no Laravel essa necessidade é realizada pela atualização das *migrations*, permitindo a todos utilizar a última versão do banco de dados da aplicação (LARAVEL DOCUMENTATION, 2016) em desenvolvimento. O Código-fonte 6 apresenta a *migration* de criação do banco de dados definido pela Figura 27.

²⁰ tipo BIGINT segundo o Manual de Referência do MySQL possui no modo *unsigned* 18.446.744.073.709.551.615 como valor máximo. Mais informações: <https://dev.mysql.com/doc/refman/5.5/en/integer-types.html>

Código-fonte 6 – Exemplo de *migration* para criação do banco de dados do SRI

```

1  <?php
2
3  use Illuminate\Database\Schema\Blueprint;
4  use Illuminate\Database\Migrations\Migration;
5
6  class CreateDatabase extends Migration {
7
8      /**
9       * Run the migrations.
10      * @return void
11      */
12     public function up() {
13
14         Schema::create('documento', function(Blueprint $table) {
15             $table->bigIncrements('id');
16             $table->string('titulo', 400);
17             $table->string('autor', 255)->nullable();
18             $table->string('diretorio', 255);
19             $table->string('arquivo', 255);
20             $table->string('permissao', 7);
21             $table->boolean('visivel');
22             $table->string('autor', 150);
23         });
24
25         Schema::create('termo', function(Blueprint $table)
26         {
27             $table->bigIncrements('id');
28             $table->string('palavra', 100);
29         });
30
31         Schema::create('indice', function(Blueprint $table)
32         {
33             $table->bigIncrements('id');
34             $table->bigInteger('ter_id')->unsigned();
35             $table->foreign('ter_id')->references('id')->on('termo');
36             $table->integer('doc_id')->unsigned()->nullable();
37             $table->foreign('doc_id')->references('id')->on('documento');
38             $table->bigInteger('frequencia');
39         });
40     }
41
42     /**
43      * Reverse the migrations.
44      * @return void
45      */
46     public function down() {
47
48         Schema::drop('indice');
49         Schema::drop('termo');
50         Schema::drop('documento');
51     }
52 }

```

Pelo Código-fonte 6 percebemos que uma *migration* possui dois simples métodos: *up()* e *down()*, respectivamente responsáveis por criar algum conteúdo no banco e desfazer alguma alteração ou remover algo criado usando a função de *rollback* do Laravel.

3.2.4.2.3 Alimentação dos documentos para composição do corpus

Para alimentar os dados necessários aos testes e à análise experimental, neste estudo foram utilizados semeadores, do inglês *seeders*, recurso oferecido pelo Laravel e demonstrado pelo Código-fonte 7 a seguir.

Código-fonte 7 – Exemplo de *seeder* para alimentação dos documentos do SRI

```

1  <?php
2
3  class DocumentoSeeder extends Seeder {
4      /**
5       * Run the database seeds.
6       * @return void
7       */
8      public function run() {
9
10         $dados = [
11             ['Análise teórica da recuperação de calor para geração de energia em
12              indústrias de cimento e cal utilizando o ciclo de rankine orgânico', '', '93-686-
13              1-PB.pdf', '-rwrwrw', 1, 'CARPIO, R. C., <i>et al</i> (2015)'],
14             ['Importância dos aspectos socioculturais na gestão de equipes em ambientes
15              de desenvolvimento distribuído de software', '', '96-697-1-PB.pdf', '-rwrwrw',
16              1, 'ZUQUELLO, A. G., <i>et al</i> (2015)']
17         ];
18
19         foreach ($dados as $d) {
20             $doc = new Documento();
21
22             $doc->titulo = $d[0];
23             $doc->diretorio = $d[1];
24             $doc->arquivo = $d[2];
25             $doc->permissao = $d[3];
26             $doc->visivel = $d[4];
27             $doc->autor = $d[5];
28
29             $doc->save();
30         }
31     }
32 }

```

Observa-se que somente dois documentos foram inseridos no sistema, a título de exemplo, sendo que a lista completa de todos os artigos utilizados pode ser conferida no Apêndice A deste trabalho.

Este passo definiu a localização e armazenamento do documento final em formato PDF que é apresentado como resultado das buscas no sistema. Entretanto, para que os termos sejam utilizáveis na indexação do sistema, é necessário que se tenha o documento PDF convertido para formato de texto puro, conforme será descrito na próxima seção.

3.2.4.2.4 Reconhecimento de texto em arquivos PDF via OCR

A tecnologia OCR, denominada reconhecimento óptico de caracteres, converte documentos em formato digital para o formato de texto puro. Apesar de ser possível a obtenção de texto puro diretamente de arquivos PDF, nem todos estes permitem tal recuperação, sendo necessário o uso da tecnologia OCR. Outro fator motivador para o uso de serviços OCR é o fato de que documentos PDF têm duas origens: podem ser gerados digitalmente por aplicativos e *softwares* como editores de texto ou podem ser resultado da digitalização de documentos oficiais com assinaturas e símbolos feitos à mão. Neste último caso, o resultado é um documento PDF contendo imagens, o que impede algoritmos de coleta de texto puro de capturar seu conteúdo.

As duas alternativas levantadas para uso neste estudo foram: desenvolver um serviço OCR ou mesmo aplicar algum de uso livre, ou; utilizar serviços OCR gratuitos disponíveis na *web*, na forma de *webservices*. No primeiro caso foram estudadas algumas alternativas, sendo a mais promissora delas o TesseractOCR (SMITH, 2007). Segundo Smith (2007, p. 629), o Tesseract é um mecanismo *open-source* de OCR desenvolvido pela Hewlett Packard (HP) entre 1984 e 1994, se destacando desde então pela sua precisão no teste anual da *University of Nevada Las Vegas* (UNLV), em 1995. Originado de uma tese de PhD, a HP resolveu desenvolver a ideia para uso na sua divisão de *scanners* de mesa. Entretanto o Tesseract nunca foi utilizado pela HP em nenhum produto, devido as apostas da empresa em outro mecanismo, sendo hoje o Tesseract oferecido e mantido pela Google como *software* livre²¹ sob licença Apache²².

²¹ Mais informações sobre o Tesseract OCR: <https://github.com/tesseract-ocr>.

²² Mais informações sobre a licença Apache: <http://www.apache.org/licenses/LICENSE-2.0>.

A implementação de um serviço modular OCR, por si só, consiste em todo um estudo extra, originando até mesmo um futuro trabalho. Objetivando minimizar o tempo de projeto, neste trabalho optou-se pela segunda alternativa: o uso de um serviço gratuito disponibilizado como *webservice*. Para isso, foi necessária uma pesquisa dos serviços disponíveis de baixo custo no mercado, dentre eles alguns testados foram:

- ***freeOCR***: ferramenta *online* gratuita de OCR. Apesar de oferecer o reconhecimento de várias linguagens, inclusive o português, não oferece uma *Application Programming Interface* (API) para uso junto à implementação do SRI. Outra desvantagem deste serviço é que em documentos PDF somente a primeira página é convertida. Mais informações em <http://www.free-ocr.com/>.
- ***HP Haven OnDemand***: conjunto de APIs para ajudar gratuitamente desenvolvedores a construir aplicações no uso de dados não estruturados. Dentre as APIs oferecidas pelo *webservice*, destaca-se o IDOL OnDemand OCR Document API, o qual oferece o processamento de OCR devolvendo os dados em formato *JavaScript Object Notation* (JSON), junto à conversão de formatos, análise de imagem, busca, análise textual e suporte à criptografia por tráfego em *Secure Socket Layer* (SSL). Um fator ainda limitante deste serviço é não oferecer suporte ao alfabeto latino, impedindo portanto o uso de caracteres especiais como cedilhas, acentos, etc. Mais informações: <https://dev.havenondemand.com/apis>.
- ***Free OCR Webservice***: oferece interface *webservice Simple Object Access Protocol* (SOAP) e *Representational State Transfer* (REST) para uso de mecanismo OCR com excelente resultado, suporte a língua portuguesa e PDF em múltiplas páginas. A principal desvantagem no modo de uso gratuito é o limite de 15 páginas/documentos por hora. Mais informações: <http://www.onlineocr.net/service/keyfeatures>.
- ***Free Online OCR***: por meio de uma API REST de simples aplicação, oferece um serviço de OCR para desenvolvedores com respostas em JSON. Aceita o alfabeto latino e tem uma limitação de 200 páginas digitalizadas gratuitamente. Mais informações: <https://www.newocr.com/api/>.
- ***ABBYY Cloud OCR SDK***: *webservice premium* ganhador de vários prêmios de precisão, com taxas de reconhecimento aproximadas de 99.8%. Rápido e de fácil acesso, o serviço da ABBYY oferece suporte nativo a JAVA, C#, C++, Ruby, Python, Perl, Objective-C e PHP. Possui limitação de uso de 50 páginas e sua grande vantagem é

oferecer licença *premium* gratuita para estudantes. Mais informações: <http://ocrsdk.com/>.

Pelo excelente resultado obtido em testes com alguns artigos, o *webservice* da ABBYY foi escolhido como referência para este estudo. Pela obtenção da licença estudantil, é possível se digitalizar 5000 páginas por mês utilizando o *Software Development Kit* (SDK) oferecido pela empresa.

Assim, todos os artigos listados pelo Apêndice A foram convertidos em formato de texto puro e enviados ao processo de indexação, conforme detalha melhor a próxima seção.

3.2.4.2.5 Processo de indexação

Dos passos do processo de indexação citados na seção 2.5 deste estudo, foram realizados: 1) a coleta dos documentos para indexação em *full text*, onde o texto passou por substituição de caracteres especiais e normalização para minúsculas e; 2) o fatiamento do texto em *tokens*, contrastando-os contra a *blacklist*, como se observa pelo Código-fonte 8.

Código-fonte 8 – Exemplo de *seeder* para alimentação do índice

```

1  <?php
2
3  class TermoIndicesSeeder extends Seeder {
4      /**
5       * Run the database seeds.
6       * @return void
7       */
8      public function run()
9      {
10         $dados = [
11             [1, '93-686-1-PB.txt'],
12             [2, '96-697-1-PB.txt']
13         ];
14
15         $blacklist = Termo::getBlacklist();
16         ini_set('memory_limit','1024M');
17         DB::disableQueryLog();
18
19         foreach ($dados as $d)
20         {
21             // obtem o caminho do documento no servidor

```

```

22     $caminho = storage_path().'/files/'. $d[1];
23     // normaliza o texto, sem caracteres especiais nem maiusculas
24     $arquivo = strtolower(Util::removeAcentos(file_get_contents($caminho)));
25     // regex para pegar apenas palavras
26     preg_match_all('/[a-zA-Z]+/', $arquivo, $termos);
27
28     // para todos os tokens adiciona ao índice
29     foreach ($termos[0] as $termo) {
30         if (strlen($termo) > 1) {
31             if (!in_array($termo, $blacklist)) {
32                 // id do documento
33                 $doc_id = $d[0];
34                 // id do termo
35                 $ter_id = Termo::adicionaTermo($termo);
36
37                 Indice::adiciona($ter_id, $doc_id);
38             }
39         }
40     }
41     // liberar a memória
42     unset($caminho);
43     unset($arquivo);
44     unset($termos);
45 }
46 }
47 }

```

Fonte – Elaborado pelo autor.

Exemplificando o processo de indexação com apenas dois documentos, o Código-fonte 8 demonstra as primeiras duas etapas do processo de indexação, lembrando que as outras duas, radicalização e atribuição de pesos em tempo de indexação, são etapas opcionais e não contempladas para este estudo. Usando a função *file_get_contents()*, linha 24, o PHP é capaz de carregar todo o conteúdo textual do arquivo, o qual serve de entrada para uma função de substituição de caracteres, *removeAcentos()*, que por sua vez serve os dados como entrada para a função *strtolower()* nativa do PHP, responsável por terminar a normalização do conteúdo. A próxima etapa se responsabiliza pela quebra do texto em *tokens*, ignorando espaços, acentos, parêntesis, chaves e demais conteúdos que não formam palavras. Usando a função *preg_match_all('/[a-zA-Z]+/', [texto], [resultado])*, linha 26, que utiliza uma expressão regular como primeiro parâmetro, é possível se realizar tal etapa sem grande dificuldade.

Por fim, o código-fonte finaliza na análise iterativa entre todos os *tokens* (linha 29), verificando se pertencem ao *blacklist* para serem descartados, caso contrário, são persistidos o termo, bem como indexada sua relação com o documento em questão. Os Código-fonte 9 e 10 demonstram a persistência de termos e indexação dos mesmos, terminando assim o processo de indexação de documentos no SRI proposto.

Código-fonte 9 – Exemplo de adição de termo ao índice

```

1  public static function adicionaTermo($termo) {
2      //verifica se o termo já existe
3      try {
4          $existe = Termo::where('palavra', '=', $termo)->firstOrFail();
5          // retorna o id do objeto já existente
6          return $existe->id;
7      } catch (Exception $e) {
8          $t = new Termo();
9          $t->palavra = $termo;
10         $t->save();
11
12         // retorna o id do objeto recém inserido
13         return $t->id;
14     }
15 }

```

Fonte – Elaborado pelo autor.

Usando-se o ORM nativo do Laravel, o Eloquent, verifica-se que o processo de persistência de novos registros se torna mais simples. O Código-fonte 9 apenas verifica se o termo existe (linha 4), caso afirmativo, não o insere e retorna o identificador previamente cadastrado (linha 6). Caso negativo, o insere e retorna o identificador do registro recém cadastrado (linha 13).

Código-fonte 10 – Exemplo de adição de indexação de termo-documento ao índice

```

1  public static function adiciona($ter_id, $doc_id) {
2      //verifica se o termo já existe
3      try {
4          $existe = Indice::whereRaw('indice.ter_id = ?
5              AND indice.doc_id = ?',
6              array($ter_id, $doc_id))->firstOrFail();
7
8          $i = $existe;
9          $i->frequencia++;
10         $i->save();
11         return $existe->id;      // retorna o id do objeto
12     } catch (Exception $e) {
13         $i = new Indice();
14         $i->ter_id = $ter_id;
15         $i->doc_id = $doc_id;
16         $i->frequencia = 1;
17         $i->save();
18         return $i->id;      // retorna o id do objeto recém inserido
19     }
20 }

```

Fonte – Elaborado pelo autor.

O Código-fonte 10 também expressa o uso da persistência pelo ORM Eloquent. Neste código-fonte, basicamente é verificada a existência do termo (linha 4) passado como parâmetro no índice referente ao documento, caso exista, sua frequência é incrementada (linha 9), caso contrário, uma nova indexação de termo é feita a tal documento (linha 13) iniciando sua frequência em um.

Estes passos repetidos iterativamente por todos os *tokens* retirados dos documentos coletados para indexação completam o processo de indexação do índice invertido proposto.

3.3 Estatísticas de implementação

O Quadro 9 apresenta algumas estatísticas sobre a implementação com o objetivo de apresentar uma noção quantitativa do esforço envolvido no desenvolvimento do SRI proposto. Foram considerados os totais de classes implementadas, linhas de código, comentários de código, documentos indexados, número de termos de indexação, entidades em banco de dados relacional e atributos de entidades implementadas por classes e interfaces.

Quadro 9 – Estatísticas de implementação do trabalho proposto

ITEM	QUANT.*	UN.
Classes	16	classes
Código-fonte	3543	linhas
Comentário	323	linhas
Entidades (tabelas em banco de dados)	12	tabelas
Atributos (de entidades)	65	atributos
Métodos (das classes)	44	métodos
Modelos de Recuperação da Informação	3	modelos
Documentos indexados	200	documentos
Termos indexados	39.623 ²³	termos

*foram consideradas as estatísticas do sistema de apoio à classificação de relevância dos documentos para construção da coleção de referência (ver seção 2.5.1).

Fonte – Elaborado pelo autor.

²³ 39.623 termos de indexação, dividido pelo número de documentos (200), equivale a uma média de 198 termos por documento, valor muito abaixo da quantidade de palavras existente em um artigo.

3.4 Implementação dos modelos clássicos

3.4.1 Modelo Booleano

Um leigo poderia imaginar uma máquina de busca como uma simples ideia: dado um certo conjunto de arquivos, recebe-se uma consulta, buscando seus termos em todos os documentos e retornam-se todos os que possuem tais termos. Este não é um pensamento errado, afinal utilizando por exemplo a função *grep*²⁴ em ambientes Unix, uma lista de documentos que contém certo termo rapidamente é retornada. Entretanto, a função *grep* não é escalável a grandes coleções, sendo necessários recursos mais poderosos para o uso em grande coleções.

O modelo booleano de Recuperação da Informação pode suprir tal escalabilidade. Para esta implementação foram consideradas consultas com o formato clássico, como exemplo **ifmg AND matemática NOT 2014**, que significa que os documentos retornados devem possuir o termo **ifmg** e o termo **matemática**, mas não devem possuir o termo **2014**. Pode-se ainda utilizar parêntesis como operadores de prioridade de operações, como exemplo: **(ifmg AND matemática) OR (ifmg AND administração)**, que retornaria documentos que possuem os termos **ifmg** e **matemática** ou documentos que possuem os termos **ifmg** e **administração**.

Com um índice invertido montado para um determinado *corpus*, ao efetuar-se uma consulta pelo sistema, a mesma passa pelo processo de especificação de consulta abordado neste estudo, seguindo os passos iniciais de acordo com o Código-fonte 11.

Código-fonte 11 – Processo de especificação de consulta no modelo Booleano

```

1  class ConsultaBooleana {
2      private $indice = 0;           // indice de controle para varrer tokens da pesquisa
3      private $tokens = array();    // coleção de tokens derivados da consulta
4      private $totalTermos;        // número total de termos (tokens) encontrados
5      private $arvore;             // árvore de busca construída para a consulta
6      private $termos_ignorados = array(); // array de termos ignorados da busca
7
8      // método construtor da consulta booleana
9      public function __construct($consulta) {
10         // limpa a string de consulta, removendo caracteres especiais e acentos

```

²⁴ *grep*: a função **grep [expressão de busca] [diretório]** retorna todos os arquivos no diretório passado como parâmetro que contêm a expressão buscada. Mais informações: <http://www.gnu.org/software/grep/manual/grep.html>

```

11 $consulta = strtolower(Util::removeAcentos($consulta));
12
13 // quebra a string de consulta, dividindo-a em tokens
14 preg_match_all('/[a-zA-Z]+|[\(\)]/', $consulta, $encontrados);
15 $this->totalTermos = count($encontrados[0]); // total de tokens encontrados
16 $this->tokens = $encontrados[0]; // array de tokens encontrados
17 $this->arvore = $this->montaArvoreConsulta(); // arvore de busca montada
18 }
19 //[...]

```

Fonte – Elaborado pelo autor.

O Código-fonte 11 apenas apresenta a primeira parte da classe de consulta booleana, mostrando os atributos da classe e seu construtor. Percebe-se que os atributos não possuem tipos definidos, afinal esta é uma característica da linguagem PHP. Dentre os atributos da classe podemos destacar: a lista de *tokens* da consulta, o total de termos da expressão de busca, a árvore, inicialmente nula, e uma lista de termos ignorados para a consulta. Neste último, os termos ignorados são aqueles removidos da consulta por constarem na *blacklist* do SRI. O construtor da classe basicamente realiza o processo de especificação da consulta do usuário, finalizando com a árvore binária de busca montada segundo o Código-fonte 12.

Código-fonte 12 – Montagem da árvore binária de busca no modelo Booleano

```

1 private function montaArvoreConsulta() {
2     $blacklist = Termo::getBlacklist();
3     // enquanto não passar por todos os tokens da consulta
4     while($this->indice < $this->totalTermos) {
5         // pega o token do índice atual
6         $token = $this->tokens[$this->indice];
7         $this->indice++;
8         // se o termo atual estiver na blacklist, então é ignorado
9         if (in_array($token, $blacklist)) {
10            $this->termos_ignorados[] = $token;
11            if (!in_array($token, ['and', 'or', 'not', '(', ')', ''])) {
12                // se o token estiver na blacklist e não estiver dentre os operadores
13                // da consulta booleana, então pula essa iteração
14                continue;
15            }
16        }
17        // verifica qual é o token
18        switch ($token) {
19            case '(':
20                // se for um parêntesis de abertura, recomeça uma subarvore aqui até que se ache um )
21                $arvore = $this->montaArvoreConsulta();
22                break;
23            case ')':
24                return $arvore; // finaliza a subárvore que estava sendo criada
25            case 'and':
26                // -----

```

```

26     case 'or':                                     // palavras reservadas do método booleano
27     case 'not':                                    // -----
28         $arvore = [
29             'acao' => $token,                       // o token atual vira um nó intermediário da árvore
30             'esquerda' => $arvore,                 // a esquerda continua com q que já havia sido construído
31             'direita' => $this->montaArvoreConsulta() // a direita vira uma nova subárvore
32         ];
33     break;
34     default:
35         $arvore = $token;                          // caso não seja nenhum caso anterior, então é um legítimo token
36     break;
37 }
38 }
39 return $arvore;
40 }

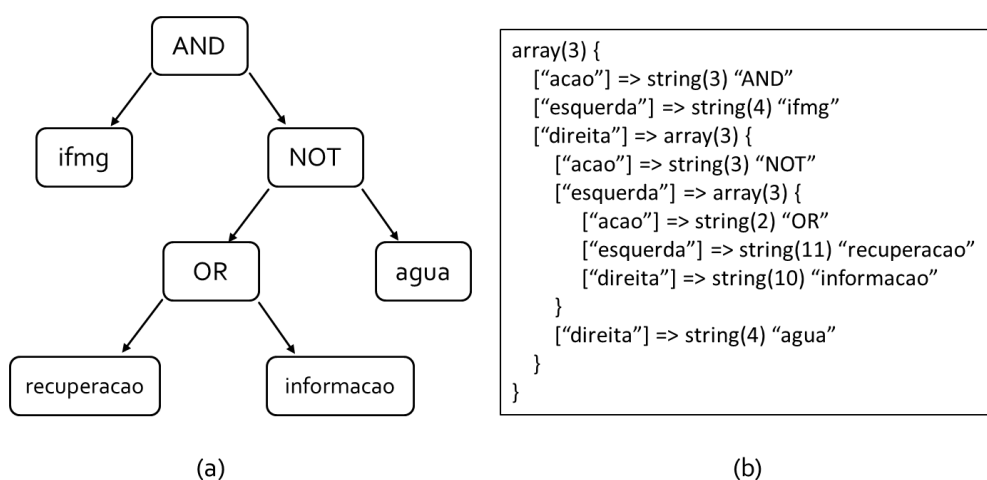
```

Fonte – Elaborado pelo autor.

A etapa de criação da árvore de busca é vital para o sucesso e correta recuperação no modelo Booleano. Esta etapa consiste na iteração entre todos os termos da expressão de busca, verificando-se quais devem ser ignorados, devido à sua presença na *blacklist*, e quais devem constituir a árvore de consulta (linha 11). Pela utilização de uma estrutural de fluxo condicional *switch* (linha 21), a árvore é montada por meio de chamadas recursivas ao mesmo método (linha 24, por exemplo). Os comentários do Código-fonte 12 podem melhor explicar seus passos.

Para melhor apresentar a árvore de busca gerada pelo Código-fonte 12, tomemos a seguinte expressão de busca, **ifmg AND (recuperação OR informação) NOT água**, e seu resultado na forma de árvore, conforme ilustra a Figura 28.

Figura 28 – Exemplo de árvore gerada pela consulta IFMG AND (recuperação OR informação) NOT água



Fonte – Elaborada pelo autor.

A Figura 28.a apresenta a árvore binária gerada pela consulta exemplo, onde os nós intermediários são operações entre conjuntos e as folhas são os termos da expressão de busca. A figura 28.b é o resultado da mesma árvore como estrutura de *arrays* do PHP, obtida pelo uso da função nativa *print_r(array)*. Assim, o atributo *\$arvore*, apresentado na classe ConsultaBooleana do Código-fonte 11, recebe uma árvore binária de busca como resultado do processo de especificação da consulta.

O próximo passo para a conclusão da consulta é demonstrado pelo método de processamento da consulta, como visto no Código-fonte 13 a seguir.

Código-fonte 13 – Montagem da árvore binária de busca no modelo Booleano

```

1 // Processamento da consulta na árvore binária
2 // utiliza o caminhamento pós-ordem para resolver a consulta
3 private function processaConsulta($arvore) {
4     // se for uma árvore
5     if(is_array($arvore)) {
6         $esquerda = $this->processaConsulta($arvore['esquerda']); // chamada recursiva à esquerda
7         $direita = $this->processaConsulta($arvore['direita']); // chamada recursiva à direita
8
9         switch($arvore['acao']) { // multiplexador de operações lógicas
10            case 'and':
11                return $this->intersecao($esquerda, $direita);
12            case 'or':
13                return $this->uniao($esquerda, $direita);
14            case 'not':
15                return $this->complemento($esquerda, $direita);
16        }
17    } else {
18        // neste caso a árvore é uma folha, ou seja, apenas um token da consulta
19        return Indice::busca($arvore);
20    }
21 }

```

Fonte – Elaborado pelo autor.

Por meio da recursão, o Código-fonte 13 oferece a solução simples para o processamento da árvore de busca montada. As operações entre conjuntos, necessárias na instrução *switch* do código-fonte apresentado são diretamente relacionadas às funções nativas do PHP para manipulação de vetores, como demonstra o Código-fonte 14.

Código-fonte 14 – Métodos de manipulação de conjuntos

```

1 // método para realizar a união de conjuntos
2 private function uniao($A, $B) {
3     return array_unique(array_merge($A, $B));

```

```

4   }
5
6   // método para realizar a interseção de conjuntos
7   private function intersecao($A, $B) {
8       return array_unique(array_intersect($A, $B));
9   }
10
11  // método para realizar o complemento entre conjuntos
12  private function complemento($A, $B) {
13      return array_unique(array_diff($A, $B));
14  }

```

Fonte – Elaborado pelo autor.

Observando os métodos do Código-fonte 14 percebe-se que além do uso de *merge()*, *intersect()* e *diff()*, em todos os casos, se faz necessária a garantia da unicidade dos termos pelo uso da função *array_unique()*, simulando assim o comportamento natural das operações entre conjuntos.

O modelo Booleano não é ideal, mas é simples. Por não ser muito intuitivo, a maioria das pessoas tendem a inverter o significado dos operadores. Contudo, é um importante passo para entender melhor os dois modelos dominantes de busca: o modelo Vetorial e o modelo Probabilístico, que serão abordados nas seções seguintes.

3.4.2 Modelo Vetorial

Como abordado em seções anteriores, um dos principais problemas com o modelo Booleano é que o resultado não é classificado e nem ordenado, fazendo com que cada documento que tem um casamento com a consulta retorne como resultado. Não há uma forma real de se apontar qual documento é melhor ou mais relevante que o outro no resultado obtido. Contudo, se pudermos atribuir pesos aos termos, podemos ordenar o resultado entre os documentos que melhor se casam com aquela consulta, o que forma a ideia principal do modelo vetorial, também chamado de modelo de espaço vetorial.

Neste estudo seguiu-se o esquema de atribuição de pesos *tf-idf*, abordados na seção 2.4.3. Utilizando o *term frequency*, colocamos um número associado a cada termo dentro de um documento, por exemplo, se um determinado termo aparece 15 vezes em um documento, seu *tf* é igual a 15. Por ser um estatística local a um documento e que não considera os outros documentos da coleção, é notável que termos como *da*, *do*, *as*, *mas* e *pois* terão altos pesos de

tf, por isso se torna tão importante o uso de *blacklists*, a fim de ignorar tais termos no momento da consulta, tornando-a assim mais ágil.

Em contrapartida, a medida inversa do *document frequency* surge para resolver o problema das palavras muito recorrentes. A estatística global *inverse document frequency idf* é o peso de um termo considerando-se sua ocorrência em toda a coleção. Pela razão entre o número de documentos total da coleção e o número de documentos que contêm tal termo, obtém-se a noção de que um termo é mais relevante na classificação se for mais incomum, mais raro, representando melhor o conteúdo de um documento.

Relembrado o esquema de atribuição de pesos aos termos/documentos abordado na seção 2.4.3, o Código-fonte 15 demonstra o processo de especificação de consulta no modelo vetorial pelo construtor da classe implementada para este estudo.

Código-fonte 15 – Processo de especificação de consulta no modelo Vetorial

```

1  <?php
2
3  class ConsultaVetorial {
4      private $tokens = [];           // coleção de tokens derivados da consulta
5      private $totalTermos;          // número total de termos (tokens) encontrados
6      private $matriz = [[]];        // matriz de busca tf-idf construída para a consulta
7      private $termos_ignorados = array(); // array de termos ignorados da busca
8      private $numTotalDocs = 0;     // total de documentos no índice
9      private $resultado = [];
10     private $docs = [];
11
12     // método construtor da consulta vetorial
13     public function __construct($consulta) {
14         // limpa a string de consulta, removendo caracteres especiais e acentos
15         $consulta = strtolower(Util::removeAcentos($consulta));
16
17         // tokeniza a string de consulta, dividindo-a em tokens
18         preg_match_all('[a-zA-Z]+|[\\(\\)]', $consulta, $encontrados);
19         $this->totalTermos = count($encontrados[0]); // total de tokens encontrados
20         $this->tokens = $encontrados[0];           // array de tokens encontrados
21         // inteiro com o total de documentos do índice
22         $this->numTotalDocs = Indice::getNumTotalDocumentos();
23         $this->matriz = $this->montaMatrizConsulta(); // matriz de busca montada
24     }
25     //[...]

```

Fonte – Elaborado pelo autor.

Dentre os passos de normalização executados pelo construtor, como remoção de caracteres especiais e conversão de maiúsculas em minúsculas (linha 15), ressalta-se a alimentação da matriz pelo método *montaMatrizConsulta()* (linha 23), método básico para a

função de *ranking* do modelo Vetorial que ocorre após a quebra do texto em *tokens*. Seu exemplo é dado pelo Código-fonte 16.

Código-fonte 16 – Montagem da matriz de consulta no modelo Vetorial

```

1  private function montaMatrizConsulta() {
2      $blacklist = Termo::getBlacklist();
3      $matriz = [];
4
5      // recupera os arquivos que têm os termos da busca
6      foreach ($this->tokens as $termo) {
7          // se o termo atual estiver na blacklist, então é ignorado e adicionado à lista de ignorados
8          if (in_array($termo, $blacklist)) {
9              $this->termos_ignorados[] = $termo;
10             continue;
11         }
12
13         // colhe todos os documentos envolvidos com os termos para posterior classificação
14         $docIDs = Indice::busca($termo);
15         // alimenta o indice qtd com o número total de docs que têm cada termo (DF)
16         $matriz[$termo]['qtd'] = count($docIDs);
17         $this->docs = array_unique(array_merge($this->docs, $docIDs));
18     }
19
20     // inclui o doc id 0, que representa a consulta (query)
21     $this->docs[] = 0;
22
23     // varre os termos e seus arquivos montando seu valor TF-IDF
24     foreach($this->tokens as $termo) {
25         // se o termo atual estiver na blacklist, então é ignorado
26         if (in_array($termo, $blacklist)) {
27             continue;
28         }
29
30         try {
31             // para cada documento, calcula seu TFIDF (preenchimento das células da tabela)
32             foreach ($this->docs as $doc) {
33                 $matriz[$termo][$doc] = $this->calculaTFIDF($termo, $doc, $matriz[$termo]['qtd']);
34             }
35         } catch (Exception $e) {
36             return 'ERRO: '.$e;
37         }
38     }
39
40     // normalizar cada vetor formado na tabela (VETOR = Representação lógica de DOCUMENTO)
41     foreach ($this->docs as $doc) {
42         $this->normalizar($matriz, $doc);
43     }
44
45     return $matriz;
46 }

```

Fonte – Elaborado pelo autor.

A observação do Código-fonte 16 junto aos comentários dispostos no código-fonte oferecem a explicação de três grandes etapas realizadas: 1) a alimentação de uma dimensão da matriz pelos termos da expressão de busca de da outra dimensão pelos documentos selecionados pelo índice do SRI como relevantes à consulta (linha 6); 2) o cálculo do *tf-idf* para todos os termos/documentos envolvidas na consulta, de acordo com o SRI (linha 24), e; 3) a normalização dos vetores criados para o espaço *n*-dimensional da consulta (linha 41).

O Código-fonte 16 inicia pela verificação de termos que serão ignorados na consulta, decidido pelo cruzamento destes com a *blacklist*. A próxima etapa é a busca no índice por documentos relevantes à consulta, recuperando-se assim uma lista de números identificadores de documentos. A indexação de posições no vetor por *strings*, nativo no PHP, é um recurso muito propício neste caso, colocando-se assim cada termo como índice de uma posição, onde podemos alocar dados como o *tf-idf* e o *df*. Um passo importante neste momento é a inclusão do documento de *id 0*, representando o vetor consulta no espaço *n*-dimensional criado. Os dois passos seguintes são o cálculo de *tf-idf* para cada termo/documento do espaço (ver Código-fonte 17) e posterior normalização de todos os vetores, tornando-os versores²⁵ (ver Código-fonte 18).

Código-fonte 17 – Método de cálculo do *tf-idf*

```

1  private function calculaTFIDF($termo, $docID, $qtd) {
2      if ($qtd == 0) {
3          return 0;
4      }
5
6      if ($docID == 0) { //se for a consulta, seu id = 0 e sua frequência = 1
7          $frequencia_termo = 1;
8      } else {
9          $frequencia_termo = Indice::getFrequencia($termo, $docID);
10     }
11
12     //soma-se um devido ao documento de consulta (query)
13     $num_total_docs = $this->numTotalDocs + 1;
14     $docs_com_termo = $qtd + 1;
15
16     return $frequencia_termo * log( $num_total_docs / $docs_com_termo, 2);
17 }

```

Fonte – Elaborado pelo autor.

²⁵ versor: é o vetor unitário de mesma direção e sentido que um dado outro vetor. (STEINBRUCH; WINTERLE, 1987, p. 5).

A montagem da matriz de frequência termo-documento segue a mesma lógica apresentada na seção 2.4.2, porém ao invés de cada interseção termo-documento armazenar o valor de frequência, é armazenado o cálculo de *tf-idf* diretamente. Para o cálculo de *tf-idf* deve ser observado (ver Código-fonte 17) que no caso do vetor consulta, seu *tf* deve ser igual a um, ou seja, o vetor de representação lógica da consulta é o vetor onde todas as componentes são iguais a um.

Ainda conforme demonstrado pelo Código-fonte 17 e abordado pela seção 2.4.3 (ver Tabela 1), o cálculo do *tf-idf* é o mero resultado da multiplicação do *tf* pelo *idf*. O uso do logaritmo sob o *idf* provê uma suavização (BAEZA-YATES; RIBEIRO-NETO, 2012, p. 34), por exemplo: se um dado termo1 é representado em X documentos, e um termo2 em $2X$ documentos, logo o termo1 é mais específico e deve retornar melhores resultados, mas não tem necessariamente o dobro de relevância na consulta. Assim o uso do logaritmo suaviza a diferença entre os dois termos.

O Código-fonte 18 apresenta o método de normalização das componentes dos vetores, tornando-os assim versores.

Código-fonte 18 – Normalização das componentes de um vetor: criação de versor

```

1  private function normalizar(&$matriz, $doc) {
2      $total = 0;
3      $blacklist = Termo::getBlacklist();
4
5      foreach($this->tokens as $termo) {
6          if (in_array($termo, $blacklist)) {
7              continue;
8          }
9          // eleva cada componente ao quadrado
10         $total += $matriz[$termo][$doc] * $matriz[$termo][$doc];
11     }
12
13     $total = sqrt($total);
14
15     if ($total > 0) {
16         foreach($this->tokens as $termo) {
17             // validação blacklist
18             if (in_array($termo, $blacklist)) {
19                 continue;
20             }
21             // divisão de cada componente pelo tamanho do vetor = normalização
22             $matriz[$termo][$doc] = $matriz[$termo][$doc] / $total;
23         }
24     }
25 }

```

Fonte – Elaborado pelo autor.

A normalização aqui abordada é descrita por Steinbruch e Winterle (1987, p. 40-41) como a criação de um versor, que nada mais é que a divisão de cada componente do vetor pelo módulo de suas componentes (linha 22), conforme a Equação 18.

$$\text{versor de } \vec{v} = \frac{\vec{v}}{|\vec{v}|} \quad (18)$$

A vantagem do uso dessa normalização está na simplificação do cálculo de similaridade por cosseno (ver Equação 6) que basicamente se torna o produto interno entre dois vetores a fim de se encontrar o cosseno do ângulo formado entre cada vetor documento e o vetor consulta, como demonstra o Código-fonte 19 que é a função de similaridade implementada para o modelo vetorial deste estudo.

Código-fonte 19 – Método de cálculo da proximidade de um vetor documento ao vetor consulta

```

1  private function calculaSimilaridade(&$matriz, $doc) {
2      $compA = []; // vetor de consulta
3      $compB = []; // vetor do documento
4      $blacklist = Termo::getBlacklist();
5      $result = 0;
6
7      foreach($this->tokens as $termo) {
8          if (in_array($termo, $blacklist)) {
9              continue;
10         }
11
12         $compB[] = $matriz[$termo][$doc];
13         $compA[] = $matriz[$termo][0]; //vetor consulta
14     }
15
16     for ($i=0; $i < count($compB) ; $i++) {
17         $result += $compA[$i] * $compB[$i]; // cálculo do cos(theta)
18     }
19
20     return $result;
21 }

```

Fonte – Elaborado pelo autor.

O Código-fonte 19 apresenta o produto interno entre cada componente de um vetor documento e o vetor consulta (linha 17) no espaço vetorial criado, sempre lembrando da remoção dos termos considerados na *blacklist*.

Para finalizar o processo de consulta implementado para o modelo vetorial, o Código-fonte 20 demonstra a solicitação do cálculo do cosseno do ângulo formado entre cada documento e o vetor consulta e sua posterior ordenação pelos cálculos de similaridade obtidos pela função cosseno, denominado no modelo como função de *ranking* (ver Código-fonte 19).

Código-fonte 20 – Método de busca vetorial

```

1  public function buscar() {
2      // calcular similaridades por cosseno de ângulos entre os vetores
3      foreach ($this->docs as $doc) {
4          //salta o vetor consulta
5          if ($doc == 0) {
6              continue;
7          }
8
9          // armazena o vetor resultado com os valores obtidos pelo produto
10         // interno de cada vetor com a consulta
11         $this->resultado[$doc] = $this->calculaSimilaridade($this->matriz, $doc);
12     }
13
14     arsort($this->resultado); // função nativa de ordenação do PHP
15     return $this->resultado;
16 }

```

Fonte – Elaborado pelo autor

Após o cálculo das similaridades pelo Código-fonte 19, no Código-fonte 20 podemos ver que utilizando a função *arsort()* nativa do PHP, do inglês *array reverse sort*, o resultado da classificação e ordenação de forma decrescente do modelo vetorial implementado (linha 14) é retornado à camada de controle da aplicação para que o *IRController* assuma a responsabilidade de encaminhá-lo formatado ao usuário.

3.4.3 Modelo Probabilístico

O esquema de atribuição de pesos a termos e funções de *ranking* formam o núcleo de todo Sistema de Recuperação da Informação moderno. O modelo Vetorial, abordado na última seção deste trabalho, com sua função de similaridade por cálculo de cossenos, provavelmente é o modelo mais conhecido e de longe o mais usado. Entretanto, existem várias alternativas ao modelo Vetorial, dentre elas o modelo Probabilístico com uso do esquema de atribuição de pesos aos termos denominado BM25, abordado na seção 2.7.3.

O modelo OKAPI/BM25 é guiado pela incerteza inerente no retorno de resultados ao usuário, onde não se sabe exatamente se os documentos entregues realmente são aquilo o que o usuário queria ou aquilo o que ele tentou expressar com sua expressão de busca. Assim, ao invés de tentar classificar o resultado de uma busca pela similaridade de documentos à consulta, neste modelo o objetivo principal está na probabilidade daqueles documentos retornados como resultado serem relevantes ao usuário, o que de fato é um processo não trivial de se estimar.

Assumamos uma simplificação à implementação do modelo Probabilístico clássico para que ele trabalhe de forma independente, ou seja, não assistido pelas etapas iterativas de *relevance feedback* (ver seção 2.7.3). Nesta implementação assume-se que a probabilidade de um documento ser relevante para o usuário é igual ao produto das probabilidades de cada termo ser relevante ao usuário.

Código-fonte 21 – Processo de especificação de consulta no modelo Probabilístico

```

1  <?php
2
3  class ConsultaProbabilistica {
4      private $tokens = [];           // coleção de tokens derivados da consulta
5      private $totalTermos;          // número total de termos (tokens) encontrados
6      private $termos_ignorados = array(); // array de termos ignorados da busca
7      private $numTotalDocs = 0;     // total de documentos no índice
8      private $mediaTermosPorDocumento = 0; // média de termos por documento
9      private $resultado = [];
10     private $docs = [];
11     private $consulta;
12
13     // método construtor da consulta booleana
14     public function __construct($consulta) {
15         // limpa a string de consulta, removendo caracteres especiais e acentos
16         $consulta = strtolower(Util::removeAcentos($consulta));
17         $this->consulta = $consulta;
18
19         // tokeniza a string de consulta, dividindo-a em tokens
20         preg_match_all('/[a-zA-Z]+|[\\(\\)]/', $consulta, $encontrados);
21         $this->totalTermos = count($encontrados[0]); // total de tokens encontrados na consulta
22         $this->tokens = $encontrados[0];           // array de tokens encontrados na consulta
23         // inteiro com o total de documentos do índice
24         $this->numTotalDocs = Indice::getNumTotalDocumentos();
25         $this->mediaTermosPorDocumento = Indice::getMediaTermosPorDocumento();
26     }
27     //[...]

```

Fonte – Elaborado pelo autor.

De acordo com o Código-fonte 21, o processo de especificação da consulta no modelo Probabilístico se inicia pela normalização da expressão de consulta (linha 16), substituindo-se

caracteres especiais e letras maiúsculas por minúsculas. Observa-se no Código-fonte 21 que diferentemente dos outros modelos implementados neste trabalho, o construtor do modelo Probabilístico não inicializa uma estrutura de dados como uma árvore ou uma matriz para atribuir pesos aos termos relacionados aos documentos, pois utiliza-se de elementos da teoria das probabilidades para chegar ao resultado esperado pelo usuário. Duas estatísticas obtidas em todo o *corpus* são adquiridas ao final do construtor (linhas 24 e 25): o número total de documentos na coleção e a média de termos por documentos, que respectivamente são atendidas pelos métodos do Código-fonte 22.

Código-fonte 22 – Métodos auxiliares da classe Índice

```

1 // retorna o total de documentos no indice
2 public static function getNumTotalDocumentos() {
3     $req = Indice::select('doc_id')->groupBy('doc_id')->count();
4     return $req;
5 }
6
7 // retorna a média de termos por documento
8 public static function getMediaTermosPorDocumento() {
9     $totalDocumentos = self::getNumTotalDocumentos();
10    $qtd = Indice::all()->sum('frequencia');
11
12    if ( ($totalDocumentos > 0) AND ($req) ) {
13        return ($qtd / $totalDocumentos); //retorna valor inteiro
14    } else {
15        return 0;
16    }
17 }

```

Fonte – Elaborado pelo autor.

Após o cálculo das estatísticas do índice (Código-fonte 22) pelo construtor, o controlador dispara o método de busca da classe *ConsultaProbabilistica*, que invoca a função *bm25()*, cujo comportamento é dado no código do Código-fonte 23.

Código-fonte 23 – Método de atribuição de pesos BM25

```

1 private function bm25($pesoTF = 1, $pesoTamanhoDoc = 0.5) {
2     $blacklist = Termo::getBlacklist();
3     $dados = [];
4     $resultado = array();
5
6     // recupera os arquivos que têm os termos da busca
7     foreach ($this->tokens as $termo) {
8         // se o termo atual estiver na blacklist, então é ignorado e adicionado à lista de ignorados
9         if (in_array($termo, $blacklist)) {

```

```

10     $this->termos_ignorados[] = $termo;
11     continue;
12 }
13
14 // colhe todos os documentos envolvidos com os termos para posterior classificação
15 $docIDs = Indice::busca($termo);
16 // alimenta o indice qtd com o número total de docs que têm cada termo
17 $dados[$termo]['qtd'] = count($docIDs);
18 $this->docs = array_unique(array_merge($this->docs, $docIDs));
19 }
20
21 // varre os termos e seus arquivos montando seu valor de pesos probabilísticos
22 foreach($this->tokens as $termo) {
23     // se o termo atual estiver na blacklist, então é ignorado
24     if (in_array($termo, $blacklist)) {
25         continue;
26     }
27
28     try {
29         // quantidade de documentos com termo (Document Frequency)
30         $df = $dados[$termo]['qtd'];
31
32         // para cada documento, calcula seu peso
33         foreach ($this->docs as $doc) {
34             // pega a frequência do termo do documento (Term Frequency)
35             $tf = Indice::getFrequencia($termo, $doc);
36             // pega o número de termos do documento
37             $numTermos = Indice::getNumTermos($doc);
38             // calcula o inverse document frequency
39             $idf = log($this->numTotalDocs / $df);
40
41             // calcula o numerador e denominador do bm25
42             $numerador = ($pesoTF + 1) * $tf;
43             $denominador = $pesoTF * ((1 - $pesoTamanhoDoc) + $pesoTamanhoDoc *
44                 ($numTermos / $this->mediaTermosPorDocumento)) + $tf;
45
46             //define a pontuação final para cada documento
47             $pontuacao = $idf * ($numerador / $denominador);
48
49             if (isset($resultado[$doc])) {
50                 $resultado[$doc] += $pontuacao;
51             } else {
52                 $resultado[$doc] = $pontuacao;
53             }
54         }
55     } catch (Exception $e) {
56         return 'ERRO: '.$e;
57     }
58 }
59 return $resultado;
60 }

```


O Código-fonte 23 consiste em duas grandes iterações por todos os termos da busca. A primeira realiza a recuperação dos termos da expressão de busca e dos documentos que possuem relação com a expressão de busca no índice do SRI, além da verificação da *blacklist* (linha 7); a segunda realiza a atribuição de pesos aos termos de acordo com a Equação 14 do cálculo do BM25 (linha 22), abordada na seção 2.7.3 deste estudo.

A primeira etapa é similar ao processo executado nos outros modelos, verificando-se cada termo se existe sua ocorrência na *blacklist* para ignorá-lo (linha 9). Logo após este passo, uma lista de documentos com casamento com os termos é recuperada (linha 15), simbolizados pelos seus números identificadores. É alimentado o *document frequency* df em `$matriz[$termo][‘qtd’]` para cada termo (linha 17) e, por fim, são adicionados ao atributo de documentos da classe os números identificadores dos documentos relacionados a cada termo analisado (*matching*) (linha 18). Este último passo é realizado incrementalmente sempre preocupando-se com a não redundância dos identificadores, garantido pela função `array_unique()` do PHP.

A segunda etapa é a execução propriamente dita da função de atribuição de pesos BM25. Primeiramente podemos observar que a função aceita dois parâmetros que por padrão têm seu valor *default* atribuído como `$pesoTF = 1` e `$pesoTamanhoDoc = 0.5` (linha 1), que simbolizam respectivamente os valores das variáveis c e b da Equação 14 (ver seção 2.7.3). De acordo com Manning, Raghavan e Schütze (2009, p. 233), estes parâmetros são definidos para otimizar a performance do modelo, ou seja, devem ser atribuídos valores de forma experimental, observando empiricamente quais oferecem melhores resultados. O significado da aplicação destes parâmetros na fórmula de cálculo do BM25 é justamente: `$pesoTF`, simboliza a importância do tf no cálculo e `$pesoTamanhoDoc`, que simboliza o tamanho dos documentos no cálculo do peso.

Assim, o Código-fonte 23 itera em sua segunda parte pelos *tokens* encontrados na expressão de busca. Usando de estatísticas como df , tf , idf , número de termos e total de documentos no *corpus*, é possível se calcular o numerador e o denominador da Equação 14, definida na seção 2.7.3. Por fim, a pontuação de cada documento vai incrementando (linha 50) de acordo com sua pontuação obtida a cada iteração de termos, resultante do produto do idf pela razão entre numerador e denominador (linha 47), o que de fato é a implementação da Equação 14.

Como abordado antes, a função de busca do modelo Probabilístico é portanto definida como o Código-fonte 24, simplesmente se responsabilizando pela chamada do método *bm25()* e ordenação decrescente dos resultados obtidos por ele.

Código-fonte 24 – Método de busca do modelo Probabilístico

```

1  public function buscar() {
2      $this->resultado = $this->bm25();
3      arsort($this->resultado);
4
5      return $this->resultado;
6  }

```

Fonte – Elaborado pelo autor.

3.5 Avaliação experimental

A avaliação experimental realizada neste trabalho consiste na análise dos resultados obtidos pelo SRI implementado, composto pelos três modelos clássicos de Recuperação da Informação. Esta variedade de métodos permite a comparação do desempenho de cada modelo utilizando 12 expressões de busca em um *corpus* formado por 200 artigos científicos (ver Apêndice A para mais detalhes sobre os artigos) publicados em periódicos de renome na área da Ciência da Computação e afins, coletados junto ao portal de publicações da Sociedade Brasileira de Computação (SBC)²⁶.

Nesta seção serão abordados aspectos sobre a coleta dos documentos e sua construção, com objetivo de formar uma coleção de referência para este estudo. Além disso, esta seção ainda aborda o processo de experimentação e construção das curvas de cobertura e precisão média interpolada em 11 pontos, como visto na seção 2.8.1.

3.5.1 Coleção de referência

²⁶ Mais informações sobre as publicações da SBC em: <http://www.sbc.org.br/publicacoes-2>.

Neste estudo optou-se pela pesquisa em um *corpus* composto por documentos em formato PDF, o que demanda a sua conversão pela tecnologia OCR (ver seção 3.2.4.2.4), necessária ao processo de indexação visto na seção 2.5. Para a composição do *corpus* de busca, decidiu-se pela utilização de artigos científicos publicados e indexados pela SBC (ver Apêndice A), com tamanho médio de aproximadamente 5 a 15 páginas, somados aos 24 artigos publicados até a presente data pela ForScience – Revista Científica do IFMG-Câmpus Formiga.

As publicações sob responsabilidade da SBC que divulgam e mantêm os artigos utilizados no *corpus* definido neste trabalho são:

- Biblioteca Digital Brasileira de Computação (BDBComp): disponibiliza e permite o acesso aos artigos publicados nos eventos promovidos pela SBC, como *workshops*, simpósios, encontros, conferências de todas as áreas contempladas pela Ciência da Computação. Dentre os principais periódicos indexados estão o *Journal of Computer Science* (INFOCOMP), *Journal of the Brazilian Computer Society* (JBACS), Revista de Informática Teórica e Aplicada (RITA), Informática Aplicada (IP), Revista Brasileira de Informática na Educação (RBIE) e a Revista de Redes de Computadores e Sistemas Distribuídos (RB-RES). Mais detalhes sobre cada periódico citado podem ser obtidos diretamente na BDBComp, no endereço: <http://www.lbd.dcc.ufmg.br/bdbcomp/servlet/ListaPeriodicos>.
- Periódico Computação Brasil: aborda temas relacionados à Tecnologia da Informação, Ensino Superior em Computação e Informática, Sociedade da Informação e políticas públicas para o setor de Informática. Mais informações em: <http://www.sbc.org.br/publicacoes-2/298-computacao-brasil>.
- Periódico *Journal of Integrated Circuits and Systems* (JICS): disponibiliza artigos sobre circuitos integrados e sistemas no estado da arte. Mais informações em: <http://www.sbmicro.org.br/jics/>.

Assim, verifica-se a diversidade de áreas de pesquisa e de temas utilizados para a composição do *corpus* deste estudo, ressaltando-se portanto a aleatoriedade na escolha dos documentos, tendo somente em comum sua relação com a área de Computação ou Informática.

Após a construção da coleção contendo os 200 documentos em formato PDF o próximo passo foi a realização do julgamento da relevância destes documentos a determinadas

expressões de busca. Para tanto, foram definidas 12 expressões de busca, apresentadas no Quadro 10.

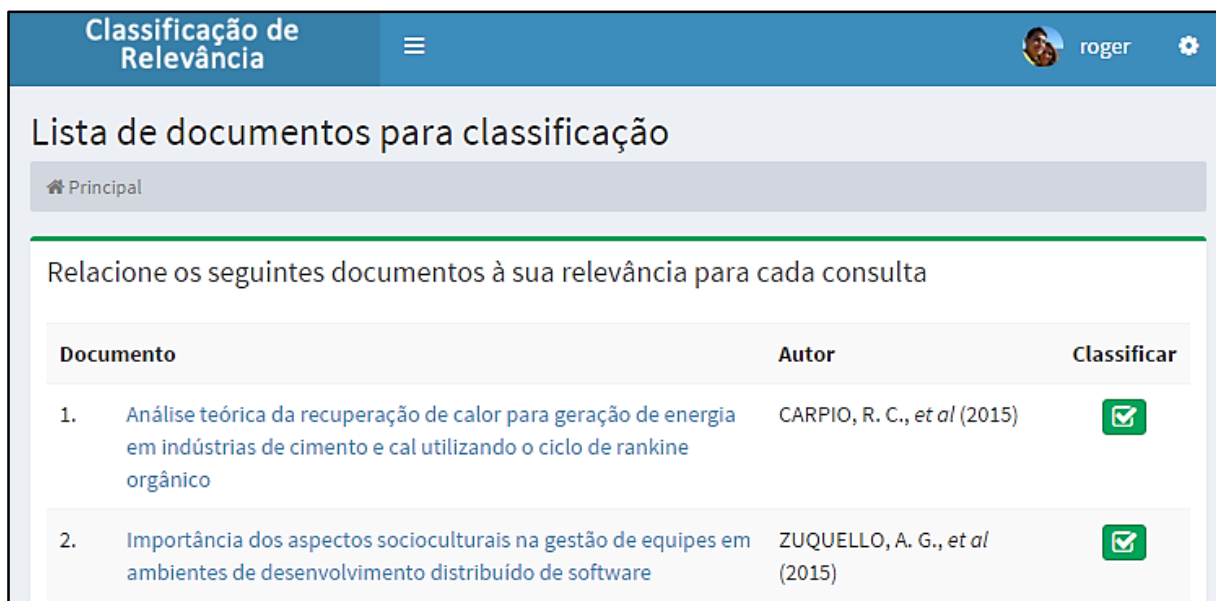
Quadro 10 – Expressões de busca definidas para o julgamento de relevância do corpus

CONSULTA	EXPRESSÃO DE BUSCA	OBJETO ESPERADO NESTA CONSULTA
cons1	software, sistemas, aplicação	ferramenta, produto de <i>software</i> ou aplicação
cons2	aprendizagem, ensino, conhecimento	educação, ensino, aprendizagem e afins
cons3	algoritmo, digital, imagem	algoritmos de processamento de imagens
cons4	saúde, análise, medicina	temas relacionados à medicina e saúde
cons5	redes, internet, web	tema de redes, computadores e afins
cons6	heurística, problema, busca	pesquisa operacional e otimização combinatória
cons7	dados, transação, mysql	tema relacionado a bancos de dados
cons8	projeto, síntese, eletrônica	ênfase no projeto de dispositivos, hardware configurável e afins
cons9	engenharia, projetos, metodologia	engenharia e desenvolvimento de <i>software</i>
cons10	objeto, procedimento, módulo	linguagens de programação
cons11	numérico, análise, modelo	métodos numéricos, análise de dados e modelagem matemática
cons12	gestão, administração, gerenciamento	gerência de processos, projetos ou produtos

Fonte – Elaborado pelo autor.

A fim de se realizar a etapa de julgamento da relevância de cada um dos documentos do Apêndice A às expressões de busca definidas no Quadro 10, foram convidados 03 (três) profissionais da área da Computação para classificarem e julgarem o *corpus*, formando assim a equipe de especialistas (ver seção 2.8). Para não haver viés no processo de escolha criou-se uma plataforma *web* (FIGURA 29) com controle de acesso por usuário e hospedada gratuitamente no *Platform-as-a-Service* da RedHat na *web*, chamado Openshift. Assim cada especialista realizou sua classificação de relevância de cada documento do *corpus* às cegas, sem interferência da classificação dos demais especialistas.

Figura 29 – Interface web para julgamento de relevância do corpus



Classificação de Relevância

Principal

Relacione os seguintes documentos à sua relevância para cada consulta

Documento	Autor	Classificar
1. Análise teórica da recuperação de calor para geração de energia em indústrias de cimento e cal utilizando o ciclo de rankine orgânico	CARPIO, R. C., et al (2015)	<input checked="" type="checkbox"/>
2. Importância dos aspectos socioculturais na gestão de equipes em ambientes de desenvolvimento distribuído de software	ZUQUELLO, A. G., et al (2015)	<input checked="" type="checkbox"/>

Fonte – Elaborada pelo autor.

A Figura 29 apresenta apenas um exemplo com o topo de uma lista contendo os 200 artigos, seu autor e um botão de classificação que ao ser pressionado inicializa a janela modal apresentada na Figura 30 para a ação de classificação do especialista. O nome dos artigos, em azul, é um *link* para *download* dos mesmos em formato PDF.

Figura 30 – Janela modal de classificação dos artigos

Classificar a relevância do documento às seguintes consultas: ×

Consulta	Descrição	Relevante?
software, sistemas, aplicação	aborda uma ferramenta, produto de software ou aplicação	<input type="checkbox"/>
aprendizagem, ensino, conhecimento	aborda educação, ensino, aprendizagem e afins	<input checked="" type="checkbox"/>
algoritmo, digital, imagem	aborda algoritmos de processamento de imagens	<input type="checkbox"/>
saúde, análise, medicina	aborda o processamento de informações médicas	<input type="checkbox"/>
redes, internet, web	aborda o tema de redes, computadores e afins	<input checked="" type="checkbox"/>
heurística, problema, busca	aborda pesquisa operacional	<input type="checkbox"/>
dados, transação, mysql	aborda o tema de bancos de dados	<input type="checkbox"/>
projeto, síntese, eletrônica	artigo com ênfase no projeto de dispositivos, hardware configurável e afins	<input type="checkbox"/>
engenharia, projetos, metodologia	aborda Engenharia de software e desenvolvimento de software	<input type="checkbox"/>
objeto, procedimento, módulo	aborda linguagens de programação	<input type="checkbox"/>
numérico, análise, modelo	aborda métodos numéricos, análise de dados e modelagem matemática	<input checked="" type="checkbox"/>
gestão, administração, gerenciamento	aborda gerência de processos, projetos ou produtos	<input checked="" type="checkbox"/>

Classificar

Fonte – Elaborada pelo autor.

Devido ao grande volume de obras no *corpus* estimou-se que seriam necessários vários dias para realizar a tarefa de classificação pelos especialistas. Para facilitar tal tarefa optou-se pela criação da aplicação *web* apresentada, hospedada em <http://classificacao-relevancia.rhcloud.com/>. Em alguns casos o especialista julgou a relevância com base no título da obra, consultando o resumo/*abstract* quando necessário. Se ainda persistiu alguma dúvida, o especialista realizou uma leitura dinâmica no texto do documento. Após decidir sobre a relevância do documento o especialista realizou a marcação da sua relevância às expressões de busca definidas pelo Quadro 10, de acordo com o exposto na Figura 30.

Após findada a atividade de classificação/julgamento pelos especialistas convidados, a próxima etapa consistiu na análise do julgamento feito, que pode ser observada no Apêndice A deste estudo (última coluna da tabela). Nesta coluna podemos perceber a quais consultas listadas no Quadro 10 cada um dos documentos foi considerado relevante. Para a decisão de

relevância entre os três julgamentos considerou-se a seguinte regra: “são relevantes os documentos que obtiverem no mínimo dois votos para uma determinada consulta”. A aplicação da regra resultou na coleção de referência usada na análise experimental deste trabalho. Além do Apêndice A, a Tabela 3 a seguir apresenta o resultado completo obtido pelo julgamento da equipe de especialistas concernente às expressões de busca do Quadro 10.

Tabela 3 – Resultado do julgamento de relevância pela equipe de especialistas

CONSULTA	ID DE DOCUMENTOS RELEVANTES
cons1	7, 10, 12, 13, 23, 24, 25, 27, 33, 44, 45, 48, 51, 52, 53, 75, 79, 80, 82, 83, 85, 106, 117, 125, 128, 139, 153, 170, 172, 174, 176, 181, 188, 197
cons2	5, 6, 8, 10, 14, 15, 26, 30, 37, 52, 60, 65, 82, 88, 92, 97, 102, 105, 108, 111, 113, 116, 120, 122, 125, 129, 133, 135, 137, 139, 141, 143, 144, 145, 146, 148, 149, 151, 153, 154, 155, 156, 158, 159, 161, 163, 164, 165, 178, 188, 189, 190, 192, 193, 194, 195, 196, 197
cons3	101
cons4	17, 20, 25, 34, 36, 43, 44, 51, 70, 71, 75, 84, 89, 91, 96, 101, 104, 130, 172, 174
cons5	31, 33, 38, 39, 49, 58, 64, 73, 80, 86, 95, 99, 103, 110, 112, 119, 121, 128, 132, 134, 168
cons6	19, 55, 56, 71, 72, 126, 140
cons7	35, 67, 109, 115, 147, 177, 181
cons8	12, 21, 22, 23, 24, 29, 92, 124, 143, 156
cons9	2, 28, 41, 47, 61, 63, 76, 79, 90, 94, 138, 150, 167, 169, 173, 181, 182, 183, 185, 191, 197
cons10	23, 48, 60, 82, 85, 88, 146, 148, 155, 158
cons11	3, 4, 7, 9, 16, 17, 20, 27, 34, 40, 43, 46, 50, 53, 54, 55, 56, 57, 67, 68, 70, 71, 72, 73, 84, 86, 87, 91, 101, 104, 107, 109, 112, 114, 115, 118, 126, 127, 130, 131, 136, 140, 147, 154, 162, 176, 184, 198, 199
cons12	2, 57, 61, 76, 90, 93, 94, 139, 150, 152, 157, 186, 187, 191

Fonte – Elaborada pelo autor.

3.5.2 Experimentação

Com o *corpus* formado pela coleção de referência previamente julgada por especialistas, procedeu-se à análise experimental de cada consulta definida no Quadro 10. Para cada método repetiu-se a consulta por 03 (três) vezes a fim de se obter um tempo médio de busca no

experimento em questão. A Tabela 4 apresenta os resultados calculados para as medidas de cobertura e precisão nos três modelos implementados, bem como o número total de documentos relevantes recuperados (a), o número de documentos julgados relevantes pelos especialistas (b) e o número de documentos recuperados ao todo pelo modelo (c).

Tabela 4 – Resultado de cobertura e precisão para os três modelos clássicos implementados

CONSULTA	Nº REL. REC. (a)	Nº REL. (b)	Nº REC. (c)	COBERTURA	PRECISÃO
cons1	34	34	191	1	0,178010471
cons2	57	58	141	0,982758621	0,404255319
cons3	1	1	148	1	0,006756757
cons4	18	20	174	0,9	0,103448276
cons5	20	21	137	0,952380952	0,145985401
cons6	7	7	166	1	0,042168675
cons7	7	7	189	1	0,037037037
cons8	9	10	123	0,9	0,073170732
cons9	19	21	153	0,904761905	0,124183007
cons10	5	10	102	0,5	0,049019608
cons11	47	49	188	0,959183673	0,25
cons12	11	14	108	0,785714286	0,101851852

Fonte – Elaborada pelo autor.

Ressalta-se que os resultados apresentados na Tabela 4 demonstram as medidas de cobertura e precisão relacionadas aos três modelos de RI implementados, considerando-se portanto apenas a sua capacidade de recuperação e ignorando-se completamente aqui a classificação e ordenação das funções de busca inerentes a cada modelo de RI, que serão abordadas em mais detalhes a seguir.

Os resultados de cobertura e precisão entre os três modelos são idênticos devido à opção de desenvolvimento de se priorizar a cobertura dos documentos relevantes, usando para isso a função de seleção dos documentos no índice por ocorrência de qualquer um dos termos da expressão de busca (ver seção 3.4). Como exemplo podemos citar que para a expressão de busca *ifmg, hardware, programável*, o SRI implementado monta um *corpus* de documentos composto por todos os documentos que têm o termo *ifmg* ou aqueles que têm o termo *hardware* ou ainda os que possuem o termo *programável*. Assim, implicitamente a busca para formação do *corpus* em todos os modelos de RI implementados utiliza o conector lógico **OU** entre todos os termos da expressão de busca, maximizando assim a cobertura.

A cobertura, conforme definida na seção 2.8.1, é a razão do número de documentos relevantes recuperados pelo número de documentos considerado relevante pelos especialistas. A precisão, por sua vez, é a razão entre o número de documentos relevantes recuperado pelo número total de documentos recuperados.

Visto que as medidas de cobertura e precisão não são aplicáveis a resultados ordenados, conforme observado na seção 2.8.1, para utilizá-las faz-se necessária a interpolação em 11 pontos da precisão média da recuperação em um gráfico de cobertura e precisão. Para este procedimento utilizou-se um *software* aplicativo de planilhas eletrônicas que permitiu a construção da tabela apresentada no Apêndice B deste trabalho. Apenas para exemplificar a obtenção das medidas aqui citadas, seguem na Tabela 5 os primeiros dez documentos recuperados pelo modelo Vetorial para a **consulta 1** (ver Quadro 10), que podem ser vistos por completo no Apêndice B.

Tabela 5 – Primeiros dez documentos recuperados pelo modelo Vetorial para a consulta 1

ID DOCs REL. (a)	POS. REL. (b)	POS. REL. (c)	ID DOCs REC. ■ = relevante (d)	COBERTURA (e)	PRECISÃO para esta cobertura (f)
7	1		109	0	
10	2		88	0	
12	3	1	197	0,029411765	0,333333333
13	4		148	0,029411765	
23	5		126	0,029411765	
24	6		179	0,029411765	
25	7	2	170	0,058823529	0,285714286
27	8	3	153	0,088235294	0,375
33	9		62	0,088235294	
44	10		136	0,088235294	
...

Fonte – Elaborada pelo autor.

De acordo com a Tabela 5, com a lista dos documentos julgados relevantes pelos especialistas (a) em mãos, o cálculo da cobertura e precisão demonstrado consiste nos seguintes passos:

- marcar na coluna (c) uma sequência numérica indicando quando os documentos do resultado apresentado na coluna (d) estiverem dentre aqueles julgados relevantes pelos especialistas, representados pela coluna (a).
- apenas para facilitar o trabalho, marcar as células da coluna (d) quando os documentos forem relevantes, de acordo com a coluna (a).
- a cobertura é calculada pela razão entre a coluna (c), que representa a posição de relevância, pelo número total de documentos considerados relevantes, que neste exemplo é definido como 34. Entre um documento relevante e o próximo considerado relevante a cobertura se mantém constante como o último valor obtido.
- a precisão é calculada pela razão entre a coluna (c), que é a posição de relevância, e a coluna (b), representando a posição na função de *ranking* do modelo implementado, sendo nula quando não se tem um documento relevante.

A título de exemplo, o resultado final para o cálculo de precisão interpolada de todos os 191 documentos recuperados pelo método Vetorial para a expressão de busca cons1 (ver Tabela 4) pode ser visto na Tabela 6 a seguir.

Tabela 6 – Resultado final para o cálculo da precisão média interpolada em 11 pontos

Nível de Cobertura (%)	Precisão interpolada
0	0,375
0,1	0,363636364
0,2	0,28
0,3	0,239130435
0,4	0,206896552
0,5	0,206896552
0,6	0,205357143
0,7	0,2
0,8	0,198757764
0,9	0,198757764
1	0,183783784

Fonte – Elaborada pelo autor.

A construção e consequente interpretação da precisão média interpolada da Tabela 6 é feita da seguinte forma: após passado o nível inicial de 0% de cobertura, ao se cobrir os primeiros 10% de documentos relevantes, a precisão interpolada é o valor máximo encontrado

de precisão encontrada dali em diante (até a linha final dos 191 documentos recuperados deste exemplo), conforme visto na Tabela 5.f. Assim, o cálculo da precisão interpolada demonstrado aqui segue os passos apresentados pela seção 2.8.1. e foi realizado para todos os experimentos que compõem esta análise experimental.

4 RESULTADOS E DISCUSSÃO

Nesta seção são apresentados os passos para a coleta de resultados, construção do gráfico de cobertura e precisão média interpolada em 11 pontos, bem como a análise a título de exemplo para o primeiro experimento feito com a consulta 1 da Tabela 10. A análise do experimento 1 aqui abordada em detalhes deve ser utilizada como referência para entendimento dos demais experimentos das seções seguintes.

4.1 Consulta 1

Dada uma determinada expressão de busca submetida ao SRI, calculando-se uma tabela similar à Tabela 6 para cada modelo de RI implementado podemos obter valores passíveis de comparação pela curva de cobertura e precisão entre eles. Neste trabalho as 12 expressões de busca dispostas no Quadro 10 tiveram seus valores calculados conforme exemplifica a Tabela 6 em todos os três modelos de RI implementados, obtendo-se ao final algo similar à Tabela 7 e que gera por fim o gráfico de cobertura e precisão média interpolada em 11 pontos, conforme o Gráfico 4.

Tabela 7 – Dados para a construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos

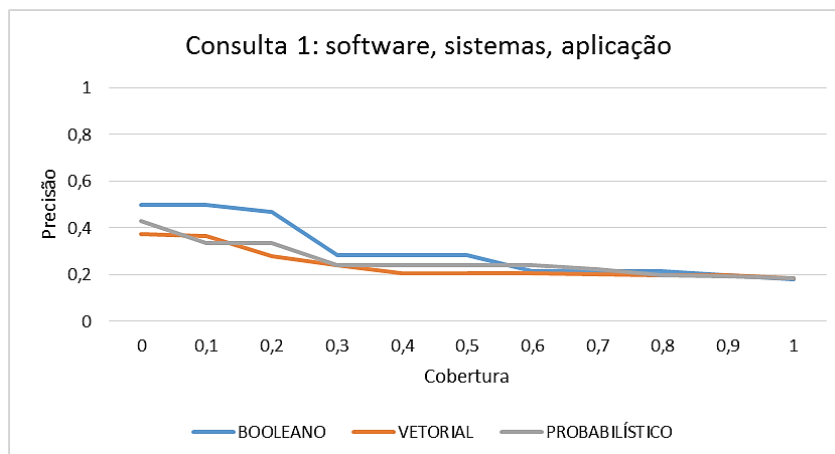
Cobertura (%)	precisão interpolada BOOLEANO	precisão interpolada VETORIAL	precisão interpolada PROBABILÍSTICO
0	0,5	0,375	0,428571429
0,1	0,5	0,363636364	0,333333333
0,2	0,466666667	0,28	0,333333333
0,3	0,283018868	0,239130435	0,23943662
0,4	0,283018868	0,206896552	0,23943662
0,5	0,283018868	0,206896552	0,23943662
0,6	0,212765957	0,205357143	0,238636364
0,7	0,212765957	0,2	0,224299065
0,8	0,212765957	0,198757764	0,195804196
0,9	0,198717949	0,198757764	0,194117647

Tabela 7 (continuação)

1	0,17989418	0,183783784	0,183783784
---	------------	-------------	-------------

Fonte – Elaborada pelo autor.

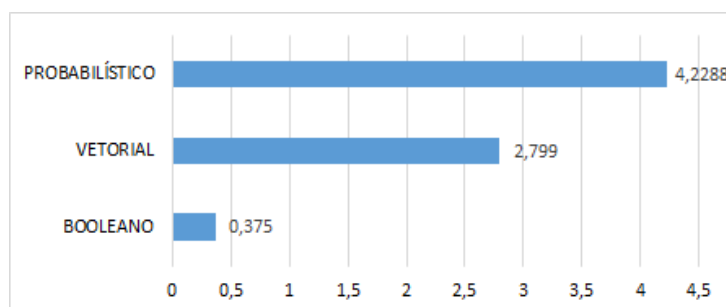
Gráfico 4 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 1



Fonte – Elaborado pelo autor.

Relembrando-se que a análise de gráficos como o demonstrado pelo Gráfico 4 é realizada observando-se a curva que mais se aproxima do canto superior direito, observamos que o modelo Booleano, mesmo sem uma função de *ranking*, ou seja, apresentando o resultado conforme a ordenação numérica que é recuperada diretamente do banco de dados, acabou, ao acaso, se saindo melhor que os outros métodos que têm toda a peculiaridade da classificação de relevância na entrega de seus resultados. Ainda segundo o Gráfico 4, observa-se que o modelo Probabilístico se saiu um pouco melhor na maior parte do experimento em comparação ao modelo Vetorial. Entretanto veremos a seguir que nem sempre o ocorrido neste primeiro exemplo entre os três modelos é verdadeiro.

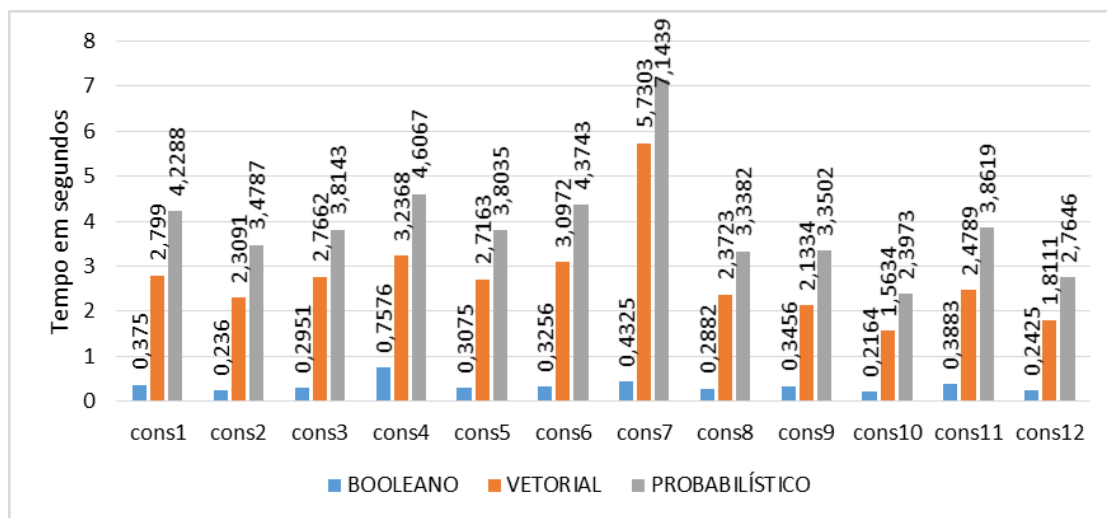
Gráfico 5 – Tempo de execução em segundos da consulta 1 nos três modelos de RI implementados



Fonte – Elaborado pelo autor.

Outra informação observada na análise experimental é o tempo de execução de consulta em segundos, conforme ilustra o Gráfico 5 para o exemplo da consulta 1 seguido nesta seção. Neste caso, apesar do modelo Probabilístico apresentar uma classificação de relevância superior em comparação ao modelo Vetorial, seu custo de tempo na execução da busca é o maior dentre os três modelos analisados.

Gráfico 6 – Tempo de busca aferido para cada consulta por cada modelo de RI



Fonte – Elaborado pelo autor.

O Gráfico 6 apresenta o tempo aferido para a realização da busca por cada consulta definida pelo Quadro 10 por cada um dos três modelos de RI, onde podemos observar um comportamento padrão entre os três modelos. O modelo Booleano, pela simplicidade da construção de seu modelo e até mesmo pelo uso da recursão²⁷, tem menores tempos para todos os experimentos realizados. O modelo Vetorial apresenta tempos medianos se comparado aos altos tempos observados no modelo Probabilístico.

4.2 Consulta 2

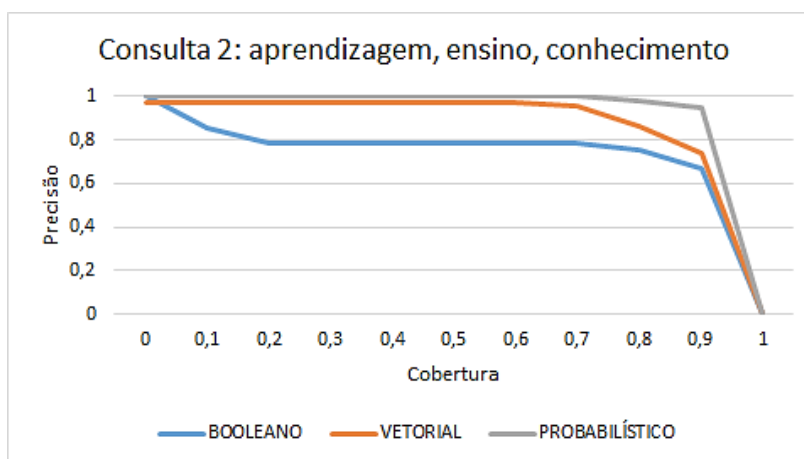
²⁷ A recursão pode utilizar o paralelismo em nível de *hardware* e *software* (GHIYA; HENDREN; ZHU, 2005; MIKHAILOV *et al.*, 2010).

Tabela 8 – Dados para construção comparativa entre três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 2

Cobertura (%)	BOOLEANO	VETORIAL	PROBABILÍSTICO
0	1	0,975	1
0,1	0,857142857	0,975	1
0,2	0,781818182	0,975	1
0,3	0,781818182	0,975	1
0,4	0,781818182	0,975	1
0,5	0,781818182	0,975	1
0,6	0,781818182	0,975	1
0,7	0,781818182	0,953488372	1
0,8	0,753623188	0,859649123	0,981132075
0,9	0,670886076	0,736842105	0,948275862
1	0	0	0

Fonte – Elaborada pelo autor.

Gráfico 7 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 2



Fonte – Elaborado pelo autor.

Observando-se o Gráfico 7 notadamente o desempenho do modelo Probabilístico se mostrou acima dos outros dois modelos, se mantendo com 100% de precisão até o nível de 70% da cobertura, ficando o modelo Vetorial como o segundo melhor e o modelo Booleano em última colocação. Há ainda a questão de que os modelos implementados não atingiram a cobertura de 100%, afinal dos 58 documentos julgados relevantes pelos especialistas, somente

57 foram recuperados, conseqüentemente causando a queda na precisão do Gráfico 7 entre os dois níveis finais de 90% e 100% de cobertura (TABELA 8).

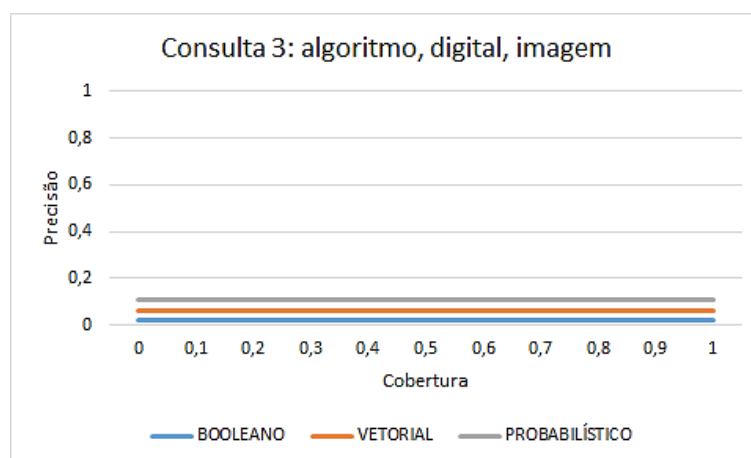
4.3 Consulta 3

Tabela 9 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 3

Cobertura (%)	BOOLEANO	VETORIAL	PROBABILÍSTICO
0	0,02222222	0,058823529	0,11111111
0,1	0,02222222	0,058823529	0,11111111
0,2	0,02222222	0,058823529	0,11111111
0,3	0,02222222	0,058823529	0,11111111
0,4	0,02222222	0,058823529	0,11111111
0,5	0,02222222	0,058823529	0,11111111
0,6	0,02222222	0,058823529	0,11111111
0,7	0,02222222	0,058823529	0,11111111
0,8	0,02222222	0,058823529	0,11111111
0,9	0,02222222	0,058823529	0,11111111
1	0,02222222	0,058823529	0,11111111

Fonte – Elaborada pelo autor.

Gráfico 8 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 3



Fonte – Elaborado pelo autor.

Com precisões baixas (TABELA 9), o experimento com a consulta 3 apresentou o mesmo resultado entre os três modelos de RI implementados no caso da consulta 2. Neste caso, todos obtiveram uma curva constante de precisão próxima ao valor 0, o que é justificado pelo julgamento do *corpus* para essa consulta apontar apenas um documento como relevante, fazendo assim com que a comparação entre os modelos seja justamente a posição na qual esse determinado documento foi classificado (GRÁFICO 8). No caso do modelo Booleano, o documento foi retornado na 45ª posição, na 17ª posição no modelo Vetorial e na 9ª posição no modelo Probabilístico.

Para este experimento, de fato a cobertura de 100% foi alcançada por todos os modelos implementados, ou seja, recuperaram um documento de um total de um documento julgado como relevante.

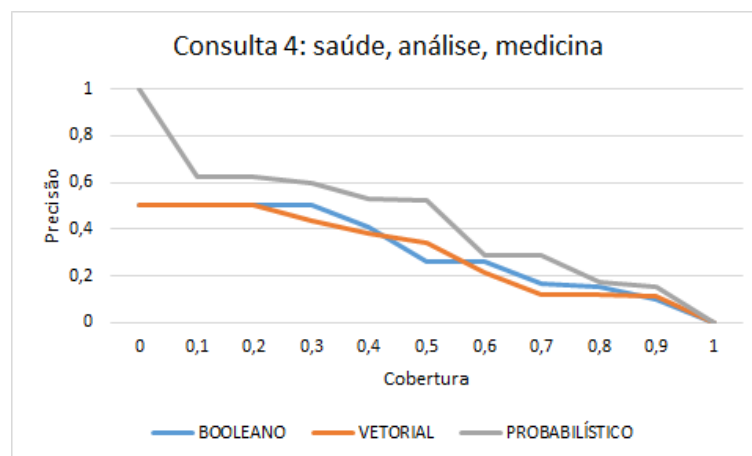
4.4 Consulta 4

Tabela 10 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 4

Cobertura (%)	BOOLEANO	VETORIAL	PROBABILÍSTICO
0	0,5	0,5	1
0,1	0,5	0,5	0,625
0,2	0,5	0,5	0,625
0,3	0,5	0,4375	0,6
0,4	0,409090909	0,380952381	0,533333333
0,5	0,260869565	0,344827586	0,526315789
0,6	0,260869565	0,214285714	0,288888889
0,7	0,166666667	0,11971831	0,285714286
0,8	0,152380952	0,11971831	0,173469388
0,9	0,103448276	0,115384615	0,151260504
1	0	0	0

Fonte – Elaborada pelo autor.

Gráfico 9 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 4



Fonte – Elaborado pelo autor.

No experimento com a consulta 4 novamente observamos um melhor desempenho do modelo Probabilístico. O que chama a atenção no resultado do Gráfico 9 é a comparação entre o modelo Booleano e o modelo Vetorial, sendo que por vários níveis de cobertura o modelo Booleano apresentou melhores resultados que o modelo Vetorial. Isto se apresenta como uma surpresa, afinal o modelo Booleano não classifica os resultados, apenas os apresenta pela ordem numérica de ID dos documentos persistidos pela aplicação. Podemos atribuir este atípico resultado ao tamanho do *corpus*, que no caso de um *corpus* com milhares de documentos poderia apresentar uma variação maior no resultado aqui visto pelo Gráfico 9 e a Tabela 10.

4.5 Consulta 5

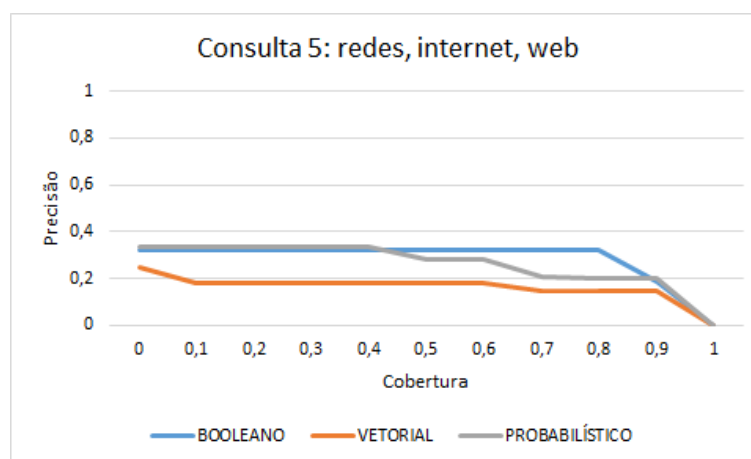
Tabela 11 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 5

Cobertura (%)	BOOLEANO	VETORIAL	PROBABILÍSTICO
0	0,320754717	0,25	0,333333333
0,1	0,320754717	0,181818182	0,333333333
0,2	0,320754717	0,181818182	0,333333333
0,3	0,320754717	0,181818182	0,333333333
0,4	0,320754717	0,181818182	0,333333333
0,5	0,320754717	0,181818182	0,282608696

0,6	0,320754717	0,181818182	0,282608696
0,7	0,320754717	0,149253731	0,205479452
0,8	0,320754717	0,149253731	0,202380952
0,9	0,188118812	0,149253731	0,20212766
1	0	0	0

Fonte – Elaborada pelo autor.

Gráfico 10 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 5



Fonte – Elaborado pelo autor.

De acordo com o Gráfico 10 e os resultados da Tabela 11 podemos entender que novamente ao acaso, o modelo Booleano, cujo resultado de classificação é meramente ordenado de acordo com o número de identificação de cada documento recuperado no banco de dados, se destacou como o modelo de 2º melhor desempenho. Observa-se ainda que os modelos de RI implementados não atingiram uma precisão satisfatória no nível de cobertura de 100% devido ao fato de recuperarem apenas 20 dos 21 documentos julgados relevantes pelos especialistas.

Quanto à recuperação nos modelos com função de *ranking*, novamente o modelo Probabilístico obteve um melhor desempenho em todo o experimento se comparado ao modelo Vetorial, que neste caso ficou como o pior desempenho dentre os três modelos analisados.

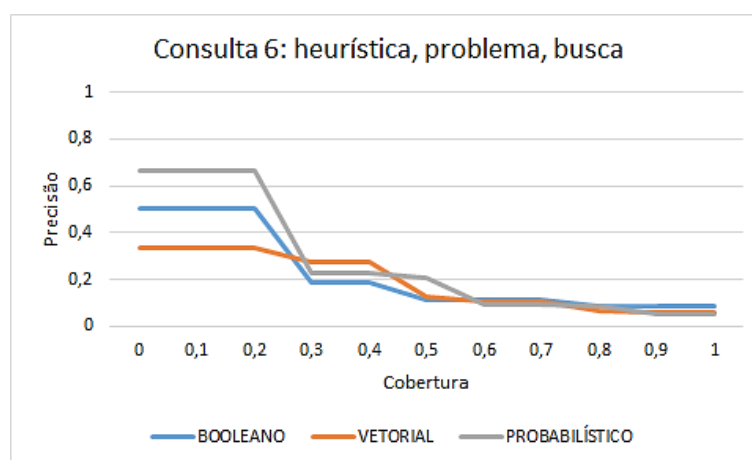
4.6 Consulta 6

Tabela 12 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 6

Cobertura (%)	BOOLEANO	VETORIAL	PROBABILÍSTICO
0	0,5	0,333333333	0,666666667
0,1	0,5	0,333333333	0,666666667
0,2	0,5	0,333333333	0,666666667
0,3	0,1875	0,272727273	0,230769231
0,4	0,1875	0,272727273	0,230769231
0,5	0,11627907	0,125	0,210526316
0,6	0,11627907	0,108695652	0,094339623
0,7	0,11627907	0,108695652	0,094339623
0,8	0,0875	0,065934066	0,085714286
0,9	0,0875	0,05982906	0,0546875
1	0,0875	0,05982906	0,0546875

Fonte – Elaborada pelo autor.

Gráfico 11 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 6



Fonte – Elaborado pelo autor.

Apesar de que até o nível de cobertura de 30% o modelo Probabilístico tenha se destacado com melhor desempenho, a partir deste nível o experimento levou à tendência de precisões similares em todos os modelos, principalmente após o nível de 60% de cobertura (Gráfico 11 e Tabela 12). Para este experimento observa-se que todos os modelos de RI implementados obtiveram cobertura de 100%, conseguindo recuperar todos os sete documentos julgados relevantes.

Neste experimento o modelo Booleano resultou um melhor desempenho final, seguido do modelo Vetorial e por último o modelo Probabilístico.

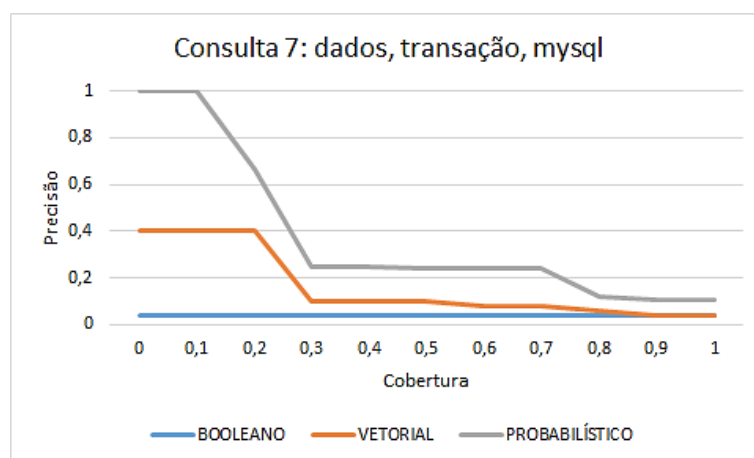
4.7 Consulta 7

Tabela 13 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 7

Cobertura (%)	BOOLEANO	VETORIAL	PROBABILÍSTICO
0	0,040935673	0,4	1
0,1	0,040935673	0,4	1
0,2	0,040935673	0,4	0,666666667
0,3	0,040935673	0,102564103	0,25
0,4	0,040935673	0,102564103	0,25
0,5	0,040935673	0,102564103	0,238095238
0,6	0,040935673	0,081967213	0,238095238
0,7	0,040935673	0,081967213	0,238095238
0,8	0,040935673	0,058823529	0,117647059
0,9	0,040935673	0,037634409	0,109375
1	0,040935673	0,037634409	0,109375

Fonte – Elaborada pelo autor.

Gráfico 12 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 7



Fonte – Elaborado pelo autor.

Dentre todos os experimentos realizados, este é o resultado que melhor demonstra as conclusões obtidas da análise. Conforme se vê no Gráfico 12 e na Tabela 13, há um claro melhor desempenho do modelo Probabilístico, seguido pelo 2º melhor desempenho do modelo Vetorial e por fim o modelo Booleano. Neste experimento podemos observar a diferença que a função de *ranking* exerce sobre os resultados entre modelos com função de classificação e aqueles que não a utilizam. Apesar de atingir o nível de cobertura de 100%, todos os modelos de RI chegam a este nível com uma precisão muito baixa, o que significa que muitos documentos não relevantes foram recuperados juntos àqueles julgados relevantes pelos especialistas.

Observa-se ainda que a provável justificativa para o resultado ter se aproximado do esperado neste experimento é devido à especificidade dos termos da expressão de busca, sendo que dentre os três apenas o termo *dados* é genérico se analisado em separado. Os outros, *transação* e *mysql* são mais específicos e ligados fortemente à área de bancos de dados.

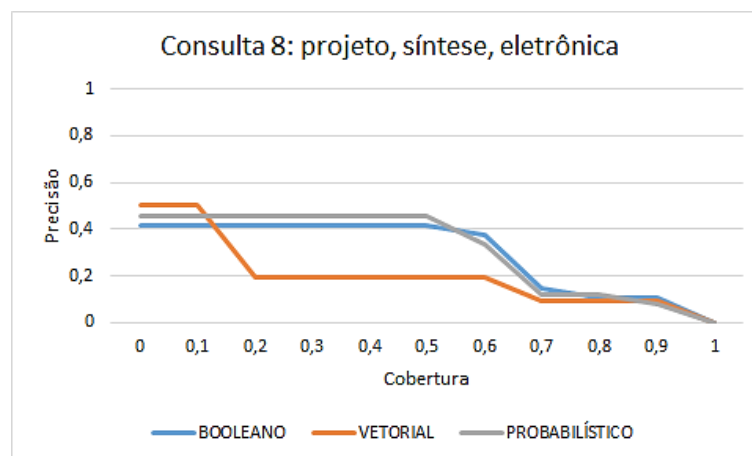
4.8 Consulta 8

Tabela 14 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 8

Cobertura (%)	BOOLEANO	VETORIAL	PROBABILÍSTICO
0	0,416666667	0,5	0,454545455
0,1	0,416666667	0,5	0,454545455
0,2	0,416666667	0,193548387	0,454545455
0,3	0,416666667	0,193548387	0,454545455
0,4	0,416666667	0,193548387	0,454545455
0,5	0,416666667	0,193548387	0,454545455
0,6	0,375	0,193548387	0,333333333
0,7	0,14893617	0,095744681	0,119402985
0,8	0,109756098	0,095744681	0,119402985
0,9	0,109756098	0,095744681	0,076923077
1	0	0	0

Fonte – Elaborada pelo autor.

Gráfico 13 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 8



Fonte – Elaborado pelo autor.

Por mais estranho que pareça, observando o Gráfico 13 e a Tabela 14 podemos notar que o modelo Probabilístico obteve desempenho praticamente semelhante ao modelo Booleano. Para este resultado não esperado podemos atribuir o mero acaso ocorrido por razão do uso de um *corpus* muito pequeno e limitado. Neste experimento o modelo que obteve um desempenho inferior foi o Vetorial. Todos os modelos de RI experimentados chegaram a recuperar apenas 9 dos 10 julgados como relevantes.

4.9 Consulta 9

Tabela 15 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 9

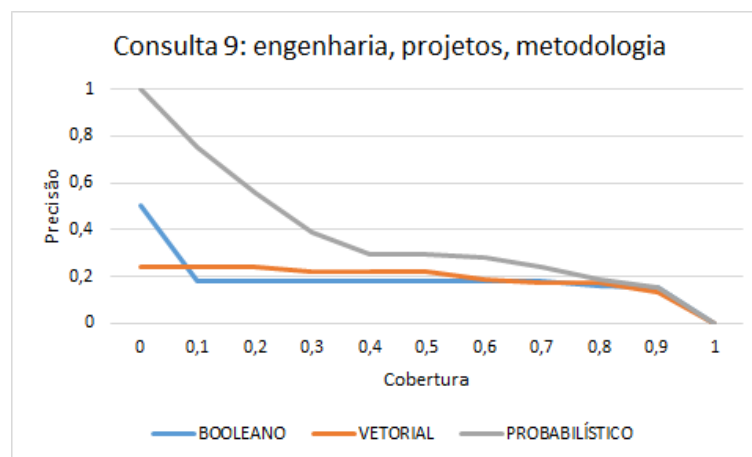
Cobertura (%)	BOOLEANO	VETORIAL	PROBABILÍSTICO
0	0,5	0,24	1
0,1	0,184210526	0,24	0,75
0,2	0,184210526	0,24	0,555555556
0,3	0,184210526	0,218181818	0,388888889
0,4	0,182926829	0,218181818	0,297297297
0,5	0,182926829	0,218181818	0,297297297
0,6	0,182926829	0,186666667	0,282608696
0,7	0,182926829	0,177083333	0,238095238

Tabela 15 (continuação)

0,8	0,157894737	0,177083333	0,184782609
0,9	0,155737705	0,136690647	0,150793651
1	0	0	0

Fonte – Elaborada pelo autor.

Gráfico 14 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 9



Fonte – Elaborado pelo autor.

Conforme se vê no Gráfico 14, uma vez mais o modelo Probabilístico obteve desempenho superior aos demais. No caso do modelo Vetorial, com exceção do primeiro nível e do nível de 70% de cobertura, obteve resultados pouco melhores que o modelo Booleano. Novamente, em 100% de cobertura todos os modelos finalizaram o experimento com precisão em 0 (TABELA 15), o que demonstra que não foram recuperados todos os documentos julgados relevantes, afinal apenas 21 dos 23 documentos julgados relevantes foram recuperados.

4.10 Consulta 10

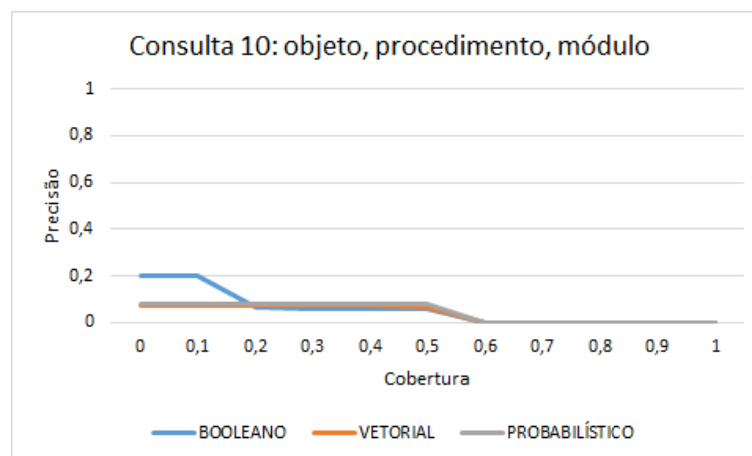
Tabela 16 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 10

Cobertura (%)	BOOLEANO	VETORIAL	PROBABILÍSTICO
0	0,2	0,075471698	0,079365079
0,1	0,2	0,075471698	0,079365079
0,2	0,064516129	0,075471698	0,079365079

0,3	0,05952381	0,075471698	0,079365079
0,4	0,05952381	0,075471698	0,079365079
0,5	0,05952381	0,064102564	0,079365079
0,6	0	0	0
0,7	0	0	0
0,8	0	0	0
0,9	0	0	0
1	0	0	0

Fonte – Elaborada pelo autor.

Gráfico 15 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 10



Fonte – Elaborado pelo autor.

Provavelmente um dos piores resultados obtidos nos experimentos deste trabalho, a análise dos resultados da consulta 10 (Gráfico 15 e Tabela 16) apresentou novamente algo inesperado até o nível de 20% de cobertura dos documentos relevantes: o modelo Booleano se sobressaiu em desempenho de precisão aos demais modelos. A partir deste nível de cobertura, todos os modelos obtiveram uma precisão similar, sendo o Probabilístico levemente melhor que os demais. Após o nível de cobertura de 60% todos os modelos de RI implementados tiveram sua precisão igualada a 0 devido ao fato de recuperarem apenas 5 dos 10 documentos julgados relevantes na coleção para esta expressão de busca.

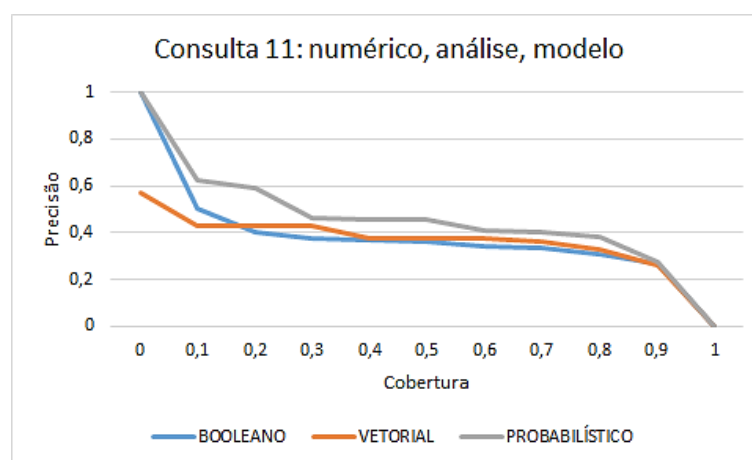
4.11 Consulta 11

Tabela 17 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 11 pontos da consulta 11

Cobertura (%)	BOOLEANO	VETORIAL	PROBABILÍSTICO
0	1	0,571428571	1
0,1	0,5	0,432432432	0,625
0,2	0,4	0,432432432	0,588235294
0,3	0,37254902	0,432432432	0,463414634
0,4	0,369230769	0,376470588	0,454545455
0,5	0,36	0,376470588	0,453125
0,6	0,343434343	0,376470588	0,407407407
0,7	0,336538462	0,36	0,4
0,8	0,310077519	0,325396825	0,384615385
0,9	0,266272189	0,260115607	0,277777778
1	0	0	0

Fonte – Elaborada pelo autor.

Gráfico 16 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 11



Fonte – Elaborado pelo autor.

Com uma precisão um pouco melhor que nos outros experimentos, os resultados experimentados pela consulta 11 se mantiveram dentro do esperado entre 10% e 90% dos níveis de cobertura (Gráfico 16 e Tabela 17). O modelo Probabilístico apresentou melhor precisão, seguido pela precisão do modelo Vetorial e, por fim, a precisão do modelo Booleano. Todos os modelos finalizaram o experimento no nível de cobertura de 100% com a precisão em valor igual a 0, devido ao fato de recuperarem apenas 47 dos 49 documentos julgados relevantes.

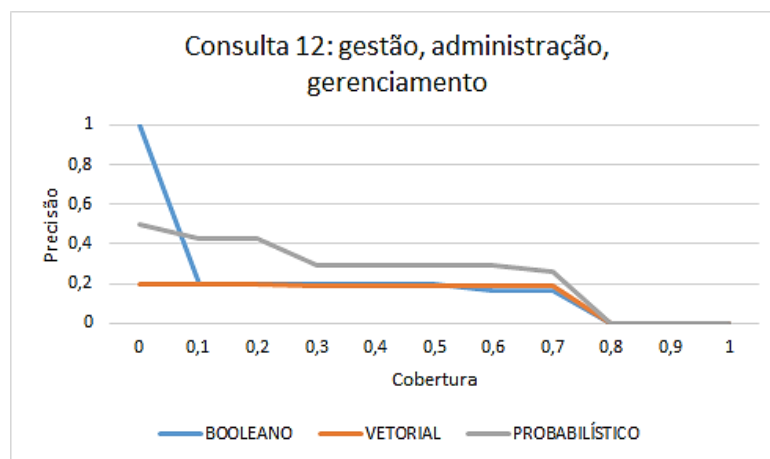
4.12 Consulta 12

Tabela 18 – Dados para construção comparativa entre os três modelos de RI do gráfico de cobertura e precisão média interpolada em 12 pontos da consulta 11

Cobertura (%)	BOOLEANO	VETORIAL	PROBABILÍSTICO
0	1	0,2	0,5
0,1	0,2	0,2	0,428571429
0,2	0,2	0,2	0,428571429
0,3	0,2	0,192307692	0,294117647
0,4	0,2	0,192307692	0,291666667
0,5	0,2	0,192307692	0,290322581
0,6	0,166666667	0,192307692	0,290322581
0,7	0,166666667	0,192307692	0,263157895
0,8	0	0	0
0,9	0	0	0
1	0	0	0

Fonte – Elaborada pelo autor.

Gráfico 17 – Curvas de cobertura e precisão média interpolada em 11 pontos para os três modelos na consulta 12



Fonte – Elaborado pelo autor.

A análise do experimento com a consulta 12 pôde oferecer pelo Gráfico17 a superioridade de precisão obtida pelo modelo Probabilístico uma vez mais. Neste experimento os modelos Vetorial e Booleano apresentaram comportamento praticamente similar, com exceção apenas do primeiro nível de cobertura, onde o modelo Booleano claramente apresentou

resultado melhor que os outros dois. No nível de cobertura de documentos relevantes de 80%, todos os modelos de RI tiveram sua precisão igualada a 0 devido à recuperação de apenas 11 dos 14 documentos julgados relevantes no *corpus* de busca.

4.13 Resumo comparativo

Pela análise das informações apresentadas até aqui nesta seção obtém-se a conclusão de que o modelo Probabilístico na maior parte dos experimentos se mostrou superior aos demais com relação à precisão de sua classificação, sendo que este se destacou em 10 dos 12 experimentos realizados. Visualmente o modelo que obteve o segundo melhor resultado foi o modelo Vetorial, se sobressaindo parcialmente sob o modelo Booleano em 7 dos 12 experimentos e, por último, o modelo Booleano, ao contrário do que se viu no exemplo da consulta 1. Embora o modelo Probabilístico tenha se destacado na maior parte dos experimentos com relação à cobertura e precisão dos resultados de sua função de *ranking*, o mesmo tem o maior custo de tempo (GRÁFICO 6), ficando assim sujeito a testes mais elaborados em *corpus* maiores, afinal uma questão de grande destaque no estudo de implantação de SRIs e o modelo de RI adotado é a sua escalabilidade.

Para melhor embasar as conclusões tiradas dos experimentos realizados, a Tabela 19 apresenta a posição em qualidade de precisão obtida pelo experimento com cada modelo de RI em cada consulta. Para a obtenção destes dados de posicionamento no resultado foram contabilizados os maiores valores aferidos, bem como os segundo maiores apresentados nas Tabelas 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 e 18.

Tabela 19 – Resumo comparativo de desempenho da precisão interpolada dos modelos de RI implementados, considerando-se total de posições ocupado

Consulta	BOOLEANO	VETORIAL	PROBABILÍSTICO
cons1	1°	3°	2°
cons2	3°	2°	1°
cons3	3°	2°	1°
cons4	2°	3°	1°
cons5	2°	3°	1°

Tabela 19 (continuação)

cons6	1°	2°	3°
cons7	3°	2°	1°
cons8	2°	3°	1°
cons9	3°	2°	1°
cons10	3°	2°	1°
cons11	3°	2°	1°
cons12	2°	3°	1°
TOTAL	2 x 1° 4 x 2° 6 x 3°	0 x 1° 7 x 2° 5 x 3°	10 x 1° 1 x 2° 1 x 3°
RESULTADO	3°	2°	1°

Fonte – Elaborada pelo autor.

Assim, e de acordo com a totalização demonstrada pela Tabela 19, conclui-se que para o *dataset* utilizado o modelo Probabilístico apresentou melhor desempenho quanto à precisão dos resultados. Em segundo lugar, o modelo Vetorial, se aproximando variadas vezes do modelo Booleano, o qual de forma completamente ao acaso em alguns experimentos apresentou melhores resultados que o modelo Vetorial e, por fim, o modelo Booleano, que de maneira esperada apresentou um desempenho abaixo dos outros modelos com função de *ranking*.

5 CONSIDERAÇÕES FINAIS

Este estudo constitui-se como uma busca pela compreensão dos fundamentos e ideias relacionados à construção de Sistemas de Recuperação da Informação, composto principalmente pelo processo de indexação de documentos, pelo processo de especificação de consulta, e pelos modelos de Recuperação da Informação, aqui optados como modelo Booleano, Vetorial e Probabilístico. De acordo com o trabalho apresentado, a compreensão de tais sistemas, junto às peculiaridades de seus vários modelos de Recuperação da Informação o tornam um assunto complexo e de difícil abstração.

A implementação de um Sistema de Recuperação da Informação se mostrou como uma tarefa complexa e abrangente. Conforme demonstrado por todo o referencial teórico disposto na seção 2 deste trabalho, a informação que se deve abstrair antes do início da implementação de um SRI se mostrou bem exigente. Dentre as atividades desempenhadas durante o desenvolvimento, a que se mostrou mais trabalhosa sem dúvidas é o processo de indexação junto ao processo de especificação de consulta, afinal envolve o uso de serviços terceiros como *webservices* OCR, o que de fato não é uma atividade trivial de se desenvolver apenas para uso singular em um sistema de informação. Por ser o fundamento de funcionamento de um SRI, a correta criação e manutenção de um índice é uma atividade meticulosa e interfere diretamente na qualidade da recuperação, seja no quesito de tempo de busca ou cobertura e precisão.

Quanto ao desenvolvimento dos modelos clássicos de Recuperação da Informação, o modelo Booleano se mostrou como o mais custoso para implementação. A análise de seu código-fonte apresenta o uso da recursão como meio de construção e caminhamento na árvore de busca, atividade definida como a função de classificação do modelo de RI, processo que é mais simples nos outros modelos desenvolvidos. Entre tais modelos o que se mostrou mais elegante é o modelo Vetorial, pela construção de um subespaço vetorial e apresentação de documentos e consultas como vetores, o que de fato, torna-o um modelo prático e de recursos bem definidos, derivados da teoria dos vetores da Geometria Analítica. O modelo Probabilístico é o modelo mais eficiente, talvez pelo uso de uma metodologia mais recente que os outros dois modelos, afinal não implementou-se o modelo clássico Probabilístico mas sim um modelo derivado não assistido pelo usuário, neste caso o modelo OKAPI/BM25.

Um detalhe importante a ser concluído de todo este trabalho é a busca por relevância e não uma busca por casamento exato como é normalmente visto em SIs em geral, e que de fato

se resume a uma busca booleana em bancos de dados. Sendo a relevância algo subjetivo e inerente ao julgamento próprio do usuário, o grande desafio dos estudiosos da área de Recuperação da Informação é conseguir atingir o máximo de cobertura e precisão em ambientes controlados, resumindo seu objetivo em: a) cobrir todos os documentos relevantes à expressão de busca do usuário, e; b) atingir uma precisão próxima a 100%, reduzindo ao máximo a recuperação de documentos não relevantes à expressão de busca.

Ainda como complemento ao estudo foram analisados e apresentados os resultados da experimentação realizada com 12 expressões de busca em uma coleção de 200 artigos científicos da área da Ciência da Computação e afins. Observou-se que a análise experimental para *corpus* pequenos e bem delimitados não apresenta resultados bem definidos, afinal em alguns experimentos o modelo Booleano, conhecidamente de desempenho inferior (BAEZA-YATES; RIBEIRO-NETO, 1999, p. 26), até mesmo se sobressaiu aos outros modelos analisados. Mesmo assim, nos experimentos realizados o modelo Probabilístico obteve melhores resultados que o Vetorial e Booleano.

A oportunidade de se trabalhar a aplicação de um *framework* moderno de desenvolvimento *web* se mostrou como um dos grandes trunfos no desenvolvimento deste trabalho, acelerando a produção, otimizando atividades rotineiras e acrescentando muitos conhecimentos e conceitos que, aplicados neste trabalho, tornam a área de Recuperação da Informação e o desenvolvimento de aplicações em ambiente *web* cada vez mais atraentes aos estudiosos da computação.

Conclui-se, por fim, que o estudo da área de Recuperação da Informação é de grande utilidade para a comunidade de sistemas de informações em geral. Por razão da explosão informacional gerada pelo número de documentos e usuários de SIs, os modelos para recuperação **precisa** da informação se tornaram valiosos no mercado atual, tanto profissional, quanto acadêmico sendo portanto necessário realizar mais pesquisas nesta área.

REFERÊNCIAS

- ÁVILA, Bruno T. **Avaliação de desempenho de sistemas de recuperação de informação**. 2014. Disponível em: <https://sites.google.com/site/renatocorrea/disciplinas/recuperacao-da-informacao/Aula06-Avalia%C3%A7%C3%A3o_de_Desempenho.pdf?attredirects=0&d=1>. Acesso em: 13 jan. 2016.
- BAEZA-YATES, Ricardo; RIBEIRO-NETO, Berthier. **Modern information retrieval**. [S.l.]: ACM Press, 1999.
- _____; _____. **Modern information retrieval: the concepts and technology behind search**. 2012. Slides de aula. cap. 3. Disponível em: <<http://grupoweb.upf.es/WRG/mir2ed/slides.php>>. Acesso em: 08 jan. 2016.
- _____; _____. **Modern information retrieval: the concepts and technology behind search**. 2012. Slides de aula. cap. 4. Disponível em: <<http://grupoweb.upf.es/WRG/mir2ed/slides.php>>. Acesso em: 10 jan. 2016.
- BARTH, Fabrício J. **Uma breve introdução ao tema Recuperação de Informação**. São Paulo: [s. n.], 2010. Disponível em: <<http://fbarth.net.br/materiais/introducaoRecuperacaoInformacao/introducaoRecuperacaoInformacao.pdf>>. Acesso em: 22 dez. 2015.
- _____. Uma introdução ao tema recuperação de informações textuais. **RITA**, v. 20, n. 2, 2013. Disponível em: <http://seer.ufrgs.br/index.php/rita/article/download/rita_v20_n2_p155WesleyVol20Nr2_247/25454>. Acesso em: 06 jan. 2016.
- BERNARDI, Raffaella. **Digital libraries: ranked evaluation**. Università Degli Studi di Trento: [s. n.], 2012. Slides de aula. Disponível em: <http://disi.unitn.it/~bernardi/Courses/DL/Slides_11_12/7.pdf>. Acesso em: 12 jan. 2016.
- BOULOS, Paulo; OLIVEIRA, Ivan. C. **Geometria analítica: um tratamento vetorial**. São Paulo: Mcgraw Hill, 1987.
- BÜTTCHER, Stefan; CLARKE, Charles L. A.; CORMACK, Gordon V. **Information retrieval: implementing and evaluating search engines**. Massachusetts, EUA: MIT Press, 2010. Disponível em: <<http://www.ir.uwaterloo.ca/book/>>. Acesso em: 02 jan. 2016.
- CARDOSO, Olinda, N. P. **Recuperação da informação**. Lavras: UFLA, DCC, 2000. Disponível em: <<http://www.dcc.ufla.br/infocomp/index.php/INFOCOMP/article/view/46/31>>. Acesso em: 05 jan. 2016.
- CIFERRI, Cristina D. de A. **Estruturas de indexação de dados**, 2013. Slides de aula. Disponível em: <<http://wiki.icmc.usp.br/images/d/d0/SCC578920131-indicesAp01.pdf>>. Acesso em: 13 jan. 2016.

COOPER, William S. **The formalism of probability theory in ir**: a foundation or an encumbrance? In: ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL (SIGIR'94), 17., 1994. New York. **Proceedings...** New York, NY, USA: Springer-Verlag, 1994. p. 242–247.

CORMEN, Thomas H. et al. **Algoritmos**: teoria e prática. 2. ed. Rio de Janeiro: Campus, 2002. 916 p. Disponível em: <<http://www.cin.ufpe.br/~ara/algoritmos-%20portugu%EA-%20cormen.pdf>>. Acesso em: 12 jan. 2016.

CRUZ, Carlos H. B. Vannevar Bush: uma apresentação. **Revista Latinoamericana de Psicopatologia Fundamental**, São Paulo, v. 14, n. 1, p. 11-13, 2011. Disponível em: <<http://www.scielo.br/pdf/rlpf/v14n1/01.pdf>>. Acesso em: 19 jan. 2016.

DAS, Gautam. **Databases and information retrieval**. 2005. Disponível em: <<http://ranger.uta.edu/~gdas/websitepages/spring05DBIR.htm>>. Acesso em: 02 jan. 2016.

DAVIS, William S.; YEN, David C. **The Information system consultant's handbook**: systems analysis and design. [S.l.]: CRC Press LLC, 1999. Disponível em: <http://www.msoffice.us/MIS/CRC%20Press%20-%20Information%20System%20Consultants%20Handbook%20Systems%20Analysis%20and%20Design/7001_PDF_TOC.pdf>. Acesso em: 02 jan. 2016.

DEERING, Sam. **5 different types of document ready examples**. 2011. Disponível em: <<http://www.sitepoint.com/types-document-ready/>>. Acesso em: 21 jan. 2016.

DICIONÁRIO INFORMAL. Disponível em: <<http://www.dicionarioinformal.com.br/>>. Acesso em: 15 jan. 2016.

DOMINICH, Sándor. **The modern algebra of information retrieval**. Veszprém: Springer Science & Business Media, 2008. 344p.

FAUZI, Irfan. **15 best free PHP frameworks of 2015**. 2015. Disponível em: <<http://beebom.com/2015/02/best-free-php-frameworks>>. Acesso em: 15 jan. 2016.

FERNEDA, Edberto. Aplicando algoritmos genéticos na recuperação de informação. **DataGramZero: revista de ciência da informação**, v. 10, n. 1, 2009. Disponível em: <http://www.dgz.org.br/fev09/Art_04.htm>. Acesso em: 12 jan. 2016.

_____. **Recuperação de informação**: análise sobre a contribuição da ciência da computação para a ciência da informação, 2003. 147 f. Tese (Doutorado) - Ciências da Comunicação - São Paulo: USP, 2003.

FOWLER, Martin; SCOTT, Kendall. **UML distilled second edition**: a brief guide to the standard object model language. Canada: Addison Wesley, 1999. 224 p.

FRAKES, William B.; BAEZA-YATES, Ricardo. **Information retrieval: data structures & algorithms**. [S.l.]: Prentice Hall, 1992. 504 p.

FRANKLIN, Alysson. **Tenha o DOM**: entenda o que é o document object model e tenha o DOM. 2011. Disponível em: <<http://tableless.com.br/tenha-o-dom/>>. Acesso em: 18 jan. 2016.

GESTÃO ELETRÔNICA DE DOCUMENTOS (GED). 2016. Disponível em: <<http://www.ged.net.br/>>. Acesso em: 08 jan. 2016.

GEY, F. Models in information retrieval. In: ACM CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL (SIGIR), 19th., 1992, [S.l.]. **Proceedings...** [S.l.]: ACM, 1992.

GOKER, Ayse; DAVIES, John. **Information retrieval: searching in the 21st century**. West Sussex: John Wiley, 2009. 320 p.

GOSPODNETIC, Otis; HATCHER, Erik. **Lucene in action**. Greenwich, United Kingdom: Manning Publications, 2005. 421 p.

GREENGRASS, Ed. **Information retrieval: a survey**. [S.l. s. n.], 2000. 224 p. Disponível em: <<http://www.csee.umbc.edu/csee/research/cadip/readings/IR.report.120600.book.pdf>>. Acesso em: 08 jan. 2016.

GROSSMAN, David. A.; FRIEDER, Ophir. **Information retrieval: algorithms and heuristics**. [S.l.]: Springer, 1998.

GUIYA, Rakesh; HENDREN, Laurie J.; ZHU, Yingchun. **Detecting parallelism in C programs with recursive data structures**. 2005. Lecture Notes in Computer Science, p. 159-173. v. 1383. Disponível em: <<http://link.springer.com/chapter/10.1007/BFb0026429>>. Acesso em: 22 jan. 2016.

HIEMSTRA, Djoerd. Information retrieval models. In: GOKER, Ayse; DAVIES, John. **Information retrieval: searching in the 21st century**, [S.l.]: Wiley, 2009. Disponível em: <<http://wwwhome.cs.utwente.nl/~hiemstra/papers/IRModelsTutorial-draft.pdf>>. Acesso em: 08 jan. 2016.

HJØRLAND, Birger. **OKAPI information retrieval system**. 2006. Disponível em: <http://www.iva.dk/bh/lifeboat_ko/SPECIFIC%20SYSTEMS/okapi_information_retrieval.htm>. Acesso em: 12 jan. 2016.

JONES, Karen S.; WALKER, Stephen; ROBERTSON, Stephen E. A probabilistic model of information retrieval: development and comparative experiments. **Information Processing and Management**, v. 36, n. 6, p. 779-808, 2000.

KNUTH, Donald. E. **The art of computer programming: fundamental algorithms**. 3. ed. [S.l.]: Addison Wesley Longman, 1997.

KOCABAS, Ilker; DINÇER, Bekir T., KARAOGLAN, Bahar. Investigation of Luhn's claim on information retrieval. **Turk J. Elec Eng & Comp Sci**, v. 19, n. 6, 2011. Disponível em: <<http://journals.tubitak.gov.tr/elektrik/issues/elk-11-19-6/elk-19-6-13-1003-448.pdf>>. Acesso em: 10 jan. 2016.

KOWALSKI, Gerald. **Information retrieval architecture and algorithms**. New York: Springer, 2011.

LARAVEL DOCUMENTATION. 2016. Disponível em: <<https://laravel.com/docs/4.2>>. Acesso em: 16 jan. 2016.

LAU, Lawrence J. **Economic Growth in the Digital Era**. Symposium on welcoming the challenge of the digital era. Stanford, 2003. Disponível em: <<http://web.stanford.edu/~ljlau/Presentations/Presentations/031129.pdf>>. Acesso em: 02 jan. 2016.

LUGO, Gustavo A. G. **Um modelo de sistemas multiagentes para partilha de conhecimento utilizando redes sociais comunitárias**. 2004. 214 f. Tese. (Doutorado) – Engenharia. USP: Escola Politécnica da Universidade de São Paulo, 2004.

MANNING, C. D.; PRABHAKAR, R.; SCHÜTZE, H. **An introduction to information retrieval**. [S.l.]: Cambridge University Press, 2009. Disponível em: <versao online>. Acesso em: 12 dez. 2015.

MARCUS, Adrian et al. Working Session: information retrieval based approaches in software evolution. In: IEEE INTERNATIONAL CONFERENCE ON SOFTWARE MAINTENANCE (ICSM'06), 22th., 2006, Philadelphia. **Proceedings...** Philadelphia: IEEE, 2006. p. 197-199. Disponível em: <<http://www.cs.wm.edu/~denys/pubs/Marcus.IR.WS2.pdf>>. Acesso em: 01 jan. 2016.

MARON, M. E.; KUHNS, J. L. On relevance, probabilistic indexing and information retrieval. **Journal of the association for computing machinery**. v. 7, p. 219-244, 1960.

MAUSAM. **Document similarity in information retrieval**. 2012. Disponível em: <<https://courses.cs.washington.edu/courses/cse573/12sp/lectures/17-ir.pdf>>. Acesso em: 10 jan. 2016.

MCCOOL, Shawn. **Laravel 3: starter**. [S.l.]: Packt Publishing, 2012.

MEYER, Christopher. **The connected economy: beyond the information age**. [S.l.]: Leader Values, 1996. Disponível em: <<http://www.leader-values.com/article.php?aid=160>>. Acesso em: 22 jan. 2016.

MIHHAILOV, Dimitri et al. **Hardware implementation of recursive algorithms**. 2010. Disponível em: <http://sweet.ua.pt/iouliia/Papers/2010/058_8208.pdf>. Acesso em: 22 jan. 2016.

MITCHELL, Tom M. **Machine learning**. [S.l.]: McGraw-Hill, 1997.

MOOERS, Calvin N. Zatocoding applied to mechanical organization of knowledge. **American documentation**, v. 2, p. 20-32, 1951. Disponível em: <<https://courses.engr.illinois.edu/cs473/fa2013/misc/zatocoding.pdf>>. Acesso em: 19 dez. 2015.

POLTROCK, Steven et al. Information seeking and sharing in design teams. In: INTERNATIONAL ACM SIGGROUP CONFERENCE ON SUPPORTING GROUP WORK (SIGGROUP '03), 2003, New York, NY, USA. **Proceedings...** New York, NY, USA: ACM, 2003. p. 239-247. Disponível em: <<http://research.microsoft.com/en-us/um/redmond/groups/coet/CIR/paper.pdf>>. Acesso em: 19 dez. 2015.

PENG, Fuchun et al. Context sensitive stemming for web search. In: ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL (SIGIR), 30th., 2007, [S.l.]. **Proceedings...** [S.l.]: ACM, 2007. p. 639-646.

PIANTADOSI, Steven T. Zipf's word frequency n natural language: a critical review and future directions. **Psychonomic Bulletin & Review**, v. 21, p. 1112-1130, 2014. Disponível em: <<https://colala.bcs.rochester.edu/papers/piantadosi2014zipfs.pdf>>. Acesso em: 07 jan. 2016.

PIROOZANIA, Mehdi; NAGARAJAN, Vijayaraj; DENG, Youping. GeneVenn: a web application for comparing gene lists using Venn diagrams. **Bioinformatics**, v. 1, p. 420-422, 2007. Disponível em: <<http://www.bioinformatics.net/001/009600012007.pdf>>. Acesso em: 14 jan. 2016.

QIAN, Gang et al. Similarity between euclidean and cosine angle distance for nearest neighbor queries. In: ACM SYMPOSIUM ON APPLIED COMPUTING, 2004, [S.l.]. **Proceedings...** [S.l.]: ACM, 2004.

REENSKAUG, Trygve; COPLIEN, James O. **The DCI architecture**: a new vision of object-oriented programming. 2009. Disponível em: <http://www.artima.com/articles/dci_vision.html>. Acesso em: 15 jan. 2016.

ROBERTSON, S. E.; Jones, K. S. Relevance weighting of search terms. **Journal of the american society for information science**, v. 27, n. 3, p. 129-146, 1976. Disponível em: <<http://www.staff.city.ac.uk/~sb317/papers/RSJ76.pdf>>. Acesso em: 11 jan. 2016.

ROWLEY, Jenifer. **A biblioteca eletrônica**. 2. ed. Brasília: Briquet de Lemos/Livros, 2002. 399 p.

RUSSEL, Stuart J.; NORVIG, Peter. **Artificial intelligence: a modern approach**. 2. ed. [S.l.]: Prentice-Hall, 2003.

SALTON, G. **The SMART retrieval system: experiments in automatic document processing**. [S.l.]: Prentice-Hall. 1971.

_____; YANG, C. On the specification of term values in automatic indexing. **Journal of Documentation**, v. 29, n. 4, p. 351-372, 1973.

SARACEVIC, T. Ciência da informação: origem, evolução e relações. **Perspectivas em ciência da informação**, v. 1, n. 1, p. 41-62, jan./jun. 1996. Disponível em: <<http://portaldeperiodicos.eci.ufmg.br/index.php/pci/article/download/235/22>>. Acesso em: 08 jan. 2016.

SAUNIER, Raphaël. **Getting started with Laravel 4**. [S.l.]: Packt Publishing. 2014.

SCHREIBER, Jacques N. C. et al. GIRS - Genetic information retrieval system. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, XXVIII., 2008, Rio de Janeiro. **Anais...** Rio de Janeiro: ENEGEP, 2008. Disponível em: <http://www.abepro.org.br/biblioteca/enegep2008_TN_STO_076_537_10710.pdf>. Acesso em: 02 jan. 2016.

SEDGEWICK, R.; WAYNE, K. **Algorithms**. 4th. ed. [S.l.]: Addison-Wesley, 2011. Disponível em: <<http://www.ime.usp.br/~pf/estruturas-de-dados/aulas/tries.html>>. Acesso em 07 jan. 2016.

SILBERSCHATZ, Avi; KORTH, Henry F.; SUDARSHAN, S. **Database system concepts**. 4th. ed. [S.l.]: McGraw-Hill, 2001.

SILVA, Renata E.; SANTOS, Plácida L. V. A. da C.; FERNEDA, Edberto. Modelos de recuperação de informação e web semântica: a questão da relevância. **Inf. Inf.**, Londrina, v. 18, n. 3, p. 27-44, set./dez. 2013. Disponível em: <http://www.uel.br/revistas/uel/index.php/informacao/article/viewFile/12822/pdf_3>. Acesso em: 05 jan. 2016.

SINGHAL, Amit. Modern information retrieval: a brief overview. **Bulletin of the IEEE Computer Society Technical Committee on Data Engineering**, v. 24, n. 4, p. 35-43. 2001. Google, Inc. Disponível em: <<http://singhal.info/ieee2001.pdf>>. Acesso em: 06 jan. 2016.

SMITH, Ray. **An overview of the Tesseract OCR engine**. Oscon: IEEE, 2007. Disponível em: <http://www.helsinki.fi/~mpsilfve/ocr_course/materials/tesseractidar2007.pdf>. Acesso em: 16 jan. 2016.

SPRAGUE Jr.; Ralph H. **Electronic document management: challenges and opportunities for information systems managers**. Hawaii: University of Hawaii, College of Business Administration, 1995. Disponível em: <<http://sprague.shidler.hawaii.edu/MISQ/MISQfina.htm>>. Acesso em: 04 jan. 2016.

STEINBRUCH, Alfredo; WINTERLE, Paulo. **Geometria analítica**. 2. ed. São Paulo: Makron-Books, 1987. 292 p.

TREC. **Text retrieval conference**. 2007. Disponível em: <<http://trec.nist.gov/data/enterprise.html>>. Acesso em: 21 jan. 2016.

W3C. **Web services definition**: position paper. [S.l.]: IONA technologies, 2001. Disponível em: <<https://www.w3.org/2001/03/WSWS-popa/paper13>>. Acesso em: 20 jan. 2016.

ZOBEL, Justin; MOFFAT, Alistair. **Inverted files for text search engines**. New York, EUA: ACM Press, 2006.

APÊNDICE A - Lista de artigos indexados e sua classificação por especialistas

ID	TÍTULO	AUTOR	RELEV. À CONS.
1	Análise teórica da recuperação de calor para geração de energia em indústrias de cimento e cal utilizando o ciclo de rankine orgânico	CARPIO, R. C. <i>et al</i> (2015)	
2	Importância dos aspectos socioculturais na gestão de equipes em ambientes de desenvolvimento distribuído de software	ZUQUELLO, A. G. <i>et al</i> (2015)	9, 12
3	Técnica de modelagem numérica pelo método dos elementos finitos para estudo da resposta acústica veicular	FERREIRA, T. S. <i>et al</i> (2015)	11
4	A distribuição beta Fréchet transmutada: propriedades e aplicação a dados de sobrevivência	SILVA, A. L.; RODRIGUES, J. A.; SILVA, G. O. (2015)	11
5	Análise de um objeto de aprendizagem na perspectiva da teoria da aprendizagem significativa: o professor diante da possibilidade de transformação	LOPES, V. (2015)	2
6	A utilização de jogos como metodologia de ensino da matemática: uma experiência com alunos do 6º ano do ensino fundamental	BARBOSA, C. P. <i>et al</i> (2015)	2
7	Ferramenta de auxílio no processo de medição de energia elétrica utilizando inteligência computacional	PEREIRA, M. A. S. (2015)	1, 11
8	O problema do carpinteiro: estudando semelhança de triângulos por meio da fachada de uma casa	PAIM, M. A. S. (2015)	2
9	Usando splines cúbicas na modelagem matemática da evolução populacional de Pirapora/MG	DOMINGUES, J. S. <i>et al</i> (2014)	11
10	ALBOR: um simulador didático para auxiliar no ensino e aprendizagem de instruções Assembly	BORTH, M. R.; OLIVEIRA, A. S. (2014)	1, 2
11	CUDA vs. OpenCL: uma comparação teórica e tecnológica	PAULA, L. C. M. (2014)	
12	Portando uma aplicação de sistema embarcado com arquitetura super loop para operar com sistema operacional de tempo real	FRIGIERI <i>et al.</i> (2014)	1, 8
13	Dimensionamento de um sistema micro-ondas para distribuição de sinais de TV digital usando o software Radio Mobile	ANJOS, A. A.; SILVA JUNIOR, R. A.; GOGLIATTI, R. (2014)	1
14	CUDA vs. OpenCL: uma comparação teórica e tecnológica	PAULA, L. C. M. (2014)	2
15	Jogos e materiais manipuláveis produzidos por alunos do IFBA, Campus de Eunápolis	PAIM, M. A. S. (2014)	2
16	Independência do Conselho de Administração e composição da estrutura de capital das empresas listadas na BOVESPA: um estudo entre os anos de 2007-2011	MOREIRA, B. C. M. <i>et al.</i> (2014)	11
17	Análise matemática de um modelo para crescimento de células-tronco cancerígenas em tumores	GOUVEA, M. E. <i>et al.</i> (2014)	4, 11

18	A legitimação dos benefícios do jiu-jitsu em uma organização do terceiro setor: um estudo de caso no tatame do bem	ABREU, A. A.; P. F.; CAMPOS, R. C. L. (2015)	
19	Otimização do arranjo físico: um estudo de caso em uma marcenaria	SILVA, P. M. S. <i>et al.</i> (2015)	6
20	Modelagem matemática e a determinação de um novo método de cálculo do volume ventricular	DOMINGUES, J. S. <i>et al.</i> (2015)	4, 11
21	Desenvolvimento de hardware reconfigurável de criptografia assimétrica	GOMES, O. S. M.; CÉSAR, J. P. C. (2015)	8
22	Projeto e desenvolvimento de um hardware reconfigurável de criptografia para a transmissão segura de dados	GOMES, O. S. M.; GUIMARÃES, R. L. M. (2015)	8
23	Projeto e desenvolvimento de um carro robô controlado por smartphone, utilizando a plataforma Amarino	GOMES, O. S. M. <i>et al.</i> (2015)	1, 8, 10
24	Robô seguidor de linha para competições	GOMES, O. S. M. <i>et al.</i> (2015)	1, 8
25	Um sistema para análise de genomas a partir de metagenomas	PEREIRA, V. M. Y.; SANTOS JÚNIOR, G. J.; DIGIAMPIETRI, L. A. (2015)	1, 4
26	Experimentação da técnica de narrativas OCC-RDD na prática. Um estudo de caso de uma aula no curso superior de ensino à computação	BUTTIGNON, K.; VEGA, I. S. (2015)	2
27	Análise de Instantes de Trânsitos em Exoplanetas Usando o Programa TAP	PEREIRA, M. G.; ALMEIDA, L. A.; OLIVEIRA, A. C. (2015)	1, 11
28	A Influência da Meta-usabilidade e Comunicabilidade nas Técnicas de Inspeção de Aplicações Web	SOUZA, B. P.; FERNANDES, P. S. (2015)	9
29	Avaliação do Consumo energético em Arquiteturas multi-Core com Memória Cache Compartilhada	SOUZA, M. A. <i>et al.</i> (2014)	8
30	Uso de avaliação por pares em disciplinas introdutórias de programação	CORREIA, A. L. <i>et al.</i> (2015)	2
31	eScience-as-a-Service: Desafios e Oportunidades para a Criação de Nuvens Científicas	COSTA, R. <i>et al.</i> (2011)	5
32	Análise de Rede de Colaboração Científica como Ferramenta na Gestão de Programas de Pós-graduação	COSTA, A. R.; RALHA, C. G. (2015)	
33	Assistente Digital para Recomendação Turística em Mapas Interativos	PORTO, H. A.; PATTO, V. S. (2015)	1, 5
34	Classificação de sinais EGG combinando Redes Neurais e Análise de Componentes Independentes	SANTOS, H.; MONTESCO, C. A. E.; JÚNIOR, M. C. (2015)	4, 11
35	Gestão Semântica de Dados Meteorológicos Apoiados Por Ontologias de Proveniência	CRUZ, S. M. S. <i>et al.</i> (2015)	7
36	Realidade Aumentada em saúde: uma revisão sobre aplicações e desafios	ZORZAL, E. R.; NUNES, F. L. S. (2014)	4
37	Scratch na produção de recursos interdisciplinares com disciplinas indígenas	RABÊLO, H. M. <i>et al.</i> (2015)	2
38	Uma Arquitetura P2P de Distribuição de Atividades para Execução Paralela de Workflows Científicos	SILVA, V. <i>et al.</i> (2013)	5

39	Uma avaliação da Distribuição de Atividades Estática e Dinâmica em Ambientes Paralelos usando o Hydra	SILVA, V. <i>et al.</i> (2011)	5
40	Aplicação da Ciência dos Dados em Hidrologia para Estimativa da Evaporação	XAVIER, F.; TANAKA, A. K. (2015)	11
41	Cultura Organizacional na Adoção de Metodologias Ágeis no Desenvolvimento de Sistemas de Informação - Rumo a um Modelo Conceitual à Luz de um Estudo Sistemático	AMARAL, J. P.; GOMES, P.; GRACIANO NETO, V. V. (2015)	9
42	Avaliação de arquiteturas manycore e do uso da virtualização de GPUs	MANFROI, L. L. F. <i>et al.</i> (2014)	
43	Implementação Computacional de Fusão Automática de Dados Distribuídos em Apoio à Gestão de Saúde	CERCEAU, R. <i>et al.</i> (2015)	4, 11
44	Ambiente Virtual para Modelagem dos Canais Radiculares	BOGONI, T. N. (2014)	1, 4
45	NanoTrack - Sistema de Gerenciamento de Dados de Nanoestruturas com Plugins Inteligentes	SILVA, A. S. F. <i>et al.</i> (2013)	1
46	Aprendizado automático de ontologias a partir de Data Warehouses	SILVA, T. O.; BAIÃO, F.; REVOREDO, K. (2015)	11
47	Avaliação de um Modelo de Maturidade para Governança Ágil em TIC usando Focus Group	ALMEIDA NETO, H. R. <i>et al.</i> (2015)	9
48	Desenvolvimento de Aplicações para Dispositivos Móveis: Tipos e Exemplo de Aplicação na plataforma iOS	SILVA, L. L. B.; PIRES, D. F.; CARVALHO NETO, S. (2015)	1, 10
49	Uma abordagem para a redução da tabela de encaminhamento sob a ótica da interface de saída dos pacotes	FARIAS, L. F. T.; DINIZ, M. C.; LUCENA, S. C. (2014)	5
50	Uso de Redes Neurais para Previsão da Temperatura da Superfície do Mar do Oceano Atlântico Tropical	MATTOS, P. <i>et al.</i> (2015)	11
51	Uso da ontologia Ontocancro para comprovar o envolvimento de genes com a barreira de progressão do câncer	FALCADE, L. <i>et al.</i> (2014)	1, 4
52	Testando a Diversão em um Jogo Sérioso para o Aprendizado Introdutório de Programação	VAHLDICK, A. <i>et al.</i> (2015)	1, 2
53	Assimilação, Controle de Qualidade e Análise de Dados de Meteorológicos Apoiados por Proveniência	FILHO, G. R. L. <i>et al.</i> (2013)	1, 11
54	Predição de Fluídos em um Reservatório Petrolífero Utilizando Métodos de Previsão de Séries Temporais	BUSTOS, H. I. A. <i>et al.</i> (2011)	11
55	Aplicação da otimização por colônia de formigas ao problema de múltiplos caixeiros viajantes no atendimento de ordens de serviço nas empresas de distribuição de energia elétrica	BARBOSA, D. F.; SILLA JR, C. N.; KASHIWABARA, A. Y. (2015)	6, 11
56	Desenvolvimento de uma Formulação Linear Inteira para o Problema de Motifs em Grafos	BRIGATTO, F.; LIMA, K. R. P. S. (2015)	6, 11
57	Estudo da aplicação de técnicas inteligentes em mineração de processos de negócio	MAITA, A. R. C.; FANTINATO, M.; PERES, S. M. (2015)	11, 12
58	XORBR: Roteamento Baseado em uma Métrica de OU-Exclusivo e Filtros de Bloom para Redes	CRUZ, E. P. F. <i>et al.</i> (2014)	5

Veiculares Urbanas			
59	Ontologia de Aplicação para o Lago Batata	DOUZA, A. N.; MEDEIROS, A. P. (2015)	
60	O Universo Lúdico da Programação de Computadores com Logo no Ensino Fundamental	SOUZA, A. <i>et al.</i> (2015)	2, 10
61	RFlow: Uma Abordagem de Reutilização de Workflows Estatísticos Legados	NASCIMENTO, J. A. P.; CRUZ, S. M. S. (2013)	9, 12
62	Métricas para ontologias no formato OBO: Um estudo utilizando o Cytoscape	ASSAIFE, A. C. G. S. <i>et al.</i> (2011)	
63	Diretrizes para uma Metodologia de Desenvolvimento de Software Aplicada a Startups de Tecnologia da Informação	SOUZA, G. <i>et al.</i> (2015)	9
64	Mecanismos de Difusão Limitada de Interesses em Redes em Malha Sem-Fio Orientadas a Conteúdo	MASCARENHAS, D. M.; MORAES, I. M. (2014)	5
65	Explorando o pensamento computacional no ensino médio: do design à avaliação de jogos digitais	FRANÇA, R. S.; TEDESCO, P. (2015)	2
66	Uma Abordagem de Gerenciamento Semântico de Experimentos Meteorológicos em Pluviometria	BARBOSA, T. M. S.; CRUZ, S. M. S. (2013)	
67	Tratamento de Inferência em Banco de Dados Ecológicos	POLTOSI, M. <i>et al.</i> (2011)	7, 11
68	Aplicação de Árvores de Decisão para Recomendação de Parâmetros e Workflows Científicos	CÂMARA, R. V.; PAES, A.; OLIVEIRA, D. (2015)	11
69	Elicitação e Estruturação de Conhecimento: Um Arcabouço Conceitual Aplicado à Biodiversidade	ALBUQUERQUE, A. C. F.; SANTOS, J. C.; CASTRO JR, A. N. (2015)	
70	Análise Comparativa de Métodos de Aprendizagem de Máquina para Classificação de Massas em Mamografias	MELO, M. C.; GAJADHAR, A. A.; BATISTA, L. V. (2014)	4, 11
71	Mineração de Regras de Classificação de Câncer utilizando Nondominated Sorting Genetic Algorithm II (NSGA-II)	COELHO, V. L.; SALES JUNIOR, C. S. (2013)	4, 6, 11
72	Auto-parametrização do GRASP com Path-Relinking no agrupamento de dados com F-Race e iterated F-Race	SILVA, J. C. <i>et al.</i> (2015)	6, 11
73	Avaliação de Desempenho de Plataformas para Validação de Redes Definidas por Software	LIBERATO, A. B. <i>et al.</i> (2015)	5, 11
74	Um Processo Exploratório para Classificação de Estrelas e Galáxias	MACHADO, E. <i>et al.</i> (2015)	
75	Motion Rehab: um jogo sério para idosos com sequelas de Acidente Vascular Encefálico	FIORIN, M. R. <i>et al.</i> (2014)	1, 4
76	Métodos Ágeis em um Núcleo de Práticas Acadêmico: Relato de Experiência	ALMENDRA, C. C. <i>et al.</i> (2015)	9, 12
77	DynAdapt: Alterações na Definição de Atividades de Workflows Científicos em Tempo de Execução	SANTOS, I. A. <i>et al.</i> (2013)	
78	Composicionalidade e Reuso em Workflows Científicos com Propriedades Não-Funcionais	MEDEIROS, V.; GOMES, A. T. A. (2011)	
79	GiveMe Views: uma ferramenta de suporte a	TAVARES, J. F. <i>et al.</i> (2015)	1, 9

	evolução de software baseada na análise de dados históricos		
80	BroFlow: Um Sistema Eficiente de Detecção e Prevenção de Instrusão em Redes Definidas por Software	LOPEZ, M. A. <i>et al.</i> (2014)	1, 5
81	Caracterizando os desafios na modelagem dos dados clínicos em Sistemas de RES baseados no OpenEHR	TULER, E. <i>et al.</i> (2014)	
82	Sistema de Apoio à Prática Assistida de Programação por Execução em Massa e Análise de Programas	OLIVEIRA, M. G.; NOGUEIRA, M. A.; OLIVEIRA, E. (2015)	1, 2, 10
83	CYCLOPS-Web: Um portal de simulações da emissão de estrelas do tipo polares	EMYGDIO, D. <i>et al.</i> (2015)	1
84	Aplicação de inferência difusa em bioinformática para identificação de SNPs	ARBEX, W. <i>et al.</i> (2011)	4, 11
85	Aplicações Android de Realidade Aumentada em Arquitetura Extensível, Flexível e Adaptável	ARAÚJO, T. <i>et al.</i> (2015)	1, 10
86	Modelo de simulação em OMNET++ para avaliação de desempenho da rede de comunicação de um SAS baseado na Norma IEC61850	MOLANO, D. L. A. <i>et al.</i> (2014)	5, 11
87	Método de Detecção de Câncer de Ovário Utilizando Padrões Proteômicos, ANálise de Componentes Independentes e Máquina de Vetores de Suporte	ARAÚJO, W. B. D.; CAMPOS, L. F. A.; FURTADO, A. S. (2014)	11
88	Resgatando a Linguagem de Programação Logo: Uma Experiencia com Calouros no Ensino Superior	RAIOL, A. A. C. <i>et al.</i> (2015)	2, 10
89	Decifrando Cisteíno Proteases em Plasmodium: Uma Estratégia de Genômica Comparativa e Modelagem Estrutural	OCAÑA, K. A. C. S.; GARCIA-ZAPATA, M. T. A. (2013)	4
90	Um Mapeamento Sistemático Sobre o Uso de Metodologias Ágeis no Processo de Experimentação Científica	ROMEIRO, A. C.; OLIVEIRA, D. (2015)	9, 12
91	Métricas Morfológicas para a Classificação de Tumores de Mama	SOUTO, L. P. M.; SANTOS, T. K. L.; SILVA, M. P. S. (2014)	4, 11
92	Uma Análise Comparativa de Kits para a Robótica Educacional	COSTA JR., A. O.; GUEDES, E. B. (2015)	2, 8
93	Expressando Atributos Não-Funcionais em Workflows Científicos	MEDEIROS, V.; GOMES, A. T. A. (2013)	12
94	Arquitetura de Software de Referência para Sistemas de Informação Governamentais	SERRANO, M.; SERRANO, M.; CAVALCANTE, A. C. (2015)	9, 12
95	Reordenando Assinaturas em Mecanismos de Inspeção de Pacotes Baseado em Prioridade Dinâmica	PETRÔNIO JÚNIOR <i>et al.</i> (2014)	5
96	Pipeline para identificação e armazenamento de repetições adjacentes	FRANCO, M. E.; MARTINS, M. A.; FACHIN, A. L. (2015)	4
97	O ensino de conceitos computacionais para alunos do ensino médio: relato de experiência de uma gincana e das estratégias utilizadas pelos alunos na resolução das atividades desplugadas	BARBOSA, A. V. S.; <i>et al.</i> (2015)	2
98	Ecossistemas de Startup de Software: Resultados Iniciais no âmbito do Estado do Pará	TORRES, N. N. J.; SOUZA, C. R. B. (2015)	

99	Identificação de gargalos de desempenho em ambientes virtuais para uso em computação em nuvem	FERREIRA, C. H. G. <i>et al.</i> (2014)	5
100	Georreferenciamento de dados biológicos legados no INPA	FARIAS, E. M. B. SANTOS, J. L. C. (2015)	
101	Estimação da idade óssea: comparativo entre valores obtidos com a metodologia Contornos Ativos Snakes versus valores médios obtidos com os métodos de Eklof & Ringertz, Tanner & Whitehouse e Greulich & Pyle	OLIVETE JÚNIOR, C.; CORREIA, R. C. M.; GARCIA, R. E. (2014)	3, 4, 11
102	O Ensino de Computação na Educação Básica apoiado por Problemas: Práticas de Licenciandos em Computação	ARAÚJO, D. C. <i>et al.</i> (2015)	2
103	PALMS+: Protocolo ALM baseado em desigualdade triangular para distribuição de streaming de vídeo	CASTRO, B. P. <i>et al.</i> (2014)	5
104	Análise de Séries Temporais de Sinais Térmicos da Mama para Detecção de Anomalias	SILVA, L. F. <i>et al.</i> (2014)	4, 11
105	Nova grade curricular do BCC-IME-USP	BATISTA, D. M. <i>et al.</i> (2015)	2
106	Ciberinfraestrutura para integração, acesso, compartilhamento e reuso de dados de pesquisa da área nuclear	SALES, L. F.; SAYÃO, L. F. <i>et al.</i> (2013)	1
107	MapReduce vs BSP para cálculo de medidas de centralidade em grafos grandes	BANHOS FILHO, F. S.; YERO, E. J. H. (2014)	11
108	DEG4Trees: Um Jogo Educacional Digital de Apoio ao Ensino de Estruturas de Dados	BARBOSA, E. A. <i>et al.</i> (2015)	2
109	Algoritmos de Cross-Matching de Dados Astronômicos	FREIRE, V. P. <i>et al.</i> (2013)	7, 11
110	Caracterização do Perfil de Tráfego de Aplicações Adaptativas de Fluxo Contínuo de Mídia sobre HTTP	ITO, M. S. <i>et al.</i> (2014)	5
111	Educação Tutorial em Ciência da Computação: uma proposta de sistematização	FERREIRA, J. M. S. <i>et al.</i> (2015)	2
112	Green Markov - Uma abordagem híbrida de Política Markoviana e Simulação Discreta para Planejamento de Alocação de Usuários em Redes Macro/Femto	NATAL, I. <i>et al.</i> (2014)	5, 11
113	A Importância do Fator Motivacional no Processo Ensino-Aprendizagem de Algoritmos e Lógica de Programação para Alunos Repetentes	SANTOS, A. <i>et al.</i> (2015)	2
114	Avaliação da Influência da Remoção de Stopwords na Abordagem Estatística de Extração Automática de Termos	BRAGA, I. A. (2009)	11
115	MapReduce vs Bancos de Dados Paralelos no cálculo de medidas de centralidade em grafos	FERNANDES, F. S.; YERO, E. J. H. (2014)	7, 11
116	Aluno Surdo na Ciência da Computação: Discutindo os Desafios da Inclusão	BOSCARIOLI, C.; <i>et al.</i> (2015)	2
117	Um Módulo de Sensoriamento Voluntário para um Sistema de Monitoramento de Desmatamento	CORRÊA, F. R. S.; LUZ, E. F. P.; RAMOS, F. M. <i>et al.</i> (2013)	1
118	Avaliação do comportamento de consumidores no processo de decisão de compra no M-Commerce e	LEMOS, F.; GÓES, L. F. (2015)	11

no E-Commerce			
119	Tornando Paxos Mais Escalável com Réplicas Leitoras	PAULA, A. P.; VIEIRA, G. M. D. (2014)	5
120	URI Online Judge Academic: Integração e Consolidação da Ferramenta no Processo de Ensino/Aprendizagem	SELIVON, M.; BEZ, J. L.; TONIN, N. A. (2015)	2
121	Um Estudo das Características de Alta Prevalência em Redes sem Fio Infraestruturadas IEEE 802.11	LUZ, K.; SOLIS, P.; GARCIA, H. (2014)	5
122	A formação didático-pedagógica do docente da área de computação: um estudo de caso em uma Universidade Brasileira	MASSA, M. S. (2015)	2
123	Reescrita de código e utilização de Paralelismo de Dados para a busca de Small RNAs (sRNAs)	ALMEIDA, A. G. D. <i>et al.</i> (2013)	
124	Análise de Desempenho em Dispositivos Limitados e Emulados Estudo de Caso: Raspberry Pi e Web Services RESTful	NUNES, L. H. <i>et al.</i> (2014)	8
125	Mundo virtual Minecraft: uma Experiência no Ensino de Circuitos Digitais	CAGNINI, H. E. L. <i>et al.</i> (2015)	1, 2
126	Uma proposta de adaptação do algoritmo Branch-and-Prune usando a interseção de quatro esferas para o Problema de Geometria de Distâncias Moleculares	SANTOS, C. S.; RODRIGUES, R. F.; MOTA, K. (2013)	6, 11
127	Avaliação de características que influenciam nos votos de utilidade de opiniões sobre serviços em Português	MARTINS, A. C. S.; TACLA, C. A. (2015)	11
128	ATTA: Um Ambiente Integrado de Testes para Redes em Malha Sem Fio IEEE 802.11s	CARVALHO, D. F.; PINHEIRO, M. C. M. A. (2014)	1, 5
129	Um Estudo sobre a Evasão no Curso de Licenciatura em Informática do IFRN - Campus Natal - Zona Norte	SOUZA, O. S. <i>et al.</i> (2015)	2
130	O apoio computacional para análise quantitativa da expressão de genes envolvidos com a barreira de progressão do câncer	NASCIMENTO, K. S. <i>et al.</i> (2013)	4, 11
131	Avaliação de Técnicas de Mineração de Dados para Predição de Desligamentos em Sistemas Elétricos de Potência	MAIA, A. T. <i>et al.</i> (2015)	11
132	Caracterizando conexões P2P entre usuários do ISP: uma análise utilizando redes complexas	DAMASCENO, A. G. M.; MARQUES-NETO, H. T. <i>et al.</i> (2014)	5
133	SACI - ainda outro ambiente para o ensino de programação	ANIDO, R. (2015)	2
134	Remoção de Mensagens Obsoletas em Conjunto com Políticas de Gerenciamento de Buffer para Redes DTN	GOMES, E. N. <i>et al.</i> (2014)	5
135	Ensino da Robótica Livre como Instrumento de Aprendizado Interdisciplinar na Rede Pública de Educação Profissional e Tecnológica	DIAS, J.; ABDALLA, D.; SABA, H. (2015)	2
136	Aplicação de Descoberta de Conhecimento em Bases de Dados na Estimativa da Evapotranspiração: um Experimento no Estado do Rio de Janeiro	XAVIER, F.; TANAKA, A. K.; REVOREDO, K. C. (2015)	11

137	Clube de Computação para Alunos de Ensino Médio: um Relato de Experiência	CHARÃO, A. S. <i>et al.</i> (2015)	2
138	Metodologia de Geração de Plataforma de Inteligência de Negócios para Comissões de Avaliação de Universidade	RIBEIRO, C. E.; RIBEIRO, A. I. J. T. (2015)	9
139	GSPROJECTS - Ambiente para simulação da gestão de projetos de software	LEITE, D. R. A. <i>et al.</i> (2015)	1, 2, 12
140	Mineração de regras de associação temporais quantitativas por meio de algoritmo genético	SILVA, S. F.; BATISTA, M. A.; TRAINA, A. J. M. (2015)	6, 11
141	RPG4Sorting - Um Jogo Educacional para Auxílio ao Ensino de Métodos de Ordenação	NUNES, I. F.; PARREIRA JÚNIOR, P. A. <i>et al.</i> (2015)	2
142	Aprendizado de Máquina em Sistemas de Recomendação Baseados em Conteúdo Textual: Uma Revisão Sistemática	BRUNIALTI, L. F. <i>et al.</i> (2015)	
143	Robô para Reconstrução Tridimensional: Uma aplicação didático-pedagógica do protótipo no âmbito da Engenharia e Computação	SILVA, J. L. S. <i>et al.</i> (2015)	2, 8
144	Oficinas de Programação com Ambientes Lúdicos para Meninas do Ensino Fundamental	SOUZA, S. M. <i>et al.</i> (2015)	2
145	Relato de Experiência de Ensino de Computação no Ensino Fundamental em Estágio Supervisionado da Universidade de Pernambuco no Campus Garanhuns	SILVA, S. F. <i>et al.</i> (2015)	2
146	Oficinas de Aprendizagem de Programação em Uma Escola Pública através do Ambiente Scratch	BATISTA, W. P. <i>et al.</i> (2015)	2, 10
147	Amadurecimento, Consolidação e Performance de SGBDs NoSQL - Estudo Comparativo	SOUZA, V. C. O.; SANTOS, M. V. C. (2015)	7, 11
148	Uma Abordagem Gamificada para o Ensino de Programação Orientada a Objetos	FIGUEIREDO, K. S. <i>et al.</i> (2015)	2, 10
149	Utilização do Índice de Estilos de Aprendizagem de Felder-Soloman em Turmas de Nível Técnico, Graduação e Pós-Graduação em Computação	AGUIAR, J. J. B.; FECHINE, J. M.; COSTA, E. B. (2015)	2
150	Melhoria de Processos de Software sob a Perspectiva dos Vieses Cognitivos: Uma Análise de Múltiplos Casos	CUNHA, J. A. O. G. <i>et al.</i> (2015)	9, 12
151	Identificando os Traços de Personalidade de Estudantes de um Curso Técnico em Informática	AGUIAR, J. J. B.; FECHINE, J. M.; COSTA, E. B. (2015)	2
152	Governança de Dados em Organizações Brasileiras	BARATA, A. M.; PRADO, E. P. V. (2015)	12
153	Uma ferramenta gamificada de apoio à disciplina introdutória de programação	CAMPOS, A.; GARDIMAN, R.; MADEIRA, C. (2015)	1, 2
154	Análise de Rede de Colaboração Científica como Ferramenta na Gestão de Programas de Pós-graduação	COSTA, A. R.; RALHA, C. G. (2015)	2, 11
155	IntroComp: Atraindo Alunos do Ensino Médio para uma Instigante Experiência com a Programação	MENESES, L. F. <i>et al.</i> (2015)	2, 10
156	A Utilização de Kits de Robótica como Ferramenta para o Ensino de Programação às Meninas do Ensino Médio	MATTOS, G. O.; SILVA, D. R. D.; MOREIRA, J. A. (2015)	2, 8

157	Apoio Automatizado ao Planejamento de Sprints em Projetos Scrum	SOUSA, H. T. <i>et al.</i> (2015)	12
158	Ensino de Programação para Alunas de Ensino Médio: Relato de Uma Experiência	RAMOS, N. <i>et al.</i> (2015)	2, 10
159	ABILSEN: Uma Abordagem para Inclusão do Licenciado em Computação no Ensino Básico	SILVA NETO, S. R.; SANTOS, H. R.; SANTOS, W. O. (2015)	2
160	Análise dos Desafios para Estabelecer e Manter Sistema de Gestão de Segurança da Informação no Cenário Brasileiro	FAZENDA, R. V.; FAGUNDES, L. L. (2015)	
161	Computação para Ensino Médio na Modalidade Semipresencial: Uma Experiência da Disciplina de Estágio Supervisionado	FERREIRA, M. A. <i>et al.</i> (2015)	2
162	Investigação sobre a Ausência de Validação nos Métodos Empregados para Quantificar Segurança da Informação	MIANI, R. S.; ZARPELÃO, B. B.; MENDES, L. S. (2015)	11
163	Sistematização da elaboração da matriz curricular de um curso de Sistemas de Informação: a metodologia dos perfis	CARDOSO, R. (2015)	2
164	Reprovações e Trancamentos nas Disciplinas de Introdução à Programação da Universidade de São Paulo: Um Estudo Preliminar	BOSSE, Y.; GEROSA, M. A. (2015)	2
165	Análise das Ementas de Estruturas de Dados das Universidades Brasileiras	NUNES, M. M. <i>et al.</i> (2015)	2
166	Gestão de Regras de Autorização Usando Modelo Conceitual	SUL, R. D. <i>et al.</i> (2015)	
167	Adaptação de um Checklist para Análise de Transparência de Software em Sites	FORTE, F. B.; VILAIN, P.; MACEDO, F. F. (2015)	9
168	Arquitetura de Compartilhamento Transparente de Conteúdos Entre Dispositivos Móveis em Redes Oportunidades	BATISTA, C. T.; SILVA, M. J.; OLIVEIRA, R. A. (2015)	5
169	Características da adoção de software de código aberto: Um estudo sobre o setor de tecnologia da informação de Minas Gerais	CARVALHO, L. G.; GOMES, O. A.; PARREIRAS, F. S. (2015)	9
170	Manejo Tecnológico de Lavouras Através de Dispositivos Móveis e Agricultura de Precisão	CRUZ, S. M. S. <i>et al.</i> (2015)	1
171	Avaliação do Nível de Matridade em Teste de Software em Micro e Pequenas Empresas do Estado de Goiás	ARAÚJO, A. F. <i>et al.</i> (2015)	
172	Avaliação do uso da plataforma de criação de prontuários eletrônicos SANA para especialistas de saúde básica	VALLE, T. B.; CARVALHO, D. B. F. (2015)	1, 4
173	Experimentação na Indústria para Aumento da Efetividade da Construção de Procedimentos ETL em um Ambiente de Business Intelligence	COSTA, J. K. G. <i>et al.</i> (2015)	9
174	Fotosenti: Um aplicativo para auxiliar em tratamentos psicológicos	CARVALHO, D. B. F.; ARAÚJO, S. M. A. (2015)	1, 4
175	Adicionando informações estruturadas ao Bulário Eletrônico da ANVISA	SILVA, J. V. F.; SILLA JR., C. N.; KASHIWABARA, A. Y. (2015)	

176	Chronos Ações: Ferramenta para Apoiar a Tomada de Decisão de Investidores da Bolsa de Valores	NASCIMENTO, T. C. S.; MIRANDA, L. C. (2015)	1, 11
177	Conhecendo a Comunidade de Sistemas de Informação no Brasil: um Estudo Comparativo Utilizando Diferentes Abordagens de Banco de Dados	RODRIGUES, N. S.; RALHA, C. G. (2015)	7
178	Ensino de Sistemas de Informação em Cursos de Computação: relato de experiência com uso de abordagem prática em TIC	ISHIKAWA, E.; RALHA, C. G. (2015)	2
179	A Propensão de Usuários à Adoção de Tecnologias: Um Estudo com Usuários e não Usuários do Programa "Nota Legal" no Distrito Federal	FARIAS, J. S.; LINS, P. V.; ALBUQUERQUE, P. H. M. (2015)	
180	Estudo sobre o engajamento de usuários de uma mídia social disponibilizada pelo governo	SILVA, C. M. C.; PRADO, E. P. V. (2015)	
181	ArgumentBind - Um Modelo para Implementação de Aplicações da ArgumentWeb Integradas com Bases de Dados Abertos e Ligados	NICHE, R.; RIGO, S. J. (2015)	1, 7, 9
182	Inspeção da Interação em sítios Governamentais: uma comparação entre métodos	MARQUES, V. F. <i>et al.</i> (2015)	9
183	Ecosistemas Digitais para o Apoio a Sistemas de Governo Abertos e Colaborativos	MAGDALENO, A. M.; ARAÚJO, R. M. (2015)	9
184	Análise Comparativa de Algoritmos de Mineração de Texto Aplicados a Históricos de Contas Públicas	SANTOS, B. S. <i>et al.</i> (2015)	11
185	Acessibilidade em Governo Eletrônico: um estudo sobre a aplicação de padrões web em sítios gov.br	OLIVEIRA, A. D. A.; ELER, M. M. (2015)	9
186	Análise do Perfil e do Papel do Analista de Negócios no Contexto Nacional	ZADRA, V. C.; PORTO, J. B. (2015)	12
187	Como a Cultura Organizacional Influencia a Evolução de BPM	JATOBÁ, I.; ALVES, C. (2015)	12
188	Livre Saber (LiSa): Um Repositório de Recursos Educacionais Abertos de Cursos a Distância	OTSUKA, J. L. <i>et al.</i> (2015)	1, 2
189	EaD entre os ditames legais e a realidade concreta	PASCHOALINO, J. B. Q. <i>et al.</i> (2015)	2
190	Uso de Pesquisa Bibliográfica em Informática na Educação: um Mapeamento Sistemático	DETROZ, J. P. <i>et al.</i> (2015)	2
191	Formalização e Validação de Padrões para Apoiar o Design de Sistemas Educacionais com Coautoria	SILVA, M. A. R.; ANACLETO, J. C. (2015)	9, 12
192	Orientações para o Design de hipermídias para aprendizagem da língua espanhola na EaD	NUNES, J. V.; GONÇALVES, B. S. (2015)	2
193	PERSONNA: proposta de ontologia de contexto e perfil de aluno para recomendação de objetos de aprendizagem	REZENDE, P. <i>et al.</i> (2015)	2
194	Uma análise exploratória de tópicos de pesquisa emergentes em Informática na Educação	BORGES, V. A.; NOGUEIRA, B. M.; BARBOSA, E. F. (2015)	2
195	Estudo de Evasão no Curso de Ciência da Computação da UFRGS	RODRIGUES, F. S.; BRACKMANN, C. P.; BARONE, D. A. C. (2015)	2
196	Atividades Extraclasse com base no Currículo	RODRIGUES, C. A.; ELIA, M.	2

	Mínimo para a Língua Inglesa usando uma Rede Social	F. (2015)	
197	Desenvolvimento de uma Ferramenta para a Construção e Integração de Personagens Virtuais Animados com Voz Sintética aos Materiais Didáticos para EAD	MACIEL, A. M. A.; RODRIGUES, R. L.; CARVALHO FILHO, E. C. B. (2015)	1, 2, 9
198	Um Estudo Empírico Exploratório em Confiabilidade de Sistemas Operacionais	ANTUNES, M. P.; MATIAS JÚNIOR, R. (2014)	11
199	Um modelo baseado na evolução temporal de consumo e sua aplicação em domínios de recomendação	ARAÚJO, C. S.; MOURÃO, F. H.; MEIRA JR., W. (2014)	11
200	Implementação de Difusão Atômica Baseada em Diagnóstico com Testes Imperfeitos	CAMARGO, E. T.; DUARTE JR., E. P.; PIETNICZKA, W. C. (2014)	

APÊNDICE B – Tabela de apoio ao cálculo da precisão média interpolada em 11 pontos

Referente ao resultado do modelo Vetorial sob a expressão de busca **cons1**. Utilizou-se um *software* aplicativo para manipulação de planilhas e letrônicas.

ID DOCs REL.	POS.	POS REL.	ID DOCs REC. ■ = relevante	COBERTURA	PRECISÃO para esta cobertura
7	1		109	0	
10	2		88	0	
12	3	1	197	0,029411765	0,333333333
13	4		148	0,029411765	
23	5		126	0,029411765	
24	6		179	0,029411765	
25	7	2	170	0,058823529	0,285714286
27	8	3	153	0,088235294	0,375
33	9		62	0,088235294	
44	10		136	0,088235294	
45	11	4	80	0,117647059	0,363636364
48	12		38	0,117647059	
51	13		147	0,117647059	
52	14		68	0,117647059	
53	15		102	0,117647059	
75	16		65	0,117647059	
79	17		145	0,117647059	
80	18		151	0,117647059	
82	19		194	0,117647059	
83	20		28	0,117647059	
85	21		26	0,117647059	
106	22	5	176	0,147058824	0,227272727
117	23		99	0,147058824	
125	24	6	125	0,176470588	0,25
128	25	7	52	0,205882353	0,28
139	26		74	0,205882353	
153	27		16	0,205882353	
170	28		180	0,205882353	
172	29		130	0,205882353	
174	30		138	0,205882353	

176
181
188
197

31		198	0,205882353	
32		143	0,205882353	
33		159	0,205882353	
34	8	51	0,235294118	0,235294118
35		67	0,235294118	
36		93	0,235294118	
37		157	0,235294118	
38	9	10	0,264705882	0,236842105
39	10	174	0,294117647	0,256410256
40		158	0,294117647	
41		183	0,294117647	
42		131	0,294117647	
43		141	0,294117647	
44		101	0,294117647	
45		113	0,294117647	
46	11	7	0,323529412	0,239130435
47		166	0,323529412	
48		40	0,323529412	
49		42	0,323529412	
50		61	0,323529412	
51		142	0,323529412	
52		161	0,323529412	
53		171	0,323529412	
54		72	0,323529412	
55		54	0,323529412	
56		173	0,323529412	
57		94	0,323529412	
58	12	23	0,352941176	0,206896552
59		3	0,352941176	
60	13	139	0,382352941	0,216666667
61		41	0,382352941	
62		190	0,382352941	
63		162	0,382352941	
64		46	0,382352941	
65		185	0,382352941	
66		124	0,382352941	
67		34	0,382352941	
68		70	0,382352941	

69		133	0,382352941	
70		77	0,382352941	
71	14	172	0,411764706	0,197183099
72		11	0,411764706	
73		14	0,411764706	
74	15	12	0,441176471	0,202702703
75		30	0,441176471	
76		187	0,441176471	
77		87	0,441176471	
78		47	0,441176471	
79		37	0,441176471	
80		154	0,441176471	
81		43	0,441176471	
82		116	0,441176471	
83	16	48	0,470588235	0,192771084
84		57	0,470588235	
85		160	0,470588235	
86	17	45	0,5	0,197674419
87	18	13	0,529411765	0,206896552
88		192	0,529411765	
89		90	0,529411765	
90		92	0,529411765	
91		36	0,529411765	
92		31	0,529411765	
93		91	0,529411765	
94	19	117	0,558823529	0,20212766
95		95	0,558823529	
96		149	0,558823529	
97		103	0,558823529	
98	20	128	0,588235294	0,204081633
99		21	0,588235294	
100		122	0,588235294	
101		177	0,588235294	
102		118	0,588235294	
103	21	53	0,617647059	0,203883495
104		66	0,617647059	
105		184	0,617647059	
106		35	0,617647059	

107		186	0,617647059	
108	22	33	0,647058824	0,203703704
109		123	0,647058824	
110		55	0,647058824	
111		86	0,647058824	
112	23	85	0,676470588	0,205357143
113		63	0,676470588	
114		114	0,676470588	
115		195	0,676470588	
116		110	0,676470588	
117		150	0,676470588	
118		167	0,676470588	
119		163	0,676470588	
120		49	0,676470588	
121		8	0,676470588	
122		76	0,676470588	
123	24	181	0,705882353	0,195121951
124		137	0,705882353	
125	25	79	0,735294118	0,2
126		168	0,735294118	
127		135	0,735294118	
128		182	0,735294118	
129		59	0,735294118	
130		84	0,735294118	
131		2	0,735294118	
132		73	0,735294118	
133		119	0,735294118	
134		169	0,735294118	
135		29	0,735294118	
136		196	0,735294118	
137		108	0,735294118	
138		146	0,735294118	
139	26	25	0,764705882	0,18705036
140	27	24	0,794117647	0,192857143
141		98	0,794117647	
142		155	0,794117647	
143		22	0,794117647	
144		156	0,794117647	

145		127	0,794117647	
146		144	0,794117647	
147		140	0,794117647	
148		178	0,794117647	
149		199	0,794117647	
150		1	0,794117647	
151	28	188	0,823529412	0,185430464
152	29	82	0,852941176	0,190789474
153		105	0,852941176	
154		100	0,852941176	
155	30	83	0,882352941	0,193548387
156		71	0,882352941	
157		97	0,882352941	
158		4	0,882352941	
159		120	0,882352941	
160	31	44	0,911764706	0,19375
161	32	75	0,941176471	0,198757764
162		18	0,941176471	
163		15	0,941176471	
164		6	0,941176471	
165		9	0,941176471	
166		5	0,941176471	
167		78	0,941176471	
168		56	0,941176471	
169		193	0,941176471	
170		104	0,941176471	
171		164	0,941176471	
172		32	0,941176471	
173		191	0,941176471	
174		81	0,941176471	
175		132	0,941176471	
176		165	0,941176471	
177		200	0,941176471	
178		115	0,941176471	
179		64	0,941176471	
180		69	0,941176471	
181		50	0,941176471	
182		39	0,941176471	

183	33	27	0,970588235	0,180327869
184		89	0,970588235	
185	34	106	1	0,183783784
186		152	1	
187		175	1	
188		112	1	
189		111	1	
190		107	1	
191		189	1	