UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO MULTIDISCIPLINAR

PIERRE ROSSI CARRIONE

Sistema de Gestão e Gerenciamento de Documentos Acadêmicos

Prof. Filipe Braida do Carmo, D.Sc.
Orientador

Nova Iguaçu, Maio de 2024

Sistema de Gestão e Gerenciamento de Documentos Acadêmicos

Pierre Rossi Carrione

Projeto Final de Curso submetido ao Departamento de Ciência da Computação do Instituto Multidisciplinar da Universidade Federal Rural do Rio de Janeiro como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Apresentado por:

Pierre Rossi Carrione

Aprovado por:

Prof. Filipe Braida do Carmo, D.Sc.

Prof. Bruno José Dembogurski, D.Sc.

Prof. Marcel William Rocha da Silva, D.Sc.

NOVA IGUAÇU, RJ - BRASIL

Maio de 2024

DOCUMENTOS COMPROBATÓRIOS Nº 9155/2024 - CoordCGCC (12.28.01.00.00.98)

(Nº do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 04/07/2024 09:49)
BRUNO JOSE DEMBOGURSKI
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: ###249#4

(Assinado digitalmente em 03/07/2024 19:56)
FILIPE BRAIDA DO CARMO
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: ###295#4

(Assinado digitalmente em 03/07/2024 20:07)
MARCEL WILLIAM ROCHA DA SILVA
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: ###807#6

(Assinado digitalmente em 04/07/2024 10:41)
PIERRE ROSSI CARRIONE
DISCENTE
Matrícula: 2015#####3

*Visualize o documento original em https://sipac.ufrrj.br/documentos/ informando seu número: **9155**, ano: **2024**, tipo: **DOCUMENTOS COMPROBATÓRIOS**, data de emissão: **03/07/2024** e o código de verificação: **2e857fd48f***

Agradecimentos

Quero, primeiramente, agradecer a Deus por nunca me desamparar e por me permitir chegar até aqui. Gostaria também de expressar minha gratidão à minha mãe, Schirley, por todo o apoio dado ao longo dessa jornada.

Agradeço à minha noiva, Emanuele, que esteve presente em todos os momentos desta caminhada. Seu apoio, confiança e acolhimento foram fundamentais para que eu superasse todos os desafios e chegasse até aqui.

Gostaria de dedicar um agradecimento especial à minha segunda mãe, Shirlene. Por todo o carinho, incentivo e apoio que você me deu, mesmo não estando mais aqui. Saiba que sua presença foi fundamental, e que essa conquista também é sua.

Agradeço aos meus amigos por me incentivarem e ajudarem a chegar até aqui. Todos os momentos de descontração, as noites no Discord, os viradões, tornaram o processo menos doloroso. Deixo minha gratidão também aos amigos que fiz na Rural, pelo apoio e por todos os bons momentos ao longo dessa trajetória.

Por fim, gostaria de expressar minha profunda gratidão a todo o Departamento de Ciência da Computação da UFRRJ-IM, em especial ao meu orientador, Filipe Braida. Sua admiração e motivação foram constantes durante todo o trabalho, sempre muito solicito e compreensivo.

Meus sinceros agradecimentos a todos vocês.

RESUMO

Sistema de Gestão e Gerenciamento de Documentos Acadêmicos

Pierre Rossi Carrione

Maio/2024

Orientador: Filipe Braida do Carmo, D.Sc.

Os avanços tecnológicos transformaram significativamente a maneira pela qual as informações são obtidas. Com a transição do formato físico para o digital, seja em computadores ou celulares, tornou-se possível adquirir conhecimento de forma rápida e acessível, independentemente do local. No entanto, apesar da facilidade proporcionada pela internet e pelos motores de busca na aquisição de informações em geral, o acesso a conteúdo acadêmico ainda costuma ser fragmentado e disperso em diversas fontes, além de algumas instituições restringirem ou cobrarem pelo acesso. Diante desse cenário, este trabalho propõe um sistema web que não apenas permite aos usuários gerir e recuperar conhecimento de maneira rápida e gratuita, mas também armazenar documentos acadêmicos. Nesse sentido, atua como um repositório centralizado, facilitando a busca e recuperação de documentos acadêmicos, ao mesmo tempo que incentiva o compartilhamento de conhecimento e contribui para a construção de uma base coletiva.

ABSTRACT

Sistema de Gestão e Gerenciamento de Documentos Acadêmicos

Pierre Rossi Carrione

Maio/2024

Advisor: Filipe Braida do Carmo, D.Sc.

Technological advances have significantly transformed the way information is obtained. With the transition from physical to digital formats, whether on computers or mobile devices, it has become possible to acquire knowledge quickly and conveniently, regardless of location. However, despite the convenience provided by the internet and search engines in accessing information in general, access to academic content still tends to be fragmented and scattered across various sources, with some institutions restricting or charging for access. In this scenario, this work proposes a web system that not only allows users to manage and retrieve knowledge quickly and free of charge but also to store academic documents. In this sense, it acts as a centralized repository, facilitating the search and retrieval of academic documents, while also encouraging knowledge sharing and contributing to the development of a collective knowledge base.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

IR	Information Retrieval
IRS	Information Retrieval Systems
TF	Term Frequency
IDF	Inverse Document Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
SC	Similarity Coefficient
SGD	Sistema de Gestão de Documentos
RF	Requisitos Funcionais
RNF	Requisitos Não Funcionais
RN	Regra de Negócios
UC	Casos de Uso
SGBD	Sistema de Gerenciamento de Banco de Dados
ORM	Object-Relational Mapping
MVC	Model-View-Controller
NPM	Node Package Manage
ER	Entidade-Relacionamento
MIT	Massachusetts Institute of Technology
HP	Hewlett-Packard
UFMG	Universidade Federal de Minas Gerais
UFRJ	Universidade Federal do Rio de Janeiro

UFRRJ	Universidade Federal Rural do Rio de Janeiro
GNU	General Public License
OPAC	Online Public Access Catalog
PDF	Portable Document Format

Sumário

Capítulo 1

Introdução

Ao longo das últimas décadas, a evolução tecnológica revolucionou a forma como lidamos com documentos, superando os desafios da organização manual e do armazenamento físico (JORDAN; ZABUKOVŠEK; KLANČNIK, 2022). Com a popularização dos computadores pessoais e da internet, surgiu a capacidade de armazenar uma grande quantidade de informações eletronicamente, levando a um notável crescimento na produção e distribuição de documentos digitais. Essa mudança não apenas otimizou a criação e distribuição de documentos, mas também revolucionou a eficiência e a preservação da informação (BABAN; MOKHTAR, 2010).

Esses avanços tecnológicos também reconfiguraram a maneira como adquirimos informações. Agora, por meio de motores de busca como o *Google*[1], é possível ter acesso a uma ampla variedade de informações, localizar pessoas e organizações, resumo de notícias e eventos, além de simplificar a comparação de preços. Esse cenário tornou-se possível devido aos serviços de recuperação da informação, desempenhando um papel vital ao proporcionar acesso a essa diversidade de informações. (BÜTTCHER; CLARKE; CORMACK, 2010)

Esse cenário foi possível devido à Information Retrieval (IR), também conhecida como Recuperação da Informação traduzido para o português, uma área indispensável para lidar com a complexidade inerente à busca eficiente. A IR concentra-se na

[1] <https://www.google.com>

representação, pesquisa e manipulação de extensas coleções de texto eletrônico e outros dados em linguagem humana. (BÜTTCHER; CLARKE; CORMACK, 2010)

Os Sistemas de Recuperação da Informação, do inglês Information Retrieval Systems (IRS), também contribuíram nesse cenário. Os IRS são ferramentas ou softwares projetados para facilitar o processo de Recuperação da Informação. Estão amplamente difundidos pelo mundo, tornando-se uma ferramenta essencial para milhões de pessoas que dependem de seus serviços diariamente para auxiliar em atividades de trabalho, educação e entretenimento. (BÜTTCHER; CLARKE; CORMACK, 2010)

Apesar da disseminação dos computadores pessoais e da internet, ainda há desafios significativos na busca e recuperação de informações. A sobrecarga de informações, a falta de filtragem de conteúdo relevante e a necessidade de acesso a recursos de qualidade são apenas alguns dos obstáculos enfrentados pelos usuários (STANLEY, 2021). Esses desafios se tornam ainda mais evidentes quando se trata de informações acadêmicas, uma vez que algumas instituições restringem ou cobram pelo acesso (WILLINSKY, 2006). Além disso, a complexidade e o custo associados à implementação de sistemas de gestão de documentos acadêmicos também representam obstáculos na busca pela informação.

Diante desses desafios, a implementação de um sistema de gestão de documentos acadêmicos torna-se imperativa. Este trabalho propõe o desenvolvimento de um sistema que visa facilitar a recuperação e o armazenamento de documentos acadêmicos, oferecendo uma solução para os problemas enfrentados pelos usuários ao buscar informações online. Além de proporcionar maior eficiência na gestão de documentos, o sistema busca incentivar o compartilhamento de conhecimento e contribuir para a construção de uma base coletiva de informações, impulsionando a inovação e a pesquisa abrangente no âmbito universitário.

1.1 Objetivo

O objetivo deste trabalho é propor e implementar um sistema de gestão de documentos acadêmicos com o propósito de facilitar a recuperação e o armazenamento

desses documentos. Além disso, busca-se promover o compartilhamento de conhecimento e contribuir para a construção de uma base coletiva, impulsionando, assim, a inovação e a pesquisa abrangente no âmbito universitário. Com isso, qualquer pessoa previamente cadastrada será capaz de encontrar, em um único local, uma ampla variedade de documentos acadêmicos, incluindo artigos, teses, TCCs, entre outros.

Para alcançar esse propósito, o trabalho foi dividido em:

- Identificar e definir os requisitos do sistema.
- Projetar a arquitetura do sistema.Definir os layouts das interfaces, design do banco de dados e lógica de negócios.
- Com base nos requisitos e na arquitetura, fazer o desenvolvimento do sistema e realizar testes.
- Tornar o código-fonte aberto para melhorias e adição de novas funcionalidades.

1.2 Organização do Trabalho

Este trabalho está organizado da seguinte forma:

- Capítulo 2: Será descrito todos os fundamentos teóricos que embasam esse trabalho, contextualizando os conceitos de gestão de documentos e recuperação da informação.
- Capítulo 3: Será apresentada a motivação desse projeto, trabalhos relacionados e a modelagem adotada para esse sistema.
- Capítulo 4: Neste capítulo será abordado sobre as ferramentes utilizadas, o desenvolvimento do projeto e sua finalização.
- Capítulo 5: Serão apresentadas as considerações, abordando as limitações e possíveis continuações que podem partir deste trabalho.

Capítulo 2

Fundamentação

Para um melhor entendimento do trabalho realizado, este capítulo tem como objetivo aprofundar-se nos conceitos e técnicas aplicados durante a construção desta pesquisa. Inicialmente, abordaremos a gestão eletrônica de documentos, discutindo sua história e importância. Em seguida, exploraremos a Recuperação da Informação, detalhando seus fundamentos, modelos e técnicas de avaliação.

2.1 Gestão Eletrônica de Documento

O registro de informações por escrito remonta a aproximadamente 3.000 a.C., quando os sumérios utilizavam tabuletas de argila inscritas com cuneiformes em espaços específicos. Desde então, reconheceu-se a importância de organizar e acessar arquivos de maneira eficiente para garantir o uso eficaz da informação, levando ao desenvolvimento de classificações especiais para identificar cada tabuleta e seu conteúdo. (SINGHAL et al., 2001)

Com os avanços da tecnologia e a popularização dos computadores pessoais, a forma como armazenamos informações passou por uma transformação significativa. A transição do formato físico para o eletrônico trouxe consigo a capacidade de armazenar volumes massivos de dados de maneira eficiente e acessível. Essa transformação não apenas otimizou a criação e distribuição de documentos, mas também revolucionou

a eficiência e a preservação da informação. (BABAN; MOKHTAR, 2010)

Os progressos significativos na eletrônica e informática, nos quais a tecnologia se integra em todos os setores da sociedade, resultaram no surgimento do que ficou conhecido como a era da informação (MITRA; CHAUDHURI, 2000). A informação e o conhecimento tornaram-se ativos essenciais, moldando a dinâmica social, econômica e cultural. A compreensão desse cenário é crucial para uma participação ativa na sociedade moderna (ADHIKARI, 2010).

Os sistemas gestores da informação contribuíram para esse cenário. No entanto, apesar de sua grande relevância, sua base fundamental de funcionamento reside nas técnicas e mecanismos de recuperação da informação. Na seção a seguir, abordaremos toda a história, técnicas, mecanismos e métricas sobre a Recuperação da Informação.

2.2 Recuperação da Informação

A disciplina da Recuperação de Informação possui raízes históricas, remontando a aproximadamente 4.000 anos. Durante esse período, a humanidade organizou informações para facilitar sua posterior recuperação e utilização. Antigos romanos e gregos registravam informações em rolos de papiro, muitos acompanhados por etiquetas contendo resumos breves para aprimorar a pesquisa. Os primeiros índices surgiram em pergaminhos gregos no século II a.C. (CERI et al., 2013)

A prática da busca informatizada teve início no final da década de 1940, impulsionada pelo aumento na produção de literatura científica, especialmente relatórios técnicos, em vez de artigos tradicionais. A disponibilidade crescente de computadores despertou o interesse na recuperação automática de documentos. No entanto, naquela época, a recuperação de documentos dependia principalmente de autor, título e palavras-chave. A pesquisa de texto completo só veio a surgir anos depois. (MANNING; RAGHAVAN; SCHÜTZE, 2009)

O termo “recuperação de informação” foi introduzido por volta de 1952, embora em 1945 Bush já falava sobre um “dispositivo no qual um indivíduo armazena todos os seus livros, registros e comunicações, e que é mecanizado para que possa ser

consultado com extrema velocidade e flexibilidade" (GÖKER; DAVIES, 2009). O primeiro representante de repositórios de documentos computadorizados para pesquisa foi o *Cornell SMART System*, desenvolvido na década de 1960. Os primeiros sistemas de IR foram usados principalmente por bibliotecários especializados como sistemas de recuperação de referência em modalidades de lote. Muitas bibliotecas ainda utilizam hierarquias de categorização para classificar seus volumes (CERI et al., 2013).

A recuperação de informações, de maneira simples, envolve a localização de dados. Nesse contexto, a IR dedica-se a representar, pesquisar e manipular extensas coleções de texto eletrônico e outros dados na linguagem humana (BÜTTCHER; CLARKE; CORMACK, 2010). A essência da recuperação de informação pode ser explicada por meio do seguinte problema típico: por um lado, temos uma pessoa com uma necessidade de informação. Essa necessidade, inicialmente vaga, precisa ser de alguma forma articulada em uma consulta que a descreva. Por outro lado, há um sistema que busca fornecer resultados que correspondam adequadamente à consulta (GÖKER; DAVIES, 2009).

Com a chegada da internet e da *World Wide Web* (abreviada como Web), as técnicas de recuperação de informação ganharam não apenas um impacto prático significativo, mas também uma importância teórica relevante (DOMINICH, 2008). A Web está se transformando em um repositório universal de conhecimento e cultura humana, possibilitando um compartilhamento sem precedentes de ideias e informações em uma escala nunca antes vista (BAEZA-YATES; RIBEIRO-NETO, 1999). Atualmente, o cenário mundial passou por mudanças substanciais, com centenas de milhões de pessoas recorrendo à recuperação de informação diariamente, seja ao utilizar um motor de busca na Web ou ao pesquisar em seus e-mails (MANNING; RAGHAVAN; SCHÜTZE, 2009).

Diante desse cenário, os sistemas de recuperação da informação tornaram-se uma parte essencial em todo tipo de organização. Um sistema de Recuperação de Informação é um sistema que recebe informações, converte-as em um formato pesquisável e oferece uma interface que possibilita que um usuário realize buscas e recupere as informações. Um sistema bastante difundido mundialmente é o *Google*

(KOWALSKI, 2011).

Um Sistema de Recuperação de Informações é constituído pelo hardware e software que possibilitam ao usuário encontrar as informações necessárias. Esse sistema não se resume a um único aplicativo, mas é composto por diversos aplicativos distintos que operam de maneira conjunta para disponibilizar as ferramentas e funções essenciais, auxiliando os usuários na obtenção de respostas às suas perguntas. O objetivo primordial de um sistema de recuperação de informação é reduzir o tempo que o usuário leva durante o processo de localização de informações valiosas. (KOWALSKI, 2011)

Diante do panorama histórico e da evolução dos sistemas de recuperação de informação, é crucial compreender os conceitos e fundamentos que sustentam essa disciplina em constante desenvolvimento. Ao longo dos anos, a busca pela organização eficiente e recuperação acessível de dados tem moldado a maneira como interagimos com a informação, desde os primeiros registros em tabuletas de argila até a era digital e a ubiquidade da internet. Essa jornada revela a necessidade contínua de estratégias aprimoradas e sistemas inovadores para lidar com a crescente complexidade e volume de dados.

Iniciar a exploração dos conceitos fundamentais da recuperação de informação nos leva a mergulhar nas estratégias de representação, indexação e busca, que formam a base para a eficácia desses sistemas. Compreender como as informações são organizadas, convertidas em formatos pesquisáveis e apresentadas aos usuários é crucial para explorar a interseção entre tecnologia, linguagem humana e a crescente demanda por acesso rápido e preciso às informações.

No contexto apresentado, nas próximas seções, serão abordados os principais fundamentos da recuperação, destacando suas técnicas e limitações. Além disso, serão discutidos os princípios-chave da recuperação de informação, analisando como esses sistemas operam, desde a formulação de consultas até a entrega de resultados relevantes, bem como suas métricas de avaliação.

2.3 Fundamentos da Recuperação da Informação

Nesta seção, serão abordados os princípios fundamentais da Recuperação de Informações, explorando cada componente desta disciplina essencial para o tratamento eficiente de grandes volumes de dados. Desde as etapas iniciais do processamento do texto até os modelos de busca, examinaremos cada aspecto que compõe o arcabouço da IR. Por fim, discutiremos as métricas que avaliam a eficácia dos sistemas, detalhando cada elemento que contribui para o alicerce robusto da IR.

2.3.1 Representação do Documento

No contexto da IR, um documento é considerado como uma unidade de informação, que pode abranger desde artigos e páginas da web até livros e outras fontes textuais. Essa visão dos documentos estabelece a base para entendermos como a informação é estruturada e, posteriormente, recuperada (BÜTTCHER; CLARKE; CORMACK, 2010). A Figura 2.1 ilustra um conjunto de documentos que será utilizado para exemplificar conjuntos de indexação e cálculos nos parágrafos seguintes.

Figura 2.1: Exemplo de uma coleção de documentos contendo os documentos d_1, d_2, d_3 e d_4.

Fonte: Adaptada de Baeza-Yates e Ribeiro-Neto (2011).

Os modelos tradicionais de IR pressupõem que cada documento pode ser caracterizado por um conjunto de palavras-chave representativas, conhecidas como termos de indexação. Esses termos, por sua vez, desempenham um papel crucial no processo de organização e busca eficiente por informações relevantes. É fundamental reconhecer que um termo de indexação pode ser tanto uma palavra específica quanto um grupo de palavras consecutivas em um documento. (KOWALSKI, 2011)

Em sua forma mais abrangente, um termo de indexação é qualquer palavra

presente na coleção de documentos. De acordo com Baeza-Yates e Ribeiro-Neto (2013), a representação de um documento ou consulta pode ser expressa da seguinte maneira: considerando t como número de termos de indexação em uma coleção de documentos e k_i como i-ésimo termo desse índice. O vocabulário $V = \{k_1, k_2, k_3, ..., k_t\}$ é o conjunto de todos os termos de indexação distintos presentes na coleção.

Cada padrão de ocorrência de termos é denominado componente conjuntiva do termo , a qual está associada de maneira única a cada documento, como $c(d_j)$, ou à consulta, como $c(q)$. Como ilustrado na Figura 2.2, a componente conjuntiva para o documento com o primeiro termo k_1 e segundo termo k_2 indexado é $c(d_j) = \{1, 0, 1, \ldots, 0\}$, enquanto que para uma consulta com todos os termos indexados, seria $c(q) = \{1, 1, 1, \ldots, 1\}$. (BAEZA-YATES; RIBEIRO-NETO, 2013)

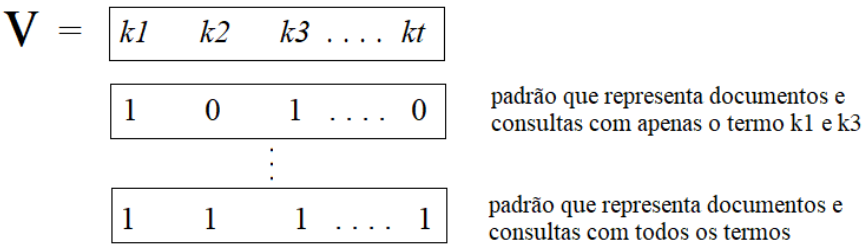


Figura 2.2: Representação lógica do documento na forma de vetor, em que 1 indica a presença do termo k_t e 0 sua ausência.

Fonte: Adaptada de Baeza-Yates e Ribeiro-Neto (2013).

A Tabela 2.1 ilustra um exemplo considerando as frases: "O céu está azul", "As nuvens são brancas" e "O sol está brilhando no céu". Observa-se que, para uma consulta $c(q)$ de todas as palavras para o documento 3 a resposta é o vetor {1, 1, 1, 0, 0, 0, 0, 0, 1, 1}, no qual apenas as palavras "o", "céu", "está", "sol" e "brilhando" estão presentes neste documento.

	o	céu	está	azul	as	nuvens	são	brancas	sol	brilhando	no
Documento 1	1	1	1	1	0	0	0	0	0	0	0
Documento 2	0	0	0	0	1	1	1	1	0	0	0
Documento 3	1	1	1	0	0	0	0	0	1	1	1

Tabela 2.1: Matriz de incidência termo-documento, na qual cada linha representa um vetor e cada coluna corresponde a uma palavra única ou *token* presente nas três frases mencionadas anteriormente. O valor 1 indica a presença do *token* no documento, enquanto 0 indica sua ausência.

Fonte: O autor.

A indexação de termos emerge como uma peça-chave no cenário da IR por várias razões. Ao lidar com grandes volumes de documentos, a indexação é fundamental para organizar e facilitar buscas eficientes por informações relevantes. Os termos de indexação são escolhidos com cuidado para representar de maneira eficaz o conteúdo dos documentos, contribuindo para a organização e categorização que impulsionam consultas futuras. A precisão na escolha desses termos é vital para garantir uma recuperação de informações eficiente e direcionada. (BÜTTCHER; CLARKE; CORMACK, 2010)

Os termos de indexação podem ser extraídos diretamente do texto dos documentos ou indicados por intervenção humana. Os computadores atuais permitem representar um documento por todas as palavras que o compõe, chamada de representação *full-text*. No entanto, em face de coleções extensas, o custo computacional associado a essa abordagem pode ser significativo, tornando desejável a redução do conjunto de palavras representativas. (BAEZA-YATES; RIBEIRO-NETO, 2013)

2.3.2 Processamento do texto

O processamento do texto é uma etapa fundamental no processo de recuperação da informação, desempenhando um papel crucial na análise e compreensão de documentos textuais. Esse processo envolve a quebra de um texto em unidades

menores, conhecidas como *tokens*[1], que podem variar desde palavras até caracteres, dependendo do contexto da análise. Ao transformar um texto em *tokens*, facilita-se a manipulação e interpretação do conteúdo textual, permitindo uma abordagem mais refinada na busca e organização de informações. (BÜTTCHER; CLARKE; CORMACK, 2010)

De acordo com Baeza-Yates e Ribeiro-Neto (2011) esse processo pode ser divido em cinco operações:

1. Realização da análise lexical no texto visando lidar com dígitos, hífens, sinais de pontuação e diferenças entre maiúsculas e minúsculas.

2. Remoção de palavras irrelevantes com o propósito de filtrar aquelas com baixo poder discriminatório para fins de recuperação.

3. Derivação das palavras restantes para eliminar afixos e facilitar a busca de documentos que contenham variações sintáticas dos termos de consulta.

4. Seleção de termos de índice para determinar quais palavras (*stems*) serão usados como elementos de indexação. Geralmente, a decisão de incluir uma palavra específica como termo de índice está relacionada à sua natureza sintática, sendo que substantivos tendem a carregar mais carga semântica em comparação com adjetivos, advérbios e verbos.

5. Estabelecimento de estruturas de categorização de termos, como tesauros ou extração de estrutura diretamente do texto, para permitir a expansão da consulta original com termos relacionados.

A Figura 2.3 demonstra uma perspectiva lógica de um documento ao longo das distintas etapas do pré-processamento de texto. A seguir abordaremos cada operação com mais detalhes.

[1] *token*: uma unidade de uma sequência de caracteres. (MANNING; NAYAK, Acesso em 30/01/2024)

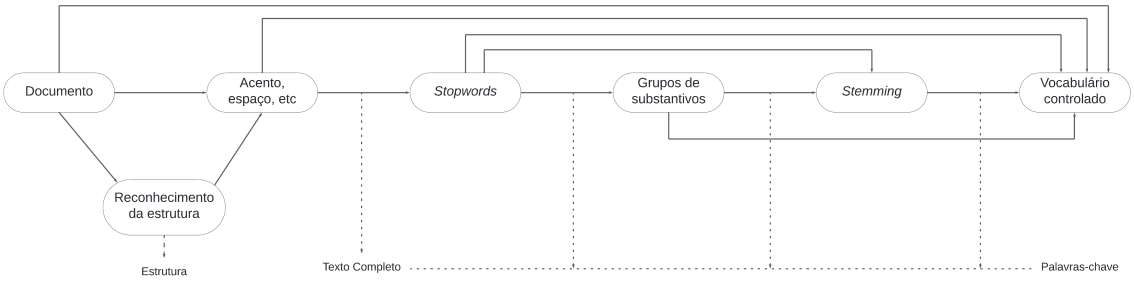


Figura 2.3: Perspectiva lógica de um documento durante as fases de pré-processamento de texto.

Fonte: Adaptado de Baeza-Yates e Ribeiro-Neto (2011).

2.3.2.1 Análise Léxica e Tokenização

A análise lexical no âmbito da recuperação da informação refere-se ao processo de examinar e compreender as palavras e sua estrutura dentro de um texto. Isso é fundamental para identificar os termos relevantes usados na indexação de documentos. Durante a análise lexical, os documentos são tokenizados, ou seja, divididos em unidades menores chamadas *tokens*, os quais podem representar palavras, frases ou caracteres. (BAEZA-YATES; RIBEIRO-NETO, 2011)

Essa análise lexical contribui para a criação de índices eficientes, onde os termos relevantes são associados aos documentos. Esses índices são utilizados pelos sistemas de recuperação da informação para identificar documentos pertinentes em resposta às consultas dos usuários, tornando o processo de busca mais preciso e eficaz. (BAEZA-YATES; RIBEIRO-NETO, 2011)

A tokenização no contexto da recuperação da informação reside na capacidade de representar o conteúdo de documentos de maneira mais granular, chamada de *tokens*. Ao dividir um texto em unidades significativas, a tokenização permite que sistemas de indexação e motores de busca identifiquem e processem palavras-chave relevantes de forma mais eficiente. Essa abordagem refinada contribui para a precisão na correspondência de consultas, melhorando a relevância dos resultados apresentados aos usuários. (MANNING; RAGHAVAN; SCHÜTZE, 2009)

Nesse contexto, a tokenização desempenha um papel crucial em lidar com desafios

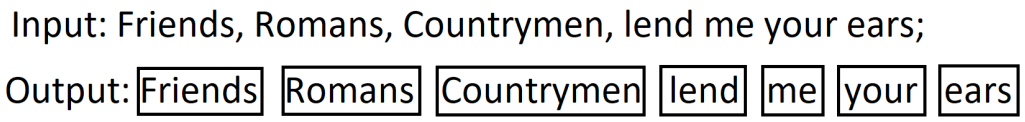


Figura 2.4: Exemplo de tokenização. A frase representada no input é dividida em tokens individuais e cada token é representado em um quadrado no output.

Fonte: Adaptada Manning, Raghavan e SchÜtze (2009).

linguísticos, como a flexão verbal e as diferentes formas de uma palavra. Ao transformar palavras em *tokens*, é possível normalizar a representação textual, garantindo que diferentes formas de uma palavra sejam tratadas de maneira uniforme. Essa normalização é essencial para superar variações linguísticas e garantir uma recuperação da informação mais abrangente e precisa. (BÜTTCHER; CLARKE; CORMACK, 2010)

2.3.2.2 Stop Words

A linguagem humana contém uma abundância de palavras funcionais, termos que têm pouco significado por si só e geralmente servem para complementar outras palavras. Entre os mais utilizados estão os determinantes, como “o”, “um”, “uma”, “aquilo” e “aqueles”. Essas palavras desempenham um papel essencial na descrição de substantivos no texto e na expressão de conceitos como localização ou quantidade. Além dos artigos, temos as preposições que ligam dois substantivos. (CROFT; METZLER; STROHMAN, 2015)

Apesar de desempenharem papel importante na linguagem, para a recuperação da informação essas palavras são consideradas comuns e pouco informativas, sendo removidas durante o processo de indexação para reduzir a complexidade computacional e melhorar a precisão dos resultados. A essas palavras ou termos é dado o nome de *stop words*. (MITRA; CHAUDHURI, 2000)

Ao eliminar essas palavras durante a etapa de pré-processamento, os sistemas de recuperação da informação conseguem focar em termos mais relevantes, como substantivos e adjetivos, proporcionando resultados mais precisos e relevantes para o usuário. A remoção de *stop words* também contribui para a economia de recursos

computacionais, uma vez que reduz a quantidade de informações a serem processadas. Isso é particularmente relevante em grandes bases de dados textuais, onde a eficiência na busca é crucial. (BAEZA-YATES; RIBEIRO-NETO, 2011)

2.3.2.3 Stemming

A riqueza da linguagem natural reside, em parte, na variedade de formas de expressar uma ideia singular. Esse aspecto pode representar um desafio para os motores de busca, que dependem da correspondência exata de palavras para localizar documentos pertinentes. Em vez de limitar as correspondências apenas a palavras idênticas, várias técnicas foram desenvolvidas para capacitar os motores de busca a identificar palavras semanticamente relacionadas. (CROFT; METZLER; STROHMAN, 2015)

A técnica de *stemming* desempenha um papel crucial na recuperação da informação, visando simplificar as palavras aos seus radicais, o que ajuda a agrupar diferentes formas de uma palavra sob uma única representação. Por exemplo, ao aplicar *stemming*, palavras como "correr", "corre", "correndo" são reduzidas ao radical "corr". Isso é especialmente valioso em sistemas de recuperação de informação, onde a consistência na representação de palavras aprimora a correspondência entre consultas dos usuários e os termos presentes nos documentos indexados. (BÜTTCHER; CLARKE; CORMACK, 2010)

Essa técnica não apenas economiza recursos computacionais, reduzindo a redundância no índice, mas também melhora a abrangência da pesquisa. Em situações onde um usuário busca por "correr", o *stemming* permite que documentos que contenham variações da palavra, como "corre" ou "corrido", sejam considerados relevantes. Essa flexibilidade na correspondência é crucial para garantir uma recuperação de informação abrangente. (BÜTTCHER; CLARKE; CORMACK, 2010)

2.3.2.4 Seleção de palavras-chave

Baeza-Yates e Ribeiro-Neto (2011) explicam que, ao adotar a representação do texto completo, todas as palavras presentes no texto serão utilizadas como termos de índice, ou seja, cada termo será indexado. Uma alternativa é adotar uma abordagem mais abstrata, na qual nem todas as palavras são consideradas como termos de índice. Isso implica que a escolha do conjunto de termos a serem utilizados como índices precisa ser criteriosa.

Outra abordagem possível é a seleção automática de candidatos para termos de índice. Um método viável consiste na identificação de grupos de substantivos. Naturalmente, uma frase em linguagem natural é composta por diversos elementos, como substantivos, pronomes, artigos, verbos, adjetivos, advérbios e conectivos. Embora cada classe gramatical desempenhe funções específicas, argumenta-se que a maior carga semântica muitas vezes está nos substantivos.

Portanto, uma estratégia intuitiva e promissora para a seleção automática de termos de índice é utilizar os substantivos presentes no texto. Essa abordagem pode ser implementada por meio da eliminação sistemática de verbos, adjetivos, advérbios, conectivos, artigos e pronomes.

Dado que é comum que dois ou três substantivos se combinem em um único componente, faz sentido agrupar substantivos que aparecem próximos no texto em um único componente de indexação. Assim, em vez de simplesmente utilizar substantivos isolados como termos de índice, optamos por adotar grupos de substantivos.

2.3.2.5 Tesauro

Um tesauro, também denominado vocabulário controlado, condensa uma lista previamente definida de conceitos cruciais e das palavras que elucidam cada conceito dentro de um domínio específico do conhecimento. Para cada conceito na lista, é compilado um conjunto de sinônimos e palavras relacionadas. Dessa forma, durante uma consulta, um sinônimo pode ser convertido para o seu conceito correspondente. Esse passo de pré-processamento desempenha um papel essencial ao proporcionar

um vocabulário padronizado para indexação e busca. (BAEZA-YATES; RIBEIRO-NETO, 2011)

A utilização de um tesauro, também referido como uma coleção de sinônimos, gera um impacto significativo na revocação em sistemas de recuperação da informação (ELMASRI; NAVATHE, 2016). Esse processo pode ser desafiador, uma vez que uma mesma palavra pode assumir significados distintos em contextos diferentes, como "ponto", que pode referir-se ao placar de uma partida de vôlei, a uma parada de ônibus ou ainda estar relacionado à sutura de um corte (FONSECA, 2020).

2.3.3 Frequência do termo (*TF*)

Embora a representação pela presença ou ausência do termo no documento seja amplamente utilizada, como demonstrado anteriormente, para coleções extensas, o custo computacional associado a essa abordagem pode ser considerável. Nessa questão, a frequência do termo oferece uma visão mais refinada e informativa dos documentos. (MANNING; NAYAK, Acesso em 30/01/2024)

A representação pela frequência do termo leva em conta não apenas a presença ou ausência de termos, mas também a frequência com que eles ocorrem. Isso oferece uma representação mais rica do conteúdo do documento, permitindo distinguir entre documentos que compartilham alguns termos, mas com diferentes ênfases ou intensidades. (SAVOY; GAUSSIER, 2010)

De acordo com Baeza-Yates e Ribeiro-Neto (2013) a frequência do termo no documento pode ser representada de forma matricial. Como ilustrado na Figura 2.5, cada elemento $f_{i,j}$ representa a frequência do termo k_i no documento d_j. Essa frequência do termo é comumente ponderada para levar em consideração a importância relativa dos termos em um documento, um tópico que exploraremos a seguir.

2.3.4 Ponderação de termos

A avaliação da importância de um termo na sumarização de um documento não é uma tarefa simples, uma vez que os termos de indexação possuem distintos níveis

$$\begin{array}{c} \\ k_1 \\ k_2 \\ k_3 \end{array} \begin{array}{c} \begin{array}{cc} d_1 & d_2 \end{array} \\ \begin{bmatrix} f_{1,1} & f_{1,2} \\ f_{2,1} & f_{2,2} \\ f_{3,1} & f_{3,2} \end{bmatrix} \end{array}$$

Figura 2.5: Matriz de frequência do termo k_i no documento d_j.

Fonte: Baeza-Yates e Ribeiro-Neto (2013).

de relevância na descrição do conteúdo documental. Nesse sentido, a presença de um termo 10 vezes em um documento não implica necessariamente que ele seja 10 vezes mais relevante do que um documento que o contenha apenas uma vez. Portanto, a relevância não cresce de maneira proporcional à frequência do termo. (BAEZA-YATES; RIBEIRO-NETO, 2013)

Para aprimorar a importância de um termo, é realizada a atribuição de pesos numéricos a cada termo de indexação presente no documento. Um peso $w_{i.j}$, $w_{i.j} > 0$ é associado a cada termo de indexação k_i de um documento d_j na coleção. Caso um termo k_i não apareça no documento, $w_{i.j} = 0$. Essa prática possibilita uma representação mais equilibrada e refinada da importância de cada termo na tarefa de sumarização, destacando a complexidade inerente à decisão sobre a relevância dos termos em um contexto específico. (BAEZA-YATES; RIBEIRO-NETO, 1999)

A primeira abordagem para ponderação da frequência dos termos, do inglês Term Frequency (TF), foi proposta por Luhn, baseando-se na seguinte premissa: o valor do peso atribuído a um termo em um documento $w_{i.j}$ é proporcional ao $f_{i.j}$. Em outras palavras, quanto mais frequente um termo aparece no texto de um documento, maior é seu peso e, por conseguinte, sua importância ao representar esse documento. Isto leva à Equação 2.1 apresentada abaixo: (BAEZA-YATES; RIBEIRO-NETO, 2011)

$$\mathbf{tf}_{i,j} = \begin{cases} 1 + \log \mathbf{f}_{i,j}, & \text{se } \mathbf{f}_{i,j} > 0 \\ 0, & \text{caso contrário ,} \end{cases} \tag{2.1}$$

onde $tf_{i,j}$ é frequência do termo k_i para o documento d_j. O log $f_{i,j}$ é o logaritmo na base 2 da frequência do termo k_i para o documento d_j. O log é um método de

normalização para o uso junto ao IDF.

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 IDF, os quais são expressados também por função logarítmica e serão abordados a seguir. A Figura 2.6 abaixo mostra a tabela comparando os valores da frequência de termo com os do TF logarítmicos tendo como base a coleção de documentos da Figura 2.1.

Palavra	$f_{i,1}$	$f_{i,2}$	$f_{i,3}$	$f_{i,4}$	$TF_{i,1}$	$TF_{i,2}$	$TF_{i,3}$	$TF_{i,4}$
to	4	2	-	-	3	2	-	-
do	2	-	3	3	2	-	2.585	2.585
is	2	-	-	-	2	-	-	-
be	2	2	2	2	2	2	2	2
or	-	1	-	-	-	1	-	-
not	-	1	-	-	-	1	-	-
is	-	2	2	-	-	2	2	-
am	-	2	1	-	-	2	1	-
what	-	1	-	-	-	1	-	-
think	-	-	1	-	-	-	1	-
therefore	-	-	1	-	-	-	1	-
da	-	-	-	3	-	-	-	2.585
let	-	-	-	2	-	-	-	2
it	-	-	-	2	-	-	-	2

Figura 2.6: Frequências dos termos $f_{i,j}$ e log dos pesos $TF_{i,j}$.

Fonte: Adaptada de Baeza-Yates e Ribeiro-Neto (2011).

2.3.5 Frequência Inversa de Documentos

A aplicação da frequência dos termos como representação de documentos e consultas é eficaz na realização de um dos propósitos da indexação: a recuperação. No entanto, a métrica de frequência de termos tem suas limitações ao não abordar adequadamente a relevância de palavras raras em documentos específicos dentro de uma coleção. Esses termos menos frequentes desempenham um papel crucial ao diferenciar documentos nos quais estão presentes daqueles que não estão. (TEAM, Acesso em 30/01/2024)

Diante dessa questão, Sparck Jones desenvolveu uma interpretação estatística da especificidade dos termos, conhecida como IDF, que se tornou essencial para a

ponderação de termos. Essa interpretação possui uma base heurística que inspirou inúmeras pesquisas em busca de abordagens que ofereçam um embasamento teórico para o IDF. Sua fundamentação está ancorada nas concepções de exaustividade e especificidade dos termos na linguagem. (BAEZA-YATES; RIBEIRO-NETO, 2013)

A exaustividade refere-se às descrições dos documentos, enquanto a especificidade é uma propriedade dos termos de indexação. A abrangência da descrição de um documento é entendida como a extensão que ela oferece aos principais tópicos do documento. Por outro lado, a especificidade de um termo de indexação é interpretada como a medida de quão bem esse termo descreve o tópico de um documento. (BAEZA-YATES; RIBEIRO-NETO, 2013)

Partindo desses conceitos e considerando N como o número total de documentos da coleção e n_i como o número de documentos nos quais o termo i ocorre, a equação pra a ponderação IDF pode ser dada da seguinte forma:

$$\text{IDF}i = \log \frac{N}{ni}. \tag{2.2}$$

Para ilustrar, a Tabela 2.2 apresenta os valores de IDF para cada um dos termos presentes nos documentos da Figura 2.1.

Palavra	n_i	$IDF_i = \log \frac{N}{n_i}$
to	2	1
do	3	0.415
is	1	2
be	4	0
or	1	2
not	1	2
is	2	1
am	2	1
what	1	2
think	1	2
therefore	1	2
da	1	2
let	1	2
it	1	2

Tabela 2.2: Valores de IDF para os termos de indexação dos documentos da coleção da Figura 2.1.

Fonte: Adaptada de Baeza-Yates e Ribeiro-Neto (2011).

2.3.6 Frequência do termo-Frequência Inversa de Documentos

A métrica TF-IDF surge como uma solução ao combinar as métricas IDF e IDF. Ela busca resolver as limitações individuais dessas métricas, atribuindo pesos a cada termo com base em sua frequência em um documento específico TF e na raridade no conjunto de documentos IDF. Assim, o TF-IDF consegue destacar termos que são frequentes em um documento específico, mas raros em todo o conjunto, enfatizando a importância relativa desses termos na representação e classificação de documentos (GUDIVADA; RAO; GUDIVADA, 2018). Sua equação é apresentada na Equação 2.3.

$$Wi,j = \begin{cases} (1 + \log(fi,j)) \times \log \frac{N}{ni} & \text{se } fi,j > 0 \\ 0 & \text{caso contrário} \end{cases} \tag{2.3}$$

A partir da Equação 2.3, podemos deduzir que $w_{i,j}$ representa o peso numérico atribuído a um dado termo k_i em um documento d_j pelo método tf-idf. O termo $(1 + \log f_{i,j})$ corresponde ao cálculo normalizado da frequência do termo tf (Equação 2.1). Por sua vez, $\log \frac{N}{ni}$ representa o cálculo normalizado da frequência do inverso da frequência do documento idf, representada na Equação 2.2.

A Tabela 2.3 exibe os valores de TF-IDF para a coleção de exemplo destacada na Figura 2.1. Apesar de sua simplicidade, esse exemplo evidencia as propriedades do esquema de ponderação TF-IDF. Termos menos comuns, como “casa” e “for”, possuem pesos mais elevados, uma vez que são mais seletivos. Além disso, os termos mais frequentes dentro de um documento apresentam frequências relativas mais altas. (BAEZA-YATES; RIBEIRO-NETO, 2013)

#	termo	d_1	d_2	d_3	d_4
1	to	3	2	-	-
2	do	0.830	-	1.073	1.073
3	is	4	-	-	-
4	be	-	-	-	-
5	or	-	2	-	-
6	not	-	2	-	-
7	I	-	2	2	-
8	am	-	2	1	-
9	what	-	2	-	-
10	think	-	-	2	-
11	therefore	-	-	2	-
12	da	-	-	-	5.170
13	let	-	-	-	4
14	it	-	-	-	4
Tamanho do documento		5.068	4.899	3.762	7.738

Tabela 2.3: Valores de IDF para os termos de indexação dos documentos da coleção Figura 2.1.

Fonte: Adaptada de Baeza-Yates e Ribeiro-Neto (2011).

2.3.7 Modelos de Recuperação

De acordo com Hiemstra (2009) há duas razões significativas para adotar modelos de recuperação de informação. A primeira é que esses modelos direcionam a pesquisa e oferecem a base para a discussão acadêmica. A segunda razão é que esses modelos podem funcionar como um guia para implementar efetivamente um sistema real de recuperação de informações.

Em diversas disciplinas científicas, modelos matemáticos são empregados com o propósito de compreender e analisar comportamentos ou fenômenos do mundo real. Um modelo de recuperação de informação tem a finalidade de antecipar e explicar

o que um usuário considerará relevante com base em sua consulta. A precisão das previsões do modelo pode ser verificada por meio de experimentos controlados. Para formular previsões e obter uma compreensão mais aprofundada da recuperação de informação, é essencial que os modelos estejam solidamente embasados em intuições, metáforas e em algum ramo da matemática. (HIEMSTRA, 2009)

Segundo Baeza-Yates e Ribeiro-Neto (2013) um modelo de Recuperação da Informação é definido pela quádrupla:

$$[D, Q, F, R(q_i, d_j)] \ . \tag{2.4}$$

A partir do modelo descrito na Equação 2.4, podemos observar que D é um conjunto que representa as visões lógicas dos documentos, enquanto Q é um conjunto que representa as visões lógicas das consultas dos usuários. F denota o *framework* (arcabouço) de modelagem para documentos e consultas. $R(q_i, d_j)$ representa a função de classificação (ranking) ou ordenação.

Considerando as representações da consulta e dos documentos, como q_i e d_j, a função de ranqueamento $\mathrm{R}(q_i, d - J)$ atribui um grau de similaridade ao documento d_j em relação à consulta q_i. Essa operação é demonstrada na Figura 2.7.

Conforme destacado por Baeza-Yates e Ribeiro-Neto (2013), os modelos de Recuperação de Informação são classificados em três categorias principais: baseados em texto, baseados em links e baseados em objetos multimídia. Dentro da categoria baseada em texto, há uma subdivisão entre modelos para texto não estruturado e modelos estruturados. No contexto do texto não estruturado, os três modelos clássicos são conhecidos como Booleano, Vetorial e Probabilístico. Este trabalho se concentrará exclusivamente na exploração desses três modelos, buscando compreender suas características e aplicabilidades.

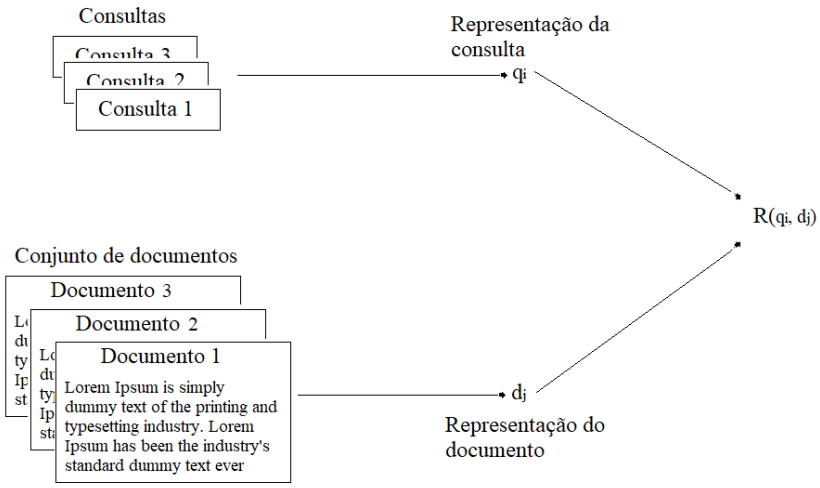


Figura 2.7: A função de ranqueamento $\mathrm{R}(q_i, d_j)$ recebe como entrada as representações da consulta e do documento e atribui um grau de similaridade ao documento d_j em relação à consulta q_i.

Fonte: Adaptada Baeza-Yates e Ribeiro-Neto (2013).

O 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. (BAEZA-YATES; RIBEIRO-NETO, 2011)

2.3.7.1 Modelo Booleano

Considerado como o primeiro modelo de Recuperação da Informação, o modelo Booleano fundamenta-se na teoria dos conjuntos e na álgebra booleana. Esse modelo considera que os termos de indexação estão presentes ou ausentes nos documentos, resultando em frequências binárias na matriz de termos por documentos. Uma consulta q é composta por termos de indexação conectados por três operadores booleanos: *or*, *and* e *not*. (HIEMSTRA, 2009)

Baeza-Yates e Ribeiro-Neto (2011) propõem a seguinte definição para o modelo Booleano: Os elementos na matriz de termos por documentos são representados por 1 para indicar a presença do termo no documento ou 0 para indicar a ausência do termo.

Uma consulta q é uma expressão booleana que envolve os termos de indexação, como, por exemplo: $[q = k_a \wedge (k_b \neg k_c)]$. Dada uma consulta, um componente conjuntivo de termos que satisfaz suas condições é denominado componente conjuntivo *c(q)* de consulta. Ao compilar todos os componentes conjuntivos da consulta, é possível reescrevê-la como uma disjunção desses componentes. Isso é conhecido como a forma normal disjuntiva da consulta, referenciada como q_{DNF}.

Para ilustrar, GÖker e Davies (2009) propõem a realização de consultas específicas. A primeira delas é "social AND economic", que resulta em um conjunto de documentos nos quais os termos "social" e "economic" estão indexados no mesmo documento, representando a interseção entre os dois conjuntos, como demonstrado na Figura 2.8.a. A segunda consulta é "social OR political", cujo resultado é ilustrado na Figura 2.8.b. Além disso, temos a consulta "(social OR political) NOT (social AND economic)", cujo resultado da busca é destacado em cinza na Figura 2.8.c.

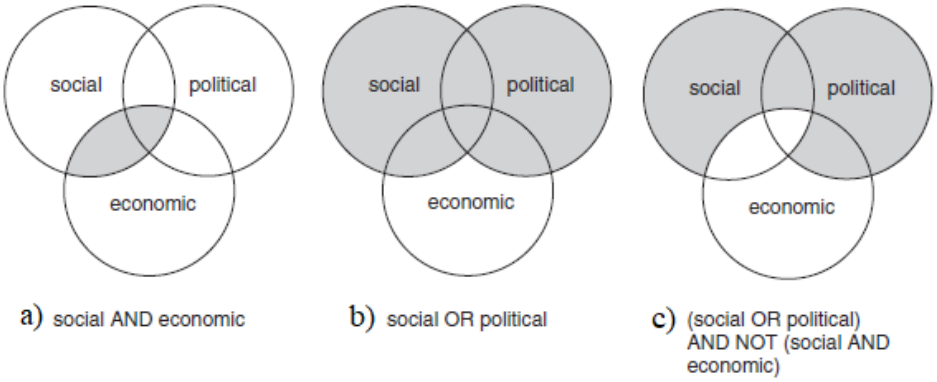


Figura 2.8: Combinações booleanas de conjuntos visualizadas como diagramas de Venn[2]. O resultado da combinação booleana descrita nos itens a, b e c está destacado em cinza em cada figura respectivamente.

Fonte: Adaptada de GÖker e Davies (2009).

O modelo booleano não realiza a classificação de documentos. Um documento é considerado relevante ou não, sem atribuir um grau específico de relevância. Além disso, não leva em conta a frequência de ocorrência dos termos nos documentos. A simplicidade de implementação é uma de suas características. Esse modelo mostra-se eficaz na recuperação de documentos relevantes, principalmente quando o usuário final possui familiaridade com o vocabulário do domínio e pode formular consultas

booleanas relativamente complexas. (GUDIVADA; RAO; GUDIVADA, 2018)

As principais vantagens do modelo Booleano incluem seu formalismo claro e sua simplicidade, que adota pesos binários para os termos de indexação. No entanto, como desvantagens, destaca-se a ausência de ranqueamento, o que pode resultar na recuperação de muitos documentos que são, na verdade, irrelevantes. Além disso, a formulação de consultas booleanas é inconveniente para a maioria dos usuários. Atualmente, é conhecido que a ponderação dos termos de indexação pode proporcionar uma melhoria significativa na qualidade da recuperação. (BAEZA-YATES; RIBEIRO-NETO, 2013)

2.3.7.2 Modelo Vetorial

O modelo de espaço vetorial é fundamentado na ideia de que, em um sentido aproximado, o significado de um documento e de uma consulta é transmitido pelas palavras que os compõem. Ao representar as palavras do documento e de uma consulta por um vetor, torna-se possível comparar documentos com consultas para determinar quão semelhante é o conteúdo deles. Se uma consulta for considerada semelhante a um documento, é possível calcular um Similarity Coefficient (SC), coeficiente de similaridade em português, que mede a semelhança entre o documento e a consulta. Documentos cujo conteúdo, medido pelos termos do documento, corresponde mais de perto ao conteúdo da consulta, são considerados os mais relevantes. (GROSSMAN; FRIEDER, 2004)

Este modelo emprega um vetor para representar os termos dos documentos e outro vetor para representar os termos da consulta. A determinação da proximidade entre qualquer vetor de documento e o vetor de consulta requer a escolha de um método. Diversos métodos podem ser utilizados, tais como Similaridade do Cosseno, Similaridade de Jaccard, Distância Euclidiana, entre outros. O método que abordaremos para calcular a similaridade será o método do Cosseno. (CERI et al., 2013)

[2]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.Educação (Acesso em 30/01/2024)

Baeza-Yates e Ribeiro-Neto (1999) estabelecem formalmente que, neste modelo, o peso $w_{i,j}$ atribuído ao par (k_i, d_j), onde k_i é o termo e k_j é o documento, é positivo e não binário. Os termos em uma consulta também são associados a um peso. Dessa forma, o vetor de uma consulta é definido no espaço n-dimensional como $\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$, 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 $\vec{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$. Observa-se que cada termo da expressão de consulta se torna um eixo do referido espaço n-dimensional.

Dessa forma, um documento d_j e uma consulta de usuário q são representados como vetores com t dimensões. O modelo vetorial, então, calcula o grau de similaridade do documento d_j em relação à consulta q como a correlação entre vetores $\vec{d_j}$ e $\vec{q}$. Essa correlação pode ser quantificada pelo cosseno do ângulo entre esses dois vetores, conforme mostrado na Equação 2.5 abaixo:

$$\text{sim}(d_j, q) = \frac{\vec{d_j} \cdot \vec{q}}{\|\vec{d_j}\| \times \|\vec{q}\|} = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}. \tag{2.5}$$

A partir dessa Equação 2.5, observa-se que $\|\vec{d_j}\|$ e $\|\vec{q}\|$ são as normas dos vetores do documento e da consulta e $\vec{d_j} \cdot \vec{q}$ é o produto interno dos dois vetores. O fator $\|\vec{q}\|$ não afeta o ranqueamento uma vez que é o mesmo para todos os documentos. Já o fator $\|\vec{d_j}\|$ faz a normalização pelo tamanho do documento. Os pesos adotados no modelo são aqueles obtidos pela métrica TF-IDF apresentado anteriormente. Com isso, tem-se:

$$W_{i,q} = (1 + \log(f_{i,q})) \times \log \frac{N}{n_i}, \tag{2.6}$$

$$W_{i,j} = (1 + \log(f_{i,j})) \times \log \frac{N}{n_i}. \tag{2.7}$$

A partir das equações acima temos que $f_{i,q}$ é a frequência do termo k_i no texto da consulta q. Como $w_{i,j} \geq 0$ e $w_{i,q} \geq 0$, $sim(d_j, q)$ varia entre 0 e 1. Diferentemente de seguir um critério de decisão binário, o modelo vetorial classifica os documentos com base no grau de similaridade em relação a uma consulta específica. Assim, um documento pode ser considerado relevante mesmo que atenda parcialmente à consulta.

Essa abordagem permite a definição de um limiar para o grau de similaridade, $sim(d_j, q)$, e a recuperação apenas de documentos com uma similaridade acima desse limiar. (BAEZA-YATES; RIBEIRO-NETO, 2013)

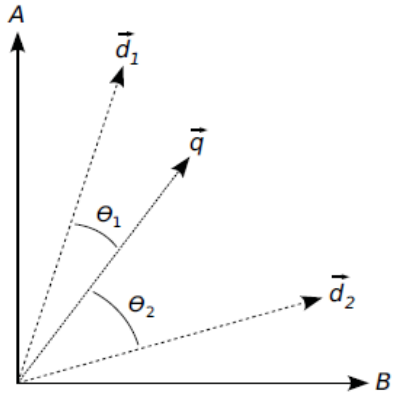


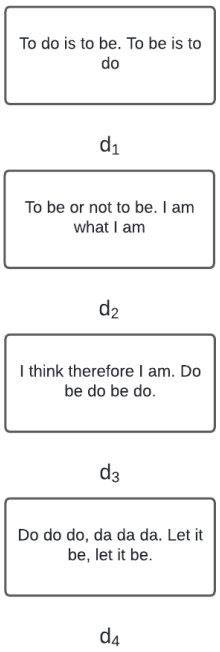
Figura 2.9: Medida de similaridade por cosseno no modelo Vetorial

Fonte: BÜttcher, Clarke e Cormack (2010).

O exemplo apresentado na Figura 2.9 acima demonstra o cálculo dos ângulos entre o vetor de consulta $\vec{q}$ e os vetores dos dois documentos, d_1 e d_2. Devido a $\theta_1 \leq \theta_2$, d_1 será posicionado em um ranking superior em relação a d_2. Vale ressaltar que neste exemplo, a expressão de busca é composta por dois termos, configurando, assim, um espaço bidimensional. (BÜTTCHER; CLARKE; CORMACK, 2010)

A Figura 2.10 exemplifica o processo de classificação para a coleção apresentada na Figura 2.1 em resposta à consulta "to do". Observa-se que o documento d_1 é o mais bem ranqueado, uma vez que contém todos os termos da consulta. Os documentos d_3 e d_4 possuem apenas o termo "do" da consulta, porém o documento d_4 recebe uma pontuação menor devido à maior norma do seu vetor. (BAEZA-YATES; RIBEIRO-NETO, 2011)

O modelo vetorial apresenta algumas vantagens: Aprimora a qualidade da recuperação por meio de seu esquema de ponderação de termos; Permite a recuperação de documentos com correspondência parcial às condições da consulta; A fórmula do cosseno organiza os documentos com base no grau de similaridade com a consulta; A normalização pelo tamanho do documento está intrinsecamente incorporada ao modelo. Entretanto, sua desvantagem reside na consideração dos termos de indexação



doc	rank computation	rank
d_1	$\frac{1 \times 3 + 0.415 \times 0.830}{5.068}$	0.66
d_2	$\frac{1 \times 2 + 0.415 \times 0}{4.899}$	0.408
d_3	$\frac{1 \times 0 + 0.415 \times 1.073}{3.762}$	0.118
d_4	$\frac{1 \times 0 + 0.415 \times 1.073}{7.738}$	0.058

Figura 2.10: Ranking dos documentos para a consulta "to do" utilizando pesos TF-IDF dados pelas Equações 2.6 e 2.7.

Fonte: Adaptada de Baeza-Yates e Ribeiro-Neto (2011).

como mutuamente independentes. (BAEZA-YATES; RIBEIRO-NETO, 2011)

2.3.7.3 Modelo Probabilístico

O modelo probabilístico tem suas bases na teoria da probabilidade e estatística, fornecendo uma estrutura matemática robusta para lidar com a incerteza inerente ao processo de recuperação de informações. Ao contrário do modelo de espaço vetorial, que representa documentos e consultas como vetores, esse modelo pressupõe que os documentos são gerados a partir de algum modelo probabilístico, e as consultas do usuário são tratadas como amostras dessa distribuição. (GÖKER; DAVIES, 2009)

No âmbito desse modelo, cada documento é classificado como relevante ou irrelevante para uma consulta específica. No entanto, é importante observar que a relevância de um documento em relação a uma consulta não fornece informações sobre a relevância de outros documentos em situações semelhantes. Essa abordagem destaca a individualidade de cada documento em relação a uma consulta, refletindo

a visão probabilística de que a relevância é uma característica intrínseca de cada par documento-consulta. (SAVOY; GAUSSIER, 2010)

Baeza-Yates e Ribeiro-Neto (2013) formalizam a seguinte definição: Considere uma consulta q como um subconjunto dos termos de indexação. Um documento d_j é representado por um vetor de pesos binários indicando a presença ou ausência de termos de indexação. Onde $w_{i,j} = 1$ se o termo k_i ocorre no documento d_i e 0 caso contrário. A Equação 2.8 seguir formaliza este conceito:

$$\vec{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{t,j}). \tag{2.8}$$

Seja R um conjunto de documentos estimados como relevantes para uma consulta q, e seja $\overline{R}$ o conjunto de documentos estimados como não relevantes, complemento de R. $P(R \mid \vec{d_{j,q}})$ é a probabilidade de que o documento d_j com a representação $\vec{d_j}$ seja relevante para a consulta q. Com isso, tem-se que $P(\overline{R} \mid \vec{d_{j,q}})$ é a probabilidade de que o documento d_j não seja relevante para a consulta q. Dessa forma, a similaridade $sim(d_{j,q})$ entre o documento d_j e a consulta q é dada por:

$$sim(d_{j,q}) = \frac{P(R \mid \vec{d_{j,q}})}{P(\overline{R} \mid \vec{d_{j,q}})}. \tag{2.9}$$

Por meio da regra de Bayes tem-se:

$$sim(d_{j,q}) = \frac{P(\vec{d_j} \mid R, q) \times P(R, q)}{P(\vec{d_j} \mid \overline{R}, q) \times P(\overline{R}, q)} = \frac{P(\vec{d_j} \mid R, q) \times P(R \mid q)}{P(\vec{d_j} \mid \overline{R}, q) \times P(\overline{R} \mid q)}, \tag{2.10}$$

onde:

- $P(\vec{d_j} \mid R, q)$: é a probabilidade de que um documento seja selecionado aleatoriamente do conjunto R dos documentos relevantes para a consulta;
- $P(\vec{d_j} \mid \overline{R}, q)$: é a probabilidade de que um documento seja selecionado aleatoriamente do conjunto R dos documentos não relevantes para a consulta;
- $P(R \mid q)$: é a probabilidade de que um documento selecionado aleatoriamente

de toda coleção seja relevante para a consulta;

- $P(\overline{R} \mid q)$: é a probabilidade de que um documento selecionado aleatoriamente de toda coleção seja não relevante para a consulta.

Como $P(R \mid q)$ e $P(\overline{R} \mid q)$ são os mesmos para todos os documentos da coleção, tem-se:

$$sim(d_{j,q}) \approx \frac{P(\vec{d_j} \mid R, q)}{P(\vec{d_j} \mid \overline{R}, q)}. \tag{2.11}$$

Considerando que a representação do documento d_j é constituída por valores binários que indicam a presença ou ausência dos termos no documento, e assumindo a independência entre os termos de indexação, obtém-se:

$$sim(d_j, q) \approx \frac{\prod_{k_i|w_{i,j}=1} P(k_i|R, q) \times \prod_{k_i|w_{i,j}=0} P(\overline{k_i}|R, q)}{\prod_{k_i|w_{i,j}=1} P(k_i|\overline{R}, q) \times \prod_{k_i|w_{i,j}=0} P(\overline{k_i}|\overline{R}, q)}, \tag{2.12}$$

onde:

- $P(k_i|R, q)$: é a probabilidade de que o termo de indexação k_i esteja presente em um documento selecionado aleatoriamente do conjunto R de documentos relevantes;

- $P(\overline{k_i}|R, q)$: é a probabilidade de que o termo de indexação k_i não esteja presente em um documento selecionado aleatoriamente do conjunto R de documentos relevantes.

Por meio de manipulações envolvendo logaritmos e produtórios da Equação 2.12, obtemos a formula para o cálculo do ranking no contexto do modelo probabilístico:

$$sim(d_j, q) \approx \sum_{k_i \in q \wedge k_i \in d_j} \log\left(\frac{p_{iR}}{1 - p_{iR}}\right) + \log\left(\frac{1 - q_{iR}}{q_{iR}}\right). \tag{2.13}$$

Dado que o conjunto R não é conhecido no início desse processo, é importante estabelecer uma forma para calcular as probabilidades p_{iR} e q_{iR} iniciais. Uma abordagem seria utilizar a tabela de contingência das incidências de termos representada pela Tabela 2.4.

Caso	Relevantes	Não relevantes	Total
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

Tabela 2.4: Tabela de contingência das incidências de termos.

Fonte: Adaptada de Baeza-Yates e Ribeiro-Neto (2011).

Supondo que a tabela acima esteja disponível para consulta, poderíamos expressar isso na Equação 2.14 abaixo:

$$p_{iR} = \frac{r_i}{R} \ , \ q_{iR} = \frac{n_i - r_i}{N - R}. \tag{2.14}$$

Substituindo a Equação 2.14 acima na Equação 2.13, tem-se como resultado a seguinte Equação 2.15:

$$sim(d_j, q) \sim \sum_{k_i[q,d_j]} \log \left(\frac{r_i(N - n_i - R + r_i)}{(R - r_i)(n_i - r_i)} \right). \tag{2.15}$$

A Equação 2.16 é aplicável para calcular o ranking no modelo probabilístico quando a informação sobre a relevância dos documentos não está disponível. Nessa formulação, para contornar valores pequenos de r_i, adicionamos 0.5 a cada termo da equação. No contexto, apenas o componente IDF está presente, considerando que os documentos foram tratados como tendo valores binários. Além disso, optou-se por $R = r_i = 0$ como uma inicialização padrão, pois a equação não pode ser computada sem estimativas de r_i e R. A partir dessas considerações e por meio de manipulações envolvendo logaritmos e produtórios, chegamos à seguinte expressão:

$$sim(d_j, q) \sim \sum_{k_i \in q \wedge k_i \in d_j} \log \left(\frac{N - n_i + 0.5}{n_i + 0.5} \right). \tag{2.16}$$

A Tabela 2.5 abaixo demonstra o cálculo do ranking para a consulta “to do” tendo a coleção da Figura 2.7 como exemplo.

Doc	Computação do escore	Escore
d_1	$\log(\frac{4-2+0.5}{2+0.5}) + \log(\frac{4-3+0.5}{3+0.5})$	-1.222
d_2	$\log(\frac{4-2+0.5}{2+0.5})$	0
d_3	$\log(\frac{4-3+0.5}{3+0.5})$	-1.222
d_4	$\log(\frac{4-3+0.5}{3+0.5})$	-1.222

Tabela 2.5: Escores computados pela equação Equação 2.16 para a consulta "to do" da coleção da Figura 2.1.

Fonte: Adaptada de Baeza-Yates e Ribeiro-Neto (2011).

Quando o peso gerado por um termo é negativo, a Equação 2.16 não terá um comportamento adequado. Esse comportamento é percebido quando $n_i > N/2$, com isso , termos negativos aparecem nos cálculos. Uma alternativa para evitar esse comportamento é retirar o fator n_i do numerador da Equação 2.16, resultando na seguinte equação:

$$sim(d_j, q) \sim \sum_{k_i \in q \wedge k_i \in d_j} \log\left(\frac{N + 0.5}{n_i + 0.5}\right). \tag{2.17}$$

A Tabela 2.6 apresenta os novos escores obtidos para a consulta "to do" na mesma coleção da Figura 2.5, porém utilizando a Equação 2.17 sem o fator n_i no numerador.

Doc	Computação do escore	Escore
d_1	$\log(\frac{4+0.5}{2+0.5}) + \log(\frac{4+0.5}{3+0.5})$	1.210
d_2	$\log(\frac{4+0.5}{2+0.5})$	0.847
d_3	$\log(\frac{4+0.5}{3+0.5})$	0.362
d_4	$\log(\frac{4+0.5}{3+0.5})$	0.362

Tabela 2.6: Escores computados pela Equação 2.17 modificada para a consulta "to do".

Fonte: Adaptada de Baeza-Yates e Ribeiro-Neto (2011).

2.4 Avaliação da Recuperação

Para avaliar a eficácia de um sistema de recuperação de informação, é crucial mensurar o quão bem o sistema atende às necessidades de informação dos usuários. Essa tarefa apresenta desafios, uma vez que as opiniões dos usuários podem variar quanto à adequação de um documento para atender a uma mesma necessidade de informação. Apesar dessa complexidade, é viável estabelecer métricas que permitam avaliar a qualidade dos resultados da recuperação, correlacionando-os com as preferências de uma amostra representativa de usuários.

2.4.1 Precisão e Revocação

As métricas de precisão e revocação fundamentam-se na classificação binária de relevância, que indica se um documento é relevante ou não para a pesquisa. A revocação, também conhecida como cobertura, recall em inglês, representa o número de documentos relevantes recuperados por uma consulta, dividido pelo número total de documentos relevantes presentes na base de dados. Por outro lado, a precisão, ou precision em inglês, é definida como o número de documentos relevantes recuperados pela busca, dividido pelo número total de documentos recuperados. (BAEZA-YATES; RIBEIRO-NETO, 2011)

$$Precisão = p = \frac{|R \cap A|}{|A|} \tag{2.18}$$

$$Revocação = r = \frac{|R \cap A|}{|R|} \tag{2.19}$$

A Figura 2.11 demonstra uma ilustração dos conjuntos Precisão e Revocação. Dentro do conjunto de todos os documentos, tem-se o conjunto R dos documentos relevantes e o conjunto A dos documentos recuperados. A área pintada de cinza representa a interseção desses dois conjuntos, respectivamente.

No caso da Figura 2.12, evidência o antagonismo entre a precisão e revocação. Apesar de serem bastante utilizadas, possuem algumas limitações. Obter a revocação máxima em uma busca específica demanda conhecimento de todos os documentos

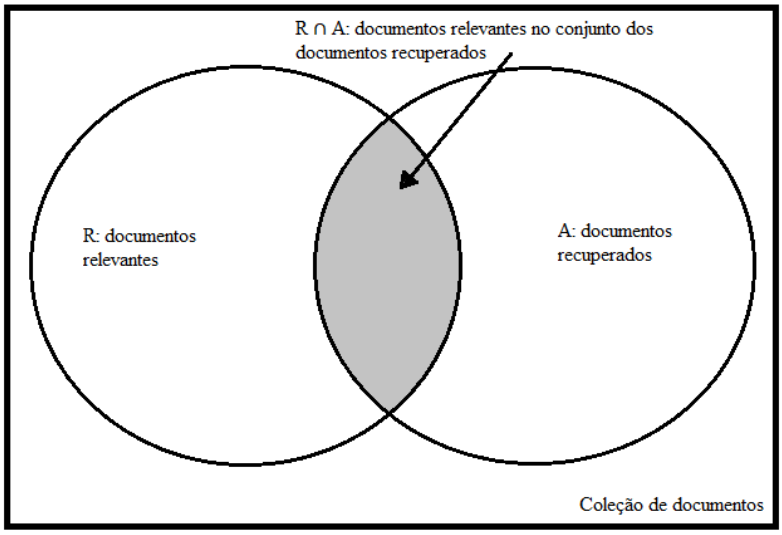


Figura 2.11: Ilustração na forma de conjunto dos itens da Equação 2.18 e da Equação 2.19.

Fonte: Adaptada de Baeza-Yates e Ribeiro-Neto (2013).

presentes no corpus. Contudo, essa tarefa torna-se inviável em bases documentais muito extensas. Além disso, essas métricas avaliam diferentes aspectos do *corpus*. (FERNEDA, Acesso em 30/01/2024)

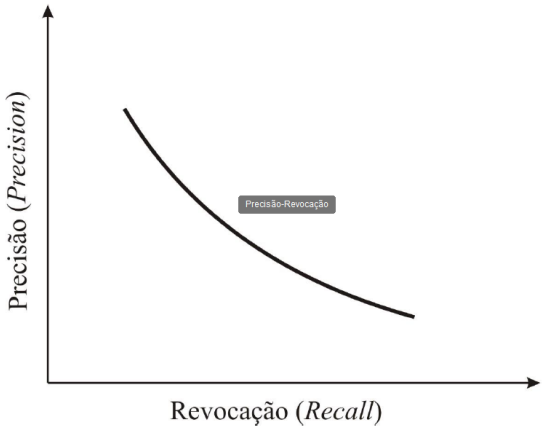


Figura 2.12: Antagonismo entre precisão e revocação. Quanto mais alto o valor da precisão, mas baixo tem-se o valor da revocação e vice-versa.

Fonte: Adaptada de Baeza-Yates e Ribeiro-Neto (2013).

.

2.4.2 Métrica F

A métrica F, do inglês *F-measure*, também conhecida como pontuação F ou medida F, é uma métrica de desempenho amplamente utilizada na avaliação de sistemas de recuperação de informação. Ela combina precisão e revocação em uma única medida, proporcionando uma avaliação mais equilibrada do desempenho do sistema. (CROFT; METZLER; STROHMAN, 2015)

Segundo Croft, Metzler e Strohman (2015), a métrica F busca o equilíbrio entre precisão e revocação. Seu cálculo é realizado por meio da média harmônica entre precisão e revocação, sendo expresso na Equação 2.20:

$$F = \frac{2 \times (\text{revocação} \times \text{precisão})}{\text{revocação} + \text{precisão}}. \tag{2.20}$$

A utilização da média harmônica destaca a relevância de valores pequenos, ao passo que a média aritmética é mais suscetível à influência de valores excepcionalmente grandes. Suponha um resultado de pesquisa que recuperasse quase toda a coleção de documentos, resultando em uma recuperação de 1.0 e uma precisão próxima de 0. Enquanto a média aritmética desses valores seria 0.5, a média harmônica se aproximaria de 0. Dessa forma, a média harmônica se revela como um resumo mais preciso da eficácia desse conjunto recuperado. (CROFT; METZLER; STROHMAN, 2015)

Capítulo 3

Sistema de Gestão e Gerenciamento de Documentos Acadêmicos

Este capítulo abordará diversos aspectos cruciais para o entendimento do sistema em desenvolvimento. Inicialmente, será apresentada a motivação, destacando os desafios e necessidades que levaram à sua concepção. Em seguida, serão discutidos os trabalhos relacionados. Além disso, será detalhada a modelagem e arquitetura do sistema, descrevendo as decisões de design e estruturação adotadas para garantir sua eficiência e escalabilidade.

Por fim, serão apresentados os requisitos do sistema, delineando as funcionalidades e características essenciais que o sistema deve possuir para atender às necessidades dos usuários e cumprir seus objetivos. Esses requisitos serviram como base para o desenvolvimento e implementação do sistema, orientando todas as etapas do processo de construção e garantindo a entrega de uma solução completa e satisfatória.

3.1 Motivação

Os sistemas de recuperação da informação têm desempenhado um papel crucial na evolução da gestão de documentos acadêmicos. Desde os primeiros catálogos de bibliotecas até os modernos motores de busca online, essas ferramentas foram

desenvolvidas para ajudar os usuários a encontrar informações relevantes de forma rápida e eficiente. No entanto, apesar dos avanços tecnológicos, ainda existem desafios significativos a serem enfrentados, especialmente no contexto da crescente quantidade de informações disponíveis e na necessidade de filtragem de conteúdo.

Um dos principais desafios na gestão de documentos é a falta de filtragem de conteúdo. Com o aumento exponencial da produção de documentos, torna-se cada vez mais difícil separar informações relevantes de conteúdos irrelevantes ou de baixa qualidade. Isso pode dificultar a busca e recuperação de informações precisas e confiáveis, prejudicando a eficiência e a eficácia da pesquisa acadêmica. (STANLEY, 2021)

Outra lacuna a ser preenchida é a escassez de sistemas dedicados à gestão de documentos acadêmicos. Embora existam várias ferramentas disponíveis para a organização de documentos, poucas foram desenvolvidas especificamente para atender às necessidades únicas dos pesquisadores e acadêmicos. Isso pode resultar em processos de gerenciamento de documentos fragmentados e ineficientes, dificultando a colaboração e a disseminação do conhecimento.

Além disso, um desafio significativo é a prevalência de conteúdos acadêmicos pagos. Muitas vezes, os pesquisadores enfrentam barreiras de acesso a artigos e publicações devido a restrições de pagamento ou assinatura. Isso limita o acesso ao conhecimento e pode criar desigualdades no acesso à informação, prejudicando a colaboração e o avanço da pesquisa. (WILLINSKY, 2006)

É importante ressaltar que nem todos os tipos de artigos podem ser enviados para sistemas de recuperação de informação devido a questões de direitos autorais. O respeito aos direitos autorais é crucial para proteger o trabalho intelectual dos autores e editores. Enviar artigos protegidos por direitos autorais sem a devida autorização pode levar a consequências legais graves, além de prejudicar a reputação da plataforma e dos usuários que a utilizam. (FERNANDES; FERNANDES; GOLDIM, 2008)

Os direitos autorais são uma forma de proteção legal concedida aos autores de obras originais, incluindo artigos acadêmicos, trabalhos de pesquisa e publicações

científicas. Quando um autor cria um artigo, ele automaticamente adquire direitos exclusivos sobre essa obra, que incluem o direito de reproduzi-la, distribuí-la, exibi-la publicamente e criar obras derivadas. Esses direitos são importantes para assegurar que o autor seja reconhecido por seu trabalho e possa controlar como ele é usado. No contexto acadêmico, a proteção dos direitos autorais é essencial para incentivar a produção de novos conhecimentos e garantir que os autores recebam o crédito devido por suas contribuições intelectuais. (FERNANDES; FERNANDES; GOLDIM, 2008)

Existem várias formas de licenciamento que os autores podem escolher ao publicar seus trabalhos. As licenças tradicionais de direitos autorais geralmente concedem à editora todos os direitos exclusivos sobre a obra, limitando a capacidade do autor de redistribuir ou reutilizar seu próprio trabalho. Em contraste, as licenças abertas, como as licenças Creative Commons, permitem que os autores mantenham os direitos autorais enquanto concedem permissões específicas ao público para usar, compartilhar e até modificar a obra, desde que sejam respeitadas as condições estipuladas pelo autor. Por exemplo, uma licença Creative Commons Attribution (CC BY) permite que outros distribuam, remixem, adaptem e construam a partir do trabalho original, desde que atribuam a devida autoria. (COMMONS, Acesso em 15/05/2024)

Diante desses desafios, o desenvolvimento de um sistema de gestão de documentos acadêmicos abrangente e acessível é de grande importância. Um sistema que permita aos usuários enviar e visualizar documentos de forma gratuita, centralize a busca em um único lugar e ofereça mecanismos robustos de filtragem é essencial para facilitar o acesso ao conhecimento e promover a excelência acadêmica. Investir em soluções inovadoras e centradas no usuário é fundamental para construir uma base sólida para o avanço contínuo da educação e da pesquisa.

3.2 Trabalhos Relacionados

Embora existam diversos softwares de gestão de documentos, boa parte deles não é voltada para a gestão acadêmica. Além disso, das opções disponíveis, poucas são disponibilizadas de forma gratuita. Nesse contexto, os seguintes softwares -

DSpace, Koha e VuFind - se destacam por serem voltados especificamente para a gestão acadêmica e oferecerem versões gratuitas. A seguir, serão detalhadas as características e funcionalidades dessas ferramentas.

3.2.1 DSpace

O DSpace[1] é uma aplicação web que possibilita a pesquisadores e acadêmicos a publicação de documentos e dados. Ele busca atender a uma necessidade específica como um sistema de arquivos digitais, focado no armazenamento, acesso e preservação de conteúdo digital a longo prazo. Com código aberto e distribuído sob a licença BSD, o DSpace é gratuito para uso e modificação.

Desenvolvido pela Massachusetts Institute of Technology (MIT) Libraries e pela Hewlett-Packard (HP) Labs em 2002, o DSpace tornou-se uma das ferramentas mais populares para a criação e gestão de repositórios digitais em instituições acadêmicas, culturais e de pesquisa em todo o mundo. Oferece uma estrutura flexível e personalizável para armazenar, gerenciar e preservar diversos tipos de conteúdo digital, como artigos de revistas, teses, dissertações e relatórios de pesquisa.

Algumas universidades brasileiras como a Universidade Federal de Minas Gerais (UFMG), Universidade Federal do Rio de Janeiro (UFRJ) e Universidade Federal Rural do Rio de Janeiro (UFRRJ) utilizam a estrutura do DSpace para gerenciar seus repositórios de arquivos. A Figura 3.1 demonstra a interface para pesquisas no repositório da UFRJ. Essa interface também é utilizada nos repositórios da UFMG e da UFRRJ.

Apesar de o DSpace ser gratuito, ele possui um modelo de filiação em que uma entidade interessada pode contribuir financeiramente e participar das decisões. Esse modelo de filiação envolve diferentes níveis de colaboração, que vão desde o nível "supporter"até o nível "platinum", cada um com diferentes benefícios e custos associados.

[1] <https://dspace.lyrasis.org/>

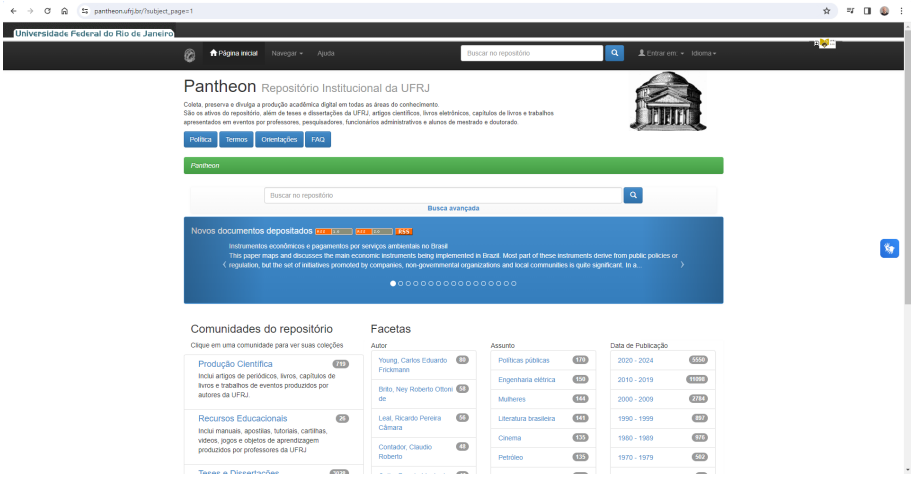


Figura 3.1: Interface para pesquisa de documentos acadêmicos da UFRJ.

Fonte: <https://pantheon.ufrj.br/simple-search>

3.2.2 Koha

O Koha[2] é um software de gestão de bibliotecas de código aberto, distribuído sob a General Public License (GNU), Licença Pública Geral traduzido para o português, e sem custo de licenciamento, amplamente adotado em milhares de bibliotecas ao redor do mundo. Desde seu lançamento em janeiro de 2000, o Koha tem sido uma opção viável para automatizar bibliotecas sem a necessidade de pagar por licenças.

Desenvolvido pela Katipo Communications para a Horowhenua Library Trust, da Nova Zelândia, o nome "Koha" significa "presente" ou "doação" em maori, a língua dos aborígenes neozelandeses. Essa escolha de nome reflete a filosofia de oferecer ao mundo uma ferramenta de automação de bibliotecas de código aberto e sem custos associados.

O sistema suporta uma variedade de padrões bibliotecários, como MARC21, Z39.50, SIP2, SRU, entre outros, facilitando a interoperabilidade com outros sistemas e a troca de dados entre bibliotecas. Além disso, o Koha oferece recursos avançados de catalogação, gerenciamento de empréstimos, reservas de materiais e geração de relatórios, garantindo uma experiência completa de automação para bibliotecas de qualquer porte.

[2] <https://koha-community.org/>

O Koha apresenta duas interfaces web distintas: uma para a equipe da biblioteca e outra para os usuários finais, conhecida como Online Public Access Catalog (OPAC). Essa separação de funcionalidades permite que a equipe interna da biblioteca gerencie eficientemente o sistema, enquanto os usuários podem acessar facilmente o catálogo online, pesquisar e fazer solicitações de empréstimo. A Figura 3.2 abaixo apresenta a interface de pesquisa.

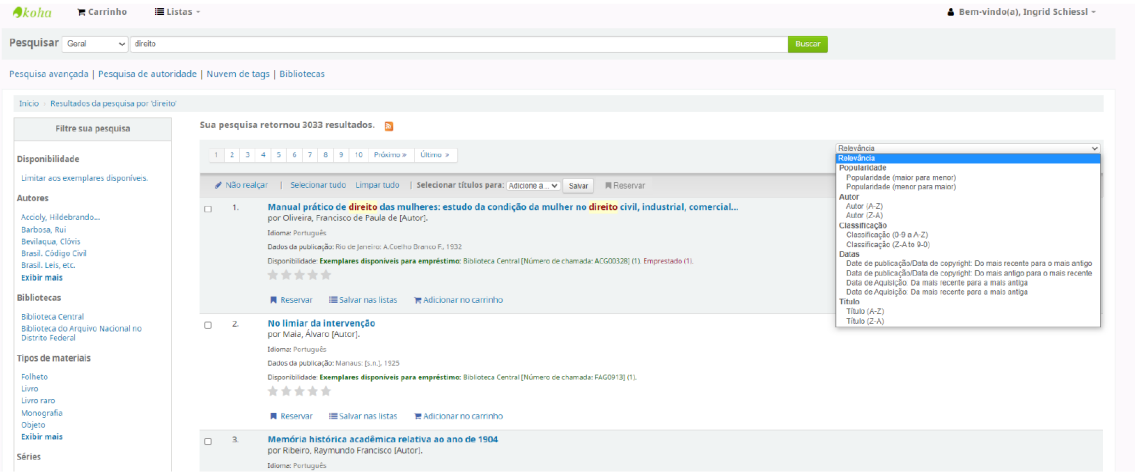


Figura 3.2: Interface para pesquisa de documentos com drop-down.

Fonte: RIDI (Acesso em 27/04/2024)

3.2.3 VuFind

O VuFind [3] é uma plataforma de código aberto desenvolvida para facilitar a descoberta e o acesso a recursos de informação em bibliotecas e instituições acadêmicas. Funciona como um portal de pesquisa, permitindo que os usuários encontrem e explorem uma ampla variedade de materiais, como livros, periódicos, artigos, mídia digital e muito mais. Sua interface intuitiva e recursos avançados de pesquisa tornam a experiência do usuário mais eficiente e produtiva.

Entre suas funcionalidades principais, o VuFind oferece recursos de pesquisa avançada, que permitem aos usuários refinar seus resultados por meio de facetas, filtros e ordenações personalizadas. Além disso, integra-se a diferentes sistemas de bibliotecas e catálogos, permitindo uma ampla cobertura de recursos de informação.

[3] <https://vufind.org/vufind/>

Também oferece recursos de gerenciamento de empréstimos, reserva de materiais, recomendações de conteúdo e suporte para diferentes idiomas e tipos de mídia.

O VuFind é distribuído sob a licença GNU, ou seja, é um software livre e de código aberto. Isso permite que instituições e desenvolvedores personalizem e modifiquem o sistema de acordo com suas necessidades específicas, além de contribuir para o desenvolvimento contínuo da plataforma por meio de colaboração e compartilhamento de código-fonte. Sua natureza flexível e sua comunidade ativa de usuários e desenvolvedores fazem do VuFind uma solução poderosa e versátil para organizações que buscam melhorar o acesso e a descoberta de informações em ambientes bibliotecários e acadêmicos. A Figura 3.3 abaixo apresenta a interface de pesquisa.

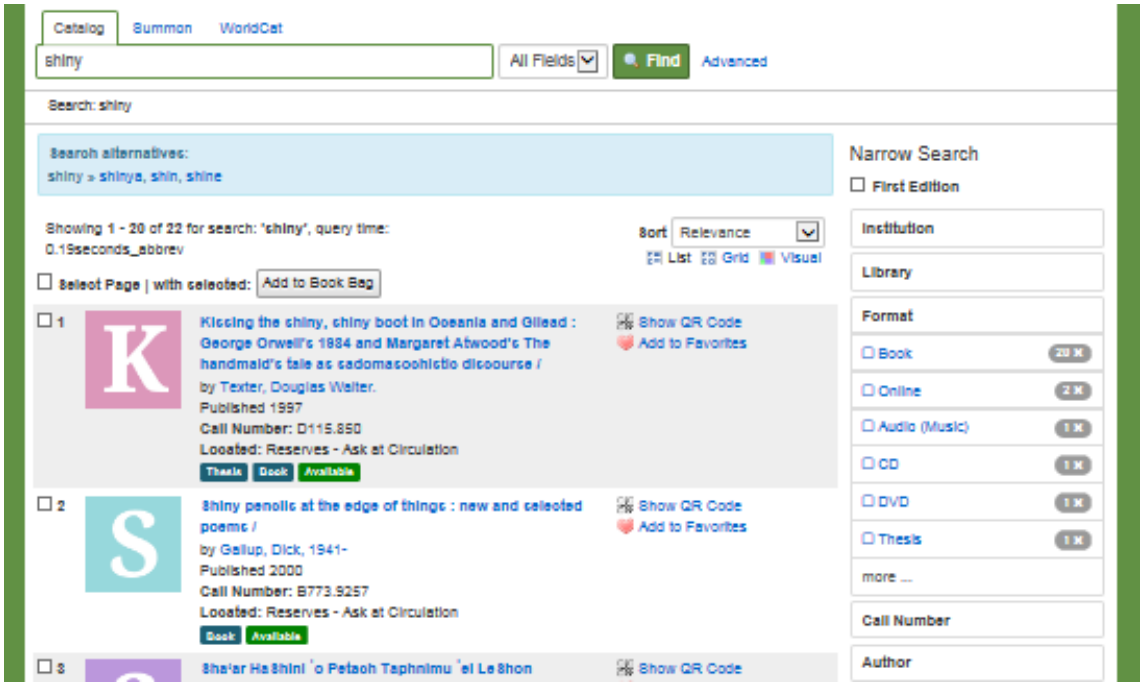


Figura 3.3: Interface para pesquisa de documentos.

Fonte: <https://vufind.org/vufind/images/search.png>

3.3 Proposta

Considerando o contexto supracitado, um Sistema de Gestão de Documentos (SGD) é de grande importância em diversos setores da sociedade, especialmente no

ambiente acadêmico. No entanto, apesar dos benefícios significativos que oferece, a implementação de um SGD pode representar um custo elevado. Mesmo com a disponibilidade de soluções como o DSpace, de código aberto e dedicado ao gerenciamento de arquivos, ainda é necessário arcar com os custos de associação. Diante disso, o presente trabalho propõe a criação de um SGD acessível, gratuito e expansível, que permitirá o fácil acesso, compartilhamento e armazenamento de documentos acadêmicos.

Conforme discutido no Capítulo 2, encontrar uma informação que se procura pode não ser fácil, ou simplesmente não resultar no que se espera. Nesse sentido o sistema aqui proposto tem o objetivo de centralizar documentos acadêmicos em um único local, visando facilitar a busca e o compartilhamento de conhecimento.

O sistema oferece uma interface simples e intuitiva na tela inicial, onde todos os documentos aprovados estão disponíveis. Além disso, os usuários têm a opção de usar um filtro avançado para refinar suas pesquisas, tornando a busca por artigos, TCCs, monografias e outros documentos acadêmicos mais eficiente.

A autenticação é um requisito para o envio de documentos para o site, somente usuários cadastrados poderão enviar documentos, garantindo a segurança e a qualidade do conteúdo disponibilizado. Com apenas dois tipos de usuários, administrador e usuário comum, o usuário administrador desempenha um papel crucial, pois pode revisar e tomar decisões sobre os documentos submetidos, aceitando ou recusando-os.

3.3.1 Arquitetura do Sistema

A aplicação adota um modelo monolítico, no qual todas as funcionalidades estão integradas em um único sistema. Essa estrutura consiste em um servidor web e um banco de dados. O servidor web representa o ponto de acesso exclusivo para os clientes, que interagem com ele por meio das interfaces gráficas. Além disso, o servidor gerencia o fluxo de dados dos usuários e dos documentos, estabelecendo conexões diretas com o banco de dados.

A Figura 3.4 representa a arquitetura do sistema dividida em módulos. Cada

módulo será detalhado a seguir.

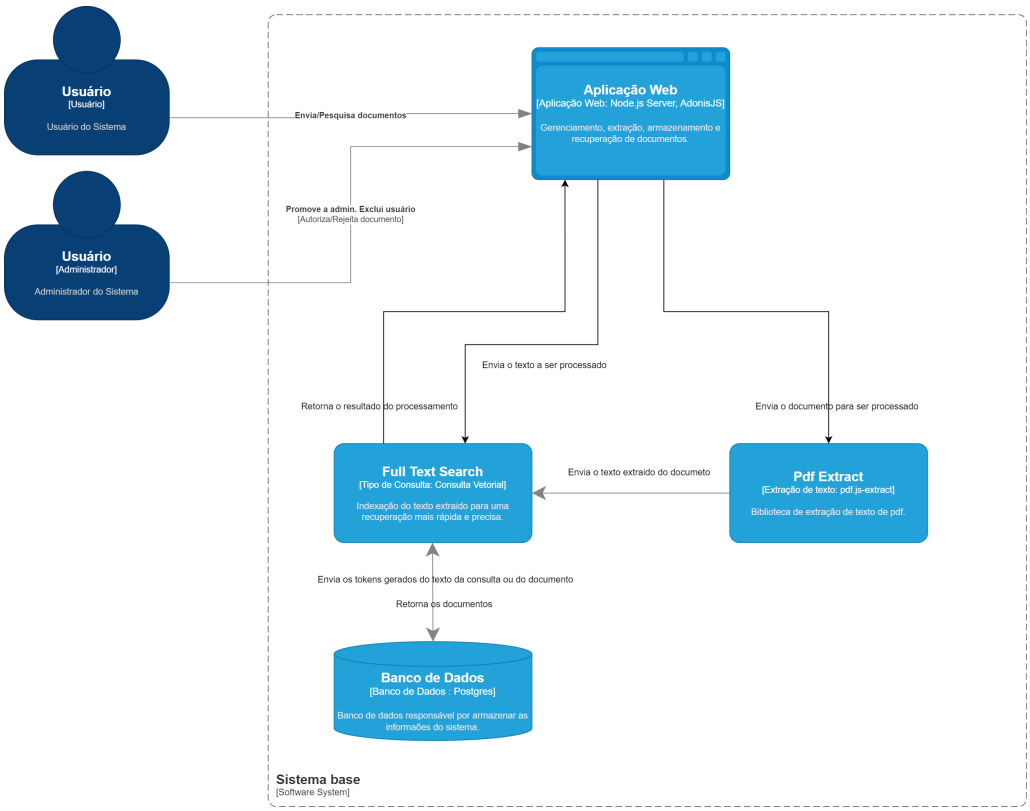


Figura 3.4: Arquitetura representada em módulos.

Fonte: O autor.

3.3.1.1 Aplicação Web

O módulo Aplicação Web representa o sistema em si, contendo todos os códigos, interfaces e controladores necessários para seu funcionamento. Ele serve como o núcleo da aplicação, sendo responsável por integrar todas as funcionalidades e garantir o fluxo adequado de dados e informações. Além disso, esse módulo é crucial para a interação do usuário com o sistema, fornecendo uma interface intuitiva e amigável para facilitar o uso e a compreensão das funcionalidades oferecidas.

3.3.1.2 Pdf Extract

O módulo de extração do Portable Document Format (PDF)[4] é um componente essencial no projeto, responsável por lidar com os serviços relacionados à manipulação

[4] <https://www.adobe.com/br/acrobat/about-adobe-pdf.html>

de arquivos PDF. A escolha do formato PDF é justificada pela sua ampla utilização em documentos acadêmicos.

O PDF foi desenvolvido pela Adobe Systems na década de 1990 e se tornou uma escolha popular devido à sua capacidade de preservar a formatação original do documento, incluindo texto, imagens e gráficos, independentemente do sistema operacional ou do software usado para visualizá-lo.

A estrutura do PDF é baseada em um conjunto de objetos, como texto, imagens e gráficos, organizados em páginas e acessíveis por meio de referências cruzadas. Essa estrutura permite uma representação precisa e consistente do conteúdo do documento em diferentes plataformas e dispositivos.

No entanto, a extração de texto de arquivos PDF apresenta desafios devido à sua natureza semiestruturada. Sendo um formato voltado para o compartilhamento e impressão, o PDF possui uma estrutura vetorial representada por coordenadas, o que torna a extração de texto uma tarefa complexa.

Um texto estruturado é caracterizado por ter uma organização clara e predefinida dos elementos, facilitando a compreensão e análise. Já um texto semiestruturado não segue uma organização predefinida, os dados pode ser apresentados de forma mais livre. Nesse sentido, analisar documentos semiestruturados requer algum nível de processamento adicional.

Para lidar com esses desafios, foi utilizada a biblioteca Pdf.js-extract. Essa biblioteca encapsula o processo de extração de texto de documentos PDF. A extração do texto do documento é feita a partir do momento que o usuário submete o arquivo para o sistema. O texto extraído é salvo no banco de dados para eventuais consultas.

3.3.1.3 Full Text Search

O módulo de pesquisa em texto completo (ou *full text search*) refere-se às técnicas empregadas na busca de conteúdo em bancos de dados. Enquanto a pesquisa tradicional busca apenas por correspondências exatas, a pesquisa em texto completo permite encontrar palavras-chave ou frases em um texto, mesmo que não correspondam

exatamente à consulta realizada. Full text search é essencial para sistemas que lidam com grandes volumes de dados não estruturados, como textos de artigos, documentos e páginas da web, utilizando técnicas como *tokenização* e *stemming* para melhorar a eficácia das buscas.

Apesar de o PostgreSQL possuir ferramentas para *full text search*, neste trabalho foi implementada uma solução própria para esse projeto. A pesquisa em texto completo faz uso de técnicas discutidas no Capítulo 2, como *tokenização* e *stemming*. Para este trabalho, optou-se pela técnica de busca vetorial, que é especialmente eficaz na representação de documentos e consultas como vetores. Essa abordagem permite calcular a similaridade entre documentos e consultas, proporcionando resultados relevantes mesmo em consultas complexas.

A partir do momento em que o documento é enviado para o sistema, é realizada a extração do texto do PDF, seguida pelo tratamento do texto para remover palavras irrelevantes e pela *tokenização*. Na tela de busca por documentos e busca por página, há um filtro em que o usuário pode refinar a busca com opções como Autor, Ano, Título e Conteúdo. A partir desses filtros, o sistema também trata a consulta para aplicar técnicas do modelo vetorial e ranqueamento, com o objetivo de trazer os documentos mais relevantes para uma dada consulta.

3.3.1.4 Banco de Dados

O módulo do banco de dados representa o local central onde todas as informações geradas e manipuladas pelo sistema serão armazenadas. O sistema de gerenciamento de banco de dados desempenha um papel crucial na garantia da integridade, segurança e eficiência no acesso aos dados.

Dentre as diversas opções disponíveis, o PostgreSQL foi escolhido como o sistema de gerenciamento de banco de dados devido à sua reputação como uma poderosa solução de banco de dados relacional de código aberto, amplamente reconhecida por sua robustez, confiabilidade e recursos avançados.

Além disso, para modelar e organizar adequadamente os dados, foi adotado o mo-

delo Entidade-Relacionamento (ER). O modelo ER oferece uma representação visual das entidades do sistema e seus relacionamentos, proporcionando uma compreensão clara da estrutura do banco de dados. Isso facilita o processo de desenvolvimento e manutenção do sistema, pois permite aos desenvolvedores visualizar e compreender facilmente a estrutura de dados.

Uma das principais vantagens do modelo ER é sua capacidade de representar complexidades do mundo real de uma forma simplificada e compreensível. Ele permite identificar e definir claramente as entidades do sistema, seus atributos e os relacionamentos entre elas, o que ajuda a garantir a integridade e a consistência dos dados. A Figura 3.5 abaixo representa a modelagem e as entidades utilizando o ER.

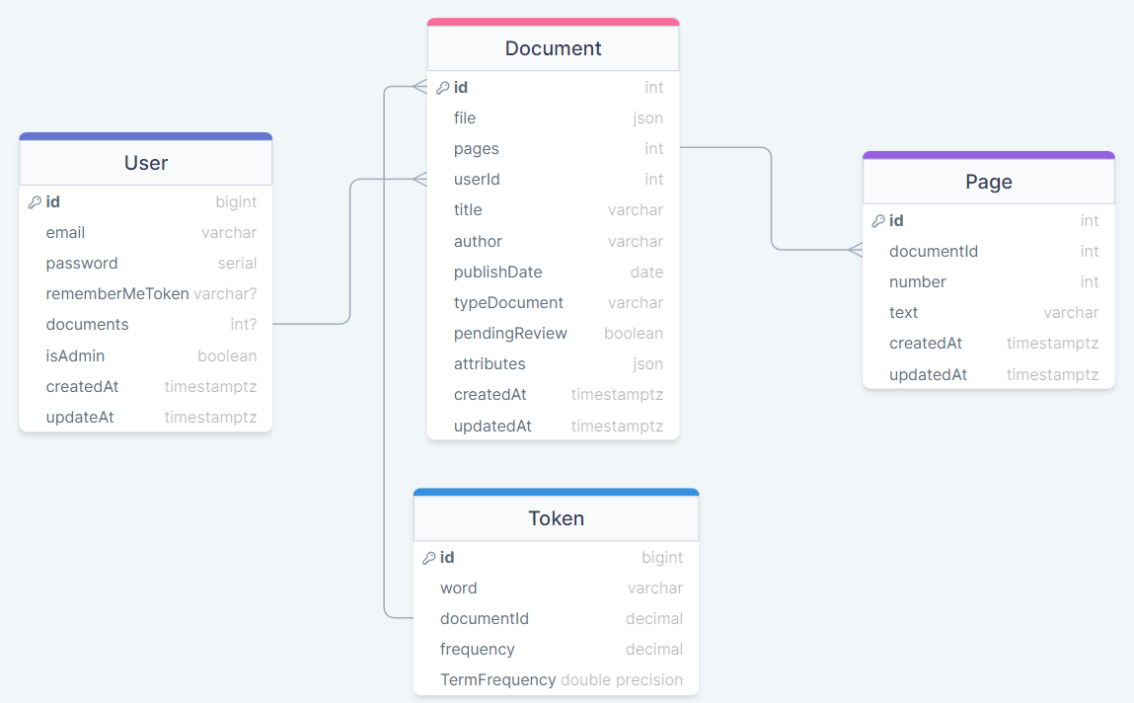


Figura 3.5: Modelo Entidade-Relacionamento para as entidades do sistema.

Fonte: O autor.

3.3.2 Requisitos do Sistema

Os requisitos do sistema definem o que um sistema deve fazer e sob quais restrições. Esses requisitos são categorizados em requisitos funcionais, não funcionais e regras de negócios. Os Requisitos Funcionais (RF) definem as funcionalidades essenciais

que o sistema deve oferecer. Por outro lado, os Requisitos Não Funcionais (RNF) estabelecem as restrições e limitações que a aplicação deve respeitar. Já as Regra de Negócios (RN) descrevem uma diretriz para cada contexto específico de um negócio, sobre qual deve ser o resultado esperado para cada ação ou decisão. Após uma análise foram identificados os seguintes RF, RNF e RN descritos abaixo:

3.3.2.1 Requisitos Funcionais

1. **RF01 - Cadastrar Usuário**

   O sistema deve permitir que um usuário crie uma conta inserindo *e-mail*, senha e confirmação de senha.

2. **RF02 - Autenticar Usuário**

   O sistema deve permitir o acesso do usuário ao serem informados *e-mail* e senha.

3. **RF03 - Tipos de Usuários**

   O sistema deve permitir a existência de dois tipos de usuários: Administrador e User.

4. **RF04 - Visualizar Documentos**

   O sistema deve exibir na tela inicial todos os documentos disponíveis para visualização independente do usuários estar logado.

5. **RF05 - Enviar Documentos**

   O sistema deve permitir que o usuário autenticado envie um documento, anexando o arquivo pdf e informando o tipo de documento, título, autor, abstract e data de publicação.

6. **RF06 - Tela Administração**

   O sistema deve exibir a tela de administração somente para usuários com o perfil de administrador.

7. **RF07 - Documentos para Aprovação**

   O sistema deve permitir que administrador aprove ou rejeite documentos enviados pelos usuários.

8. **RF08 - Excluir Usuário**

   O sistema deve permitir ao próprio usuário excluir sua conta e somente ao administrador excluir conta de outros usuários.

9. **RF09 - Promover a Administrador**

   O sistema deve permitir que administradores promovam usuários ao perfil de administrador.

3.3.2.2 Requisitos Não Funcionais

1. **RNF01 - Proteção dos Dados**

   O sistema deve garantir a segurança dos dados do usuário, suas informações pessoais e documentos.

2. **RNF02 - Desempenho**

   O sistema deve ser capaz de lidar com múltiplos usuários e documentos simultaneamente, sem comprometer o desempenho.

3. **RNF03 - Usabilidade**

   A interface do usuário deve ser amigável e de fácil navegação para garantir uma experiência positiva do usuário.

4. **RNF04 - Disponibilidade**

   O sistema deve estar disponível e acessível para os usuários sempre que necessário, com tempo de inatividade mínimo.

3.3.2.3 Regras de Negócio

1. **RN01 - Confirmação de senha**

   A confirmação de senha deve corresponder à senha inserida no momento.

2. **RN02 - Envio de documento**

   Apenas usuários logados podem enviar documentos.

3. **RN03 - Aprovação de documentos**

   Todos os documentos devem ser aprovados por um administrador antes de ficarem disponíveis para visualização.

4. **RN04 - Tela Administração**

   A tela de administração deve ser visível apenas para usuários com perfil de administrador.

5. **RN05 - Promoção a administrador**

   A promoção de usuários para administradores so pode ser realizada por um administrador.

3.3.3 Casos de Usos

Os Casos de Uso (UC) representam interações entre os atores e o sistema. Eles documentam as principais funcionalidades do sistema do ponto de vista do usuário final. A seguir será listado os casos de usos identificados para o sistema aqui proposto.

1. **UC01 - Cadastro no Sistema**

   Ator: Usuário

   Descrição: Cadastrar um novo usuário no sistema

   Fluxo Principal:

   (a) O usuário acessa a tela de cadastro de usuário

   (b) O usuário fornece um *e-mail*, uma senha e uma confirmação de senha

   (c) O sistema valida os dados informados pelo usuário

   (d) O sistema cria uma nova conta de usuário

   Fluxo Alternativo:

   (a) Se o *e-mail* informado for inválido, o sistema exibirá uma mensagem de erro

   (b) Se a confirmação de senha não for igual à senha, o sistema exibirá uma mensagem de erro

2. **UC02 - Envio de Documento**

   Ator: Usuário

   Descrição: Enviar novo documento em formato pdf para o sistema

   Fluxo Principal:

(a) O usuário faz o login no sistema

(b) O usuário acessa a tela de envio de documentos

(c) O usuário seleciona o arquivo pdf a ser enviado

(d) O usuário preenche os campos obrigatórios, como Título, Autor, Data de Publicação e Tipo de Documento

(e) O sistema valida os dados informados pelo usuário

(f) O sistema armazena o documento enviado

Fluxo Alternativo:

(a) Se o *e-mail* ou senha estiverem errados o sistema irá exibir uma mensagem de dados inválidos

(b) Se o arquivo pdf não for válido, o sistema irá exibir uma mensagem de erro

(c) Se os campos obrigatórios não forem preenchidos, o sistema irá exibir uma mensagem indicando quais campos faltam ser preenchidos

(d) Se o usuário não estiver logado, o sistema irá redirecioná-lo para tela de cadastro

3. **UC03 - Aprovar Documento**

Ator: Administrador

Descrição: Aprovar um documento enviado por um usuário

Fluxo Principal:

(a) O administrador acessa a tela de administração

(b) O administrador visualiza os documentos pendentes de aprovação.

(c) O administrador pode aceitar ou rejeitar o documento selecionado

(d) O administrador aceita o documento selecionado

(e) O documento fica visível na tela inicial

Fluxo Alternativo:

(a) Se o administrador rejeitar o documento, o mesmo não ficará mais visível na tela de documentos pendentes

4. **UC04 - Exclusão de Usuário**

   Ator: Administrador

   Descrição: Este caso de uso descreve como um administrador pode excluir um usuário do sistema.

   Fluxo Principal:

   (a) O administrador acessa a tela de administração

   (b) O administrador visualiza todos os usuários registrados

   (c) O administrador clica no botão de excluir o usuário

   (d) O administrador confirma a exclusão do usuário

   (e) O usuário é excluído do sistema

   Fluxo Alternativo:

   (a) Se o administrador optar por cancelar a ação durante a confirmação, o sistema retorna à lista de usuários sem realizar nenhuma alteração

5. **UC05 - Promoção de Usuário a Administrador**

   Ator: Administrador

   Descrição: Este caso de uso descreve como um administrador pode promover um usuário do sistema a administrador.

   Fluxo Principal:

   (a) O administrador acessa a tela de administração

   (b) O administrador visualiza todos os usuários registrados

   (c) O administrador clica no botão de promover o usuário a administrador

   (d) O administrador confirma a promoção do usuário

   (e) O sistema atualiza os privilégios do usuário selecionado para administrador

   Fluxo Alternativo:

(a) Se o administrador optar por cancelar a ação durante a confirmação, o sistema retorna à lista de usuários sem realizar nenhuma alteração

6. **UC06- Visualização de Documentos Disponíveis**

   Ator: Usuário

   Descrição: Este caso de uso descreve como um usuário pode visualizar todos os documentos disponíveis no sistema.

   Fluxo Principal:

   (a) O usuário acessa a tela inicial

   (b) O sistema exibe todos os documentos disponíveis para visualização.

7. **UC07- Busca por Documentos Específicos**

   Ator: Usuário

   Descrição: Este caso de uso descreve como um usuário pode buscar por documentos com critérios específicos.

   Fluxo Principal:

   (a) O usuário acessa a tela de documentos

   (b) O usuário clica no ícone de filtro

   (c) O sistema exibe as opções de filtros disponíveis, como Título, Autor, Data, Tipo do Documento e Conteúdo

   (d) O usuário seleciona o filtro disponível

   (e) O usuário insere sua pesquisa

   (f) O sistema processa os filtros selecionados e realiza a busca nos documentos

   (g) O sistema retorna uma lista de documentos que correspondem aos critérios de busca

   (h) O sistema exibe os documentos filtrados na tela

   Fluxo Alternativo:

   (a) Se a busca não retornar nenhum documento, o sistema exibe uma mensagem informando que nenhum resultado foi encontrado

Capítulo 4

Implementação do Sistema

Neste capítulo, será detalhada a solução proposta no capítulo anterior. Além disso, será apresentado o padrão de arquitetura Model-View-Controller (MVC), que é fundamental para a organização e manutenção do código-fonte. Por fim, serão exibidas as imagens do sistema de acordo com os princípios de design estabelecidos, juntamente com as decisões estratégicas que orientaram o desenvolvimento desta solução.

4.1 Tecnologias utilizadas

Para o desenvolvimento do sistema proposto, foi utilizado AdonisJS[1], um *framework* completo para Node.js[2]. As interfaces foram criados a partir dos templates do Tailwind CSS[3] em conjunto com o Edge, *template engine* e AlpineJS[4] do próprio AdonisJS. Para a extração e manipulação dos textos dos documentos pdf foi utilizado a biblioteca Pdf.js-extract[5] para Node.js. Para o armazenamento dos dados foi utilizado o banco PostgreSQL[6], um Sistema de Gerenciamento de Banco de Dados (SGBD) gratuito e de código aberto.

[1] <https://adonisjs.com/>
[2] <https://nodejs.org/>
[3] <https://tailwindcss.com/>
[4] <https://alpinejs.dev/>
[5] <https://www.npmjs.com/package/pdf.js-extract>
[6] <https://www.postgresql.org/>

4.1.1 AdonisJS

O AdonisJS é um framework abrangente voltado para o desenvolvimento de aplicações web. Ele simplifica uma variedade de processos de desenvolvimento, como autenticação e roteamento, fornecendo middlewares que podem ser configurados facilmente, além de oferecer suporte para conexão e consultas ao banco de dados usando Object-Relational Mapping (ORM) e a criação de páginas dinâmicas utilizando Template Engine. Uma das características marcantes do AdonisJS é sua arquitetura padrão Model-View-Controller (MVC), que divide o código em modelos, visualizações e controladores, facilitando a manutenção e escalabilidade do projeto.

4.1.2 Node.js

O Node.js é um software de código aberto que possibilita a execução de códigos JavaScript no lado do servidor, independentemente do navegador do cliente. Suas principais características incluem alta escalabilidade, leveza e rápida capacidade de cache, além de ser adequado para o desenvolvimento de aplicações em várias plataformas. Sua utilização é facilitada pelo fato de ser baseado em uma única linguagem e contar com amplo suporte da comunidade.

O Node.js utiliza o V8, também conhecido como Chrome's V8 JavaScript engine, um interpretador JavaScript poderoso desenvolvido pela Google que permite a execução de código de forma assíncrona. Sua arquitetura se baseia no modelo I/O não bloqueante e orientado a eventos, o que possibilita lidar com um grande volume de chamadas sem gerar bloqueios ou gargalos. Além disso, o Node.js oferece o Node Package Manage (NPM), um gerenciador de pacotes que disponibiliza uma ampla variedade de pacotes de código aberto e reutilizável.

4.1.3 TailwindCSS

O TailwindCSS é um framework CSS projetado para simplificar a estilização de sistemas. Ele oferece uma ampla variedade de classes CSS pré-estilizadas, prontas para uso direto no HTML, que permitem estilizar elementos da página, como cores,

tamanhos e sombras, entre outros aspectos. Essas classes são responsivas e têm nomes intuitivos, facilitando sua aplicação e funcionamento para telas de mobile e desktop. No contexto deste trabalho, as classes do TailwindCSS foram utilizadas para estilizar as marcações HTML, construídas com o auxílio do Edge.

4.1.4 AlpineJS

No desenvolvimento das interfaces, foi empregado o AlpineJS, um microframework JS concebido para adicionar comportamento e controle de estado a elementos HTML. A incorporação desse framework simplificou consideravelmente as tarefas relacionadas ao gerenciamento da visibilidade de menus e à execução de funções assíncronas, o que resultou em uma implementação mais eficaz e dinâmica dessas interações no sistema.

4.1.5 Pdf.js-extract

O PDF é um formato de arquivo amplamente utilizado para a apresentação de documentos, caracterizado por sua independência de aplicativos de software, hardware e sistemas operacionais. Cada arquivo pdf encapsula uma descrição completa do layout do documento, incluindo texto, fontes, gráficos vetoriais, imagens rasterizadas e outras informações necessárias para sua exibição. No entanto, extrair texto de um arquivo pdf pode ser uma tarefa desafiadora devido à sua complexidade estrutural.

Diante desse desafio, a biblioteca Pdf.js-extract foi adotada neste trabalho. Esta biblioteca se destaca por ser gratuita, simples e de fácil compreensão, permitindo a leitura e exportação de textos de arquivos pdf, bem como a extração de dados de tabelas estruturadas. Essa escolha foi motivada pela necessidade de uma solução eficiente e acessível para lidar com a extração de informações de documentos pdf.

4.1.6 PostgreSQL

O PostgreSQL é um sistema de gerenciamento de banco de dados relacional de código aberto, conhecido por sua confiabilidade e flexibilidade. Possui uma vasta

gama de recursos, incluindo suporte a SQL avançado e transações ACID, tornando-se uma escolha popular entre os desenvolvedores em busca de soluções de banco de dados mais robustas. Sua facilidade de integração com o AdonisJS é um fator determinante para sua preferência.

4.2 Arquitetura Model View Controller (MVC)

O modelo arquitetural MVC divide a aplicação em três camadas principais: Models (Modelos), Views (Visualizações) e Controllers (Controladores). Neste projeto, a arquitetura MVC foi utilizada com a inclusão da camada de Serviço (Service). A adição da camada de serviço proporciona uma vantagem significativa ao separar as responsabilidades de lógica de negócios e manipulação de dados. Isso promove uma arquitetura mais modular e organizada, facilitando a manutenção e a extensão do sistema, além de permitir que as operações de serviço sejam reutilizadas em diferentes partes da aplicação. Essa abordagem contribui para um código mais coeso, escalável e de fácil manutenção.

É importante mencionar que existem outras arquiteturas que podem ser consideradas, como a arquitetura hexagonal (também conhecida como Ports and Adapters). A arquitetura hexagonal busca isolar o núcleo da aplicação (domínio) de seus detalhes externos (interfaces, bancos de dados, APIs), promovendo uma maior independência e flexibilidade na troca de componentes externos sem afetar a lógica central da aplicação. Embora essa abordagem ofereça vantagens em termos de desacoplamento e testabilidade, para esse projeto, a utilização da arquitetura MVC com a adição da camada de Serviço mostrou-se mais adequada. Isso se deve à sua simplicidade e à facilidade de implementação, que se alinham melhor às necessidades específicas e ao contexto do projeto.

4.2.1 Models

Na arquitetura MVC, a camada de Modelos é encarregada de representar os dados e implementar as regras de negócio da aplicação. Ela engloba os métodos

responsáveis por acessar e manipular os dados da aplicação para operações de leitura, criação, exclusão ou atualização. Adicionalmente, a camada de Modelo interage com o banco de dados, realizando consultas e atualizações, e retorna os dados requisitados pelo Controlador.

- **Document :** O modelo Document é responsável por representar os documentos cadastrados no sistema. Esse modelo possui uma relação de muitos-para-um com o modelo User e uma relação de um-para-muitos com o modelo Page.
- **User :** O modelo User é responsável por representar os usuários cadastrados no sistema. Ele é responsável por indicar os perfils user e administrador. Esse modelo possui uma relação de um-para-muitos com o modelo Document, indicando que um User pode ter muitos documentos associados a ele.
- **Page :** O modelo Page é responsável por representar as páginas que um documento possui. Esse modelo possui um relacionamento de um-para-muitos com o modelo Document, em que uma página pertence a um documento.
- **Token :** O modelo Token é responsável por representar todas as palavras individuais que um documento possui. Esse modelo possui um relacionamento de um-para-muitos com o modelo Document, em que um documento possui 1 ou mais tokens.

4.2.2 Views

As visualizações deste sistema, como mencionado anteriormente, foram desenvolvidas principalmente com o auxílio de três ferramentas: TailwindCSS, AlpineJS e template engine Edge. O AlpineJS foi empregado para criar elementos dinâmicos nas páginas, enquanto o TailwindCSS desempenha o papel de estilizar essas páginas. O template engine Edge, além de renderizar as páginas, possibilitou alcançar alta reusabilidade de código com seu sistema de componentes e layouts. Essa reutilização de código também facilita a manutenção de um estilo consistente em todo o sistema, pois é necessário estilizar apenas um componente e replicá-lo em toda a aplicação.

As principais views do presente sistema, serão mostradas a seguir. A Figura 4.1 mostra a tela inicial onde o usuário, logado ou não, irá ver todos os documentos aprovados disponíveis para visualização. A Figura 4.2 mostra a tela de login em que o usuário deverá informar *e-mail* e senha para autenticar com o sistema. A Figura 4.3 ilustra a tela para cadastro de um novo usuário. A Figura 4.4 mostra a tela para envio de um documento para o sistema. A Figura 4.5 ilustra a tela com os documentos aguardando aprovação por um administrador.

Todas as views a seguir são disponíveis apenas para administradores. A Figura 4.6 ilustra a tela com um overview e os últimos registros do sistema. A Figura 4.7 ilustra a tela com todos os usuários do sistema, podendo o administrador promover um usuário a administrador. A Figura 4.7 ilustra a tela com todos os documentos aguardando aprovação, podendo ser aprovados ou não.

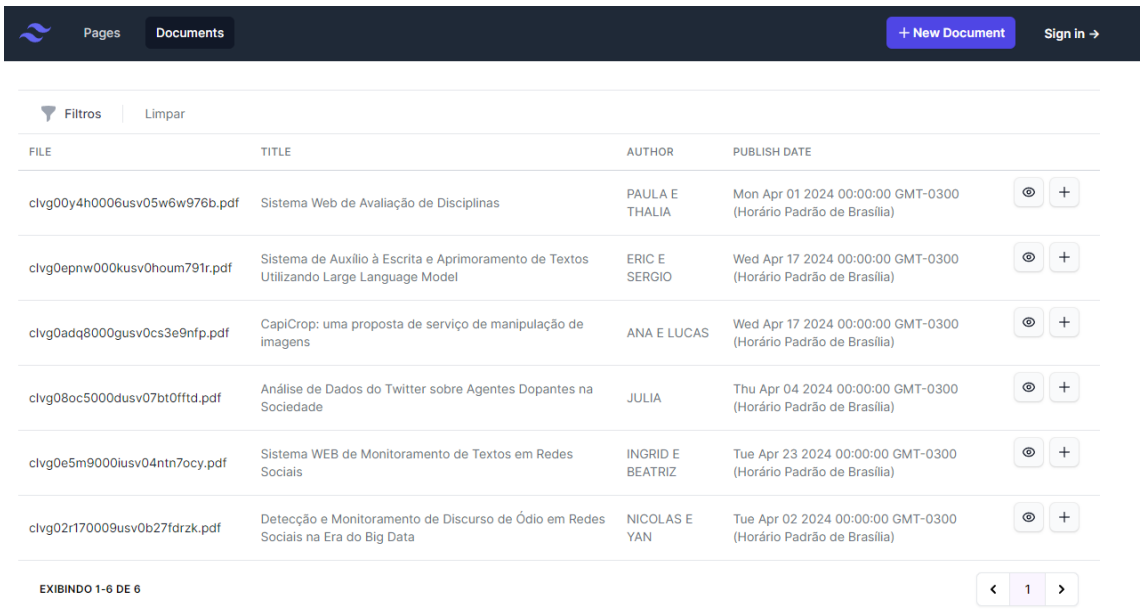


Figura 4.1: Tela inicial onde será mostrado todos os documentos aprovados disponíveis para visualização.

Fonte: O autor.

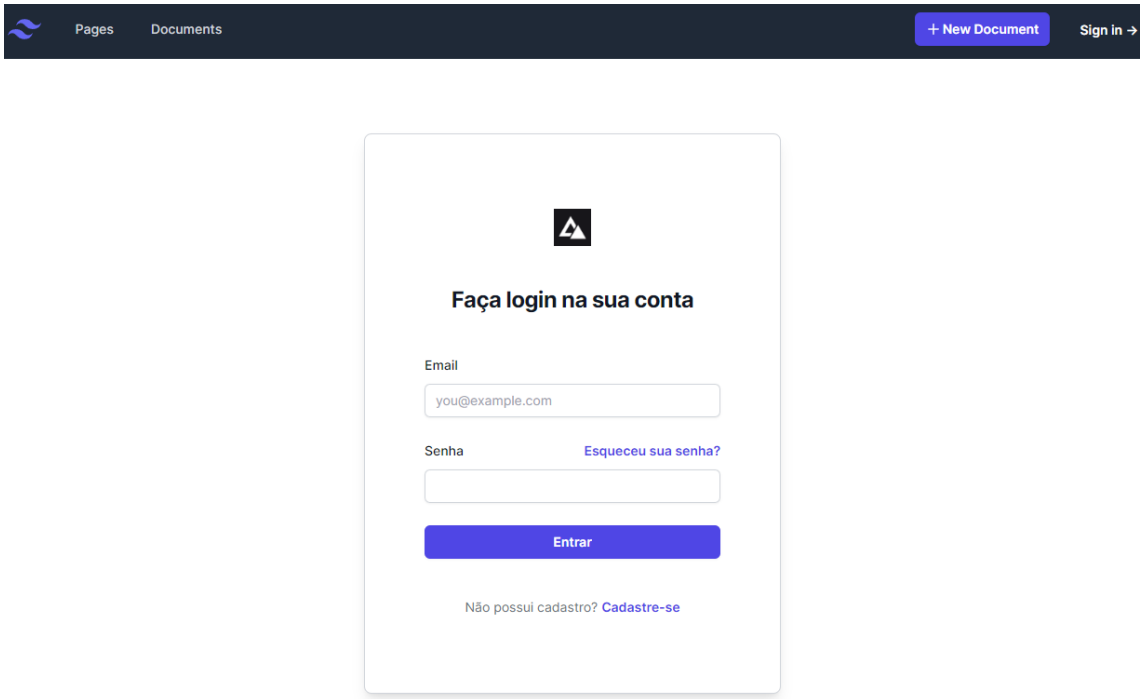


Figura 4.2: Tela de login.

Fonte: O autor.

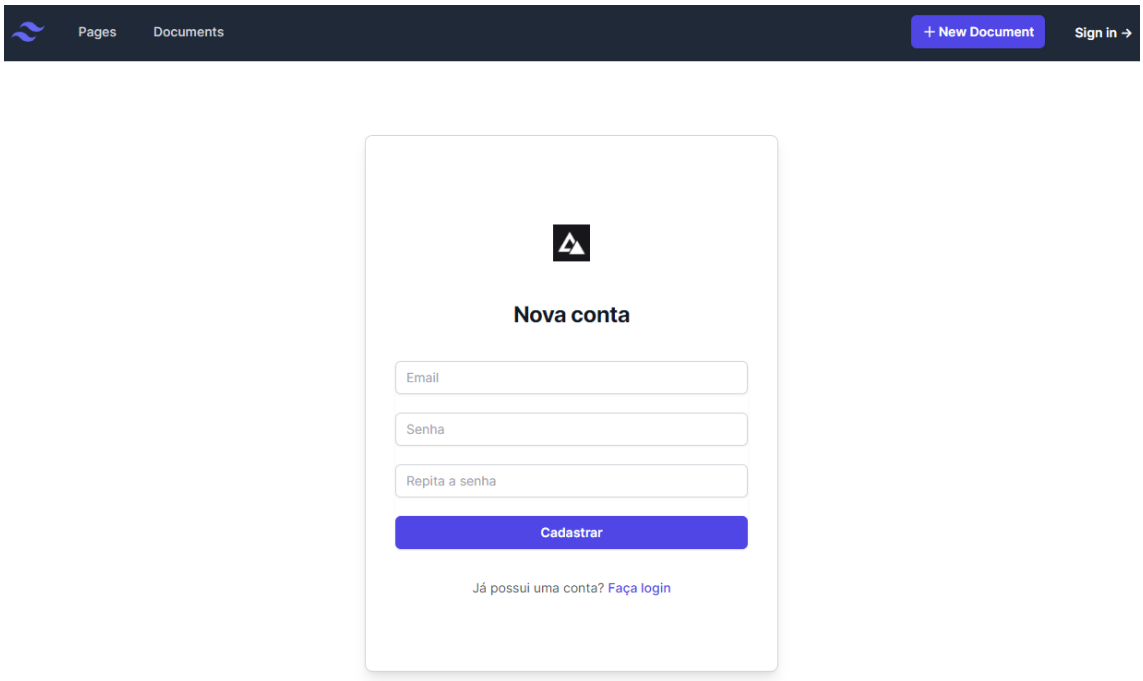


Figura 4.3: Tela para cadastro de um novo usuário.

Fonte: O autor.

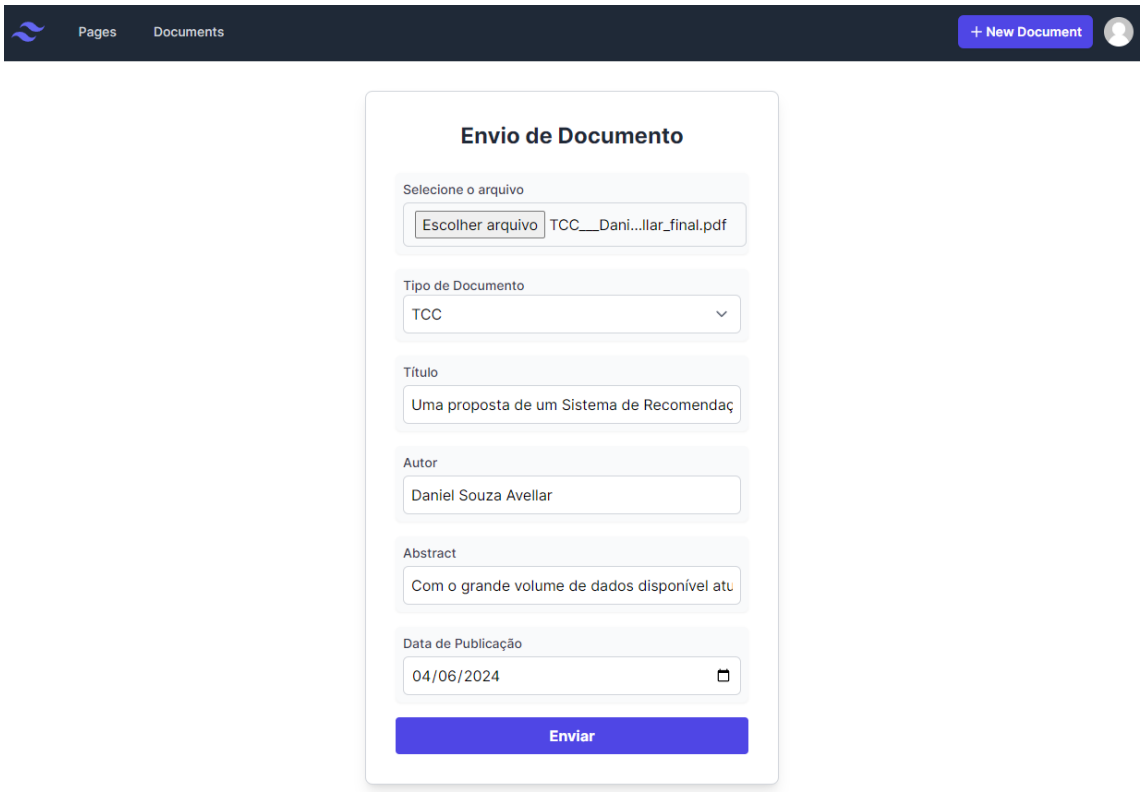


Figura 4.4: Tela para envio de um novo documento.

Fonte: O autor.

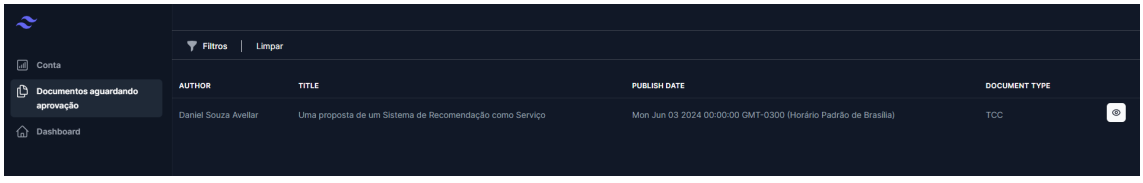


Figura 4.5: Tela com os documentos aguardando aprovação do usuário, disponível para qualquer usuário com login e autenticado no sistema.

Fonte: O autor.

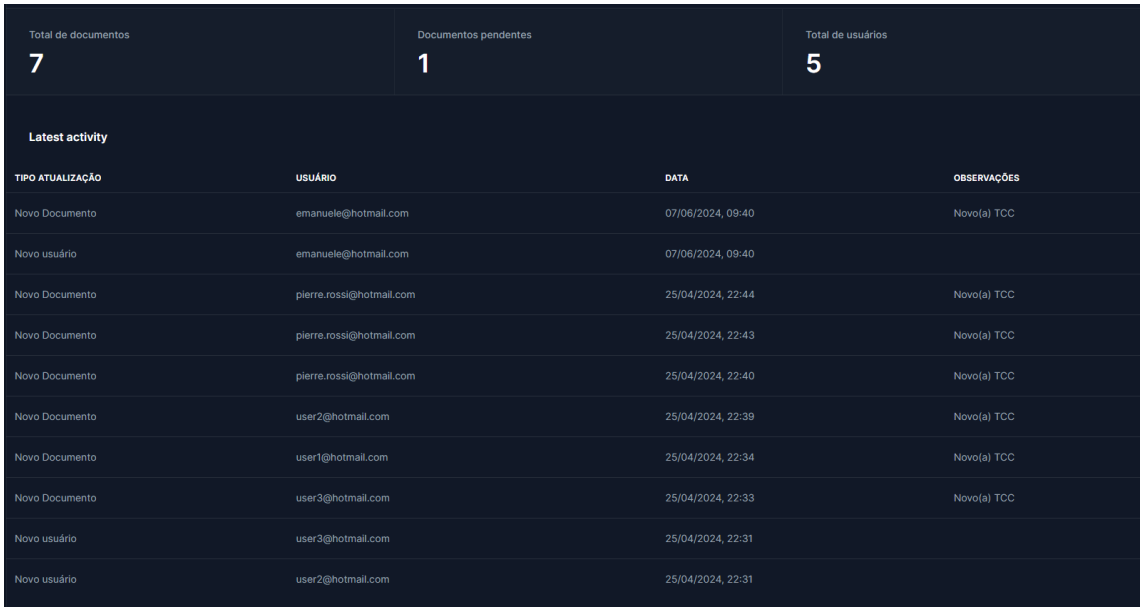


Figura 4.6: Tela com um resumo do sistema, disponível apenas para os administradores.

Fonte: O autor.

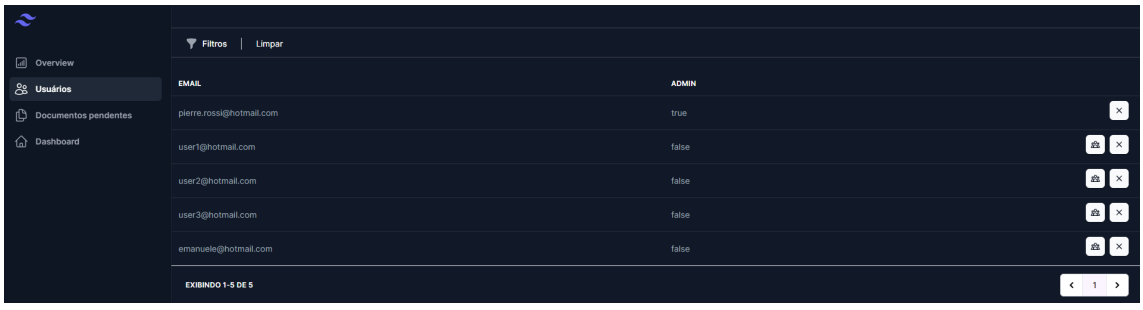


Figura 4.7: Tela que lista todos os usuários do sistema, disponível apenas para administradores.

Fonte: O autor.

Figura 4.8: Tela com todos os documentos pendentes, disponível apenas para administradores.

Fonte: O autor.

4.2.3 Controllers

A camada de Controladores, também conhecida como Controllers, desempenha um papel intermediário entre as Visualizações e os Modelos. Sua principal função é interpretar as solicitações do usuário e coordenar a interação entre as outras camadas. Além disso, os Controladores gerenciam todo o fluxo e a lógica da aplicação.

Antes de ser processada pela Controller, uma solicitação do usuário passa por um Middleware e uma Rota. O Middleware atua como um filtro inicial, preparando a solicitação antes de ser encaminhada para a Rota. Esta última é responsável por associar a requisição feita a uma URL específica a um método correspondente na Controller. Em seguida, o Controlador interage com o Modelo para executar as ações solicitadas. Essas ações podem incluir atualizações nos dados do Modelo, operações de inserção ou busca de informações.

Após atualizar o Modelo, a Controller notifica a Visualização para que as telas do sistema sejam atualizadas de acordo com as alterações realizadas. Essa comunicação entre Controller e View permite que o usuário veja as mudanças refletidas na interface do sistema em tempo real. A Figura 4.9 representa os controladores utilizados neste sistema. A Figura 4.10 demonstra algumas rotas utilizadas pelos controladores.

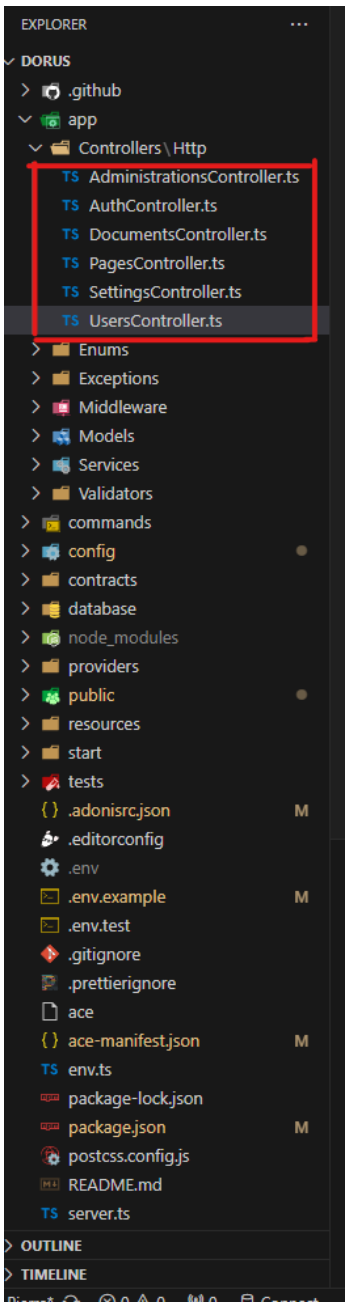


Figura 4.9: Controladores utilizados no sistema aqui proposto.

Fonte: O autor.

```
23  Route.get('/', async ({ response }) => {
24    return response.redirect().toRoute('pages.index')
25  }).as('index')
26
27  Route.group(() => {
28    Route.get('/', 'PagesController.index').as('index')
29  })
30    .prefix('/pages')
31    .as('pages')
32
33  Route.group(() => {
34    Route.get('/', 'DocumentsController.index').as('index')
35    Route.get('/new', 'DocumentsController.create').as('create')
36    Route.post('/', 'DocumentsController.store').as('store')
37    Route.get('/:id', 'DocumentsController.show').as('show')
38    Route.get('/:id/edit', 'DocumentsController.edit').as('edit')
39    Route.patch('/:id', 'DocumentsController.update').as('update')
40    Route.delete('/:id', 'DocumentsController.destroy').as('destroy')
41  })
42    .prefix('/documents')
43    .as('documents')
44
45  Route.group(() => {
46    Route.get('/', 'AuthController.showLogin').as('showLogin')
47    Route.post('/', 'AuthController.login').as('login')
48
49  }).prefix('/login')
50    .as('login')
51
52  Route.group(() => {
53    Route.get('/', 'AuthController.showLogout').as('showLogout')
54    Route.post('/', 'AuthController.logout').as('logout')
55
56  }).prefix('/logout')
57    .as('logout').middleware('auth')
58
59  Route.group(() => {
60    Route.get('/new', 'UsersController.create').as('create')
61    Route.post('/', 'UsersController.store').as('store')
62    Route.patch('/:id', 'UsersController.update').as('update')
63    Route.delete('/:id', 'UsersController.destroy').as('destroy')
64
65  }).prefix('/users')
66    .as('users')
```

Figura 4.10: Algumas das rotas utilizadas pelos controladores.

Fonte: O autor.

4.2.4 Services

A camada de Serviço desempenha um papel intermediário entre o Controlador, o Modelo e, em alguns casos, serviços externos. Sua principal função é encapsular a lógica de negócios da aplicação, promovendo modularidade e reutilização de código. Quando recebe solicitações do Controlador, o Serviço realiza operações específicas relacionadas aos negócios, como interagir com o Modelo para acessar ou manipular dados no banco de dados, ou se comunicar com serviços externos.

Dois serviços foram utilizados: o pdf extract e o *full text search*. No pdf extract, as classes foram projetadas para se comunicar eficientemente com a biblioteca pdf.js, encapsulando todo o processo de extração de texto dos arquivos PDFs e retornando o texto extraído no formato desejado. No *full text search*, foi desenvolvido um código que transforma o texto do documento e da consulta em *tokens*, utiliza o modelo vetorial para classificar as consultas e os documentos como vetores, calcula sua similaridade e retorna os documentos mais correspondentes à pesquisa.

No desenvolvimento desse código, foram utilizadas técnicas abordadas no Capítulo 2. Após a biblioteca pdf extract extrair o texto do documento PDF, o texto é transformado em *tokens* e todos os caracteres que não sejam letras e números são removidos. Em seguida, para cada termo, é calculada sua frequência no documento específico, e esse valor é armazenado no banco de dados. Quando um usuário envia uma consulta para o sistema, a consulta também passa pelo processo de tokenização e remoção de caracteres que não sejam letras e números. Depois disso, são calculados tanto o TF-IDF da consulta quanto dos documentos, e a similaridade é determinada utilizando o método vetorial, ranqueando esses resultados.

Esse método permite identificar quais documentos são mais relevantes para a consulta, retornando-os de forma ordenada. Essa abordagem melhora significativamente a eficácia na recuperação de informações, garantindo que os documentos mais bem ranqueados sejam aqueles mais pertinentes à consulta realizada. Por fim, os documentos são retornados ao usuário em ordem de maior similaridade para menor.

Capítulo 5

Conclusão

Neste capítulo, serão apresentadas as considerações finais acerca deste trabalho, bem como as limitações associadas a ele e as perspectivas para possíveis trabalhos futuros.

5.1 Considerações finais

Neste projeto, foi desenvolvido um sistema web de gestão de documentos acadêmicos com o objetivo de facilitar o acesso, armazenamento e a colaboração na produção científica. O sistema permite que usuários acessem gratuitamente todos os documentos cadastrados, além de fornecer a funcionalidade de envio de documentos.

A implementação de um padrão de arquitetura que facilita a manutenção do código foi fundamental para garantir a eficiência e a escalabilidade do sistema. Além disso, técnicas de cache e de recuperação da informação, descritas no Capítulo 3, foram empregadas para tornar a pesquisa mais rápida e precisa.

O sistema alcançou os objetivos propostos, apresentando uma solução robusta e útil para a gestão de documentos acadêmicos. A utilização de tecnologias modernas garantiram uma experiência de usuário satisfatória, enquanto a arquitetura escalável proporciona flexibilidade para futuras atualizações e expansões. O projeto oferece uma ferramenta acessível e versátil para a comunidade acadêmica, facilitando o

acesso e a colaboração na produção e disseminação do conhecimento.

5.2 Limitações e trabalhos futuros

Durante o desenvolvimento do trabalho, foram observadas algumas limitações que podem afetar a eficácia e usabilidade do sistema, especialmente à medida que a demanda de usuários e documentos aumenta. No entanto, essas limitações podem ser abordadas em trabalhos futuros.

Uma das limitações identificadas está relacionada à escalabilidade e desempenho do sistema. Com o crescimento do número de usuários e documentos, pode-se enfrentar dificuldades para lidar com a carga de trabalho. Para superar esse desafio, seria necessário implementar técnicas avançadas de paralelismo e sistemas distribuídos, distribuindo as tarefas de processamento em vários servidores para melhorar o desempenho e a capacidade de resposta do sistema.

Outra limitação do projeto é a ausência de uma página de feedback dos usuários. Isso impede que os usuários relatem problemas encontrados no sistema ou sugiram melhorias e novos recursos. A inclusão de uma página de feedback permitiria que os usuários expressassem suas necessidades e expectativas em relação ao sistema, contribuindo assim para sua evolução contínua.

Além disso, a falta de uma versão do aplicativo móvel é uma limitação significativa. O desenvolvimento de uma versão móvel facilitaria o acesso dos estudantes aos documentos acadêmicos, permitindo que contribuam com o sistema de forma conveniente a partir de seus dispositivos móveis, independentemente de sua localização ou horário.

Por fim, embora o sistema funcione bem atualmente, é crucial considerar melhorias na implementação das técnicas de recuperação da informação. Com o aumento da demanda, é essencial otimizar consultas, aprimorar algoritmos de busca e utilizar técnicas mais avançadas de indexação e recuperação de documentos para garantir resultados mais rápidos e precisos, mesmo com um grande volume de dados.

Referências

ADHIKARI, D. R. Knowledge management in academic institutions. *International Journal of Educational Management*, Emerald Group Publishing Limited, v. 24, n. 2, p. 94–104, 2010.

BABAN, H.; MOKHTAR, S. Online document management system for academic institutes. In: IEEE. *2010 3rd International Conference on Information Management, Innovation Management and Industrial Engineering*. [S.l.], 2010. v. 4, p. 315–319.

BAEZA-YATES, R.; RIBEIRO-NETO, B. *Modern Information Retrieval*. [S.l.]: ACM Press, 1999.

BAEZA-YATES, R.; RIBEIRO-NETO, B. *Modern Information Retrieval: the concepts and technology behind search*. [S.l.]: Pearson, 2011.

BAEZA-YATES, R.; RIBEIRO-NETO, B. *Recuperação de Informação: Conceitos e Tecnologias das Máquinas de Busca*. [S.l.]: Bookman, 2013.

BÜTTCHER, S.; CLARKE, C. L. A.; CORMACK, G. V. *Information Retrieval: Implementing and Evaluating Search Engines*. [S.l.]: The MIT Press, 2010.

CERI, S. et al. *Web Information Retrieval*. [S.l.]: Springer, 2013.

COMMONS creative. *Licenças*. Acesso em 15/05/2024. Disponível em: <https://br.creativecommons.net/licencas/>.

CROFT, W. B.; METZLER, D.; STROHMAN, T. *Search Engines: Information Retrieval in Pratice*. [S.l.]: Pearson, 2015.

DOMINICH, S. *The Modern Algebra of Information Retrieval*. [S.l.]: Springer, 2008.

EDUCAçãO, M. *Diagramas Venn*. Acesso em 30/01/2024. Disponível em: <https://mundoeducacao.uol.com.br/matematica/diagramas-venn.htm>.

ELMASRI, R.; NAVATHE, S. B. *Fundamentals of Database Systems 7th ed.* [S.l.]: Pearson, 2016.

FERNANDES, M. S.; FERNANDES, C. F.; GOLDIM, J. R. Autoria, direitos autorais e produção científica: aspectos éticos e legais. *Revista HCPA. Porto Alegre. Vol. 28, n. 1,(2008), p. 26-32*, 2008.

FERNEDA, E. *Recall e Precision*. Acesso em 30/01/2024. Disponível em: <https://www.marilia.unesp.br/Home/Instituicao/Docentes/EdbertoFerneda/ri-ead-10-medidas-de-avaliacao.pdf>.

FONSECA, G. H. G. da. *Ciência de Dados Recuperação de Informação*. 2020.

GÖKER, A.; DAVIES, J. *Information Retrieval: Searching in the 21ST Century*. [S.l.]: WILEY, 2009.

GROSSMAN, D. A.; FRIEDER, O. *Information Retrieval: Algorithms and Heuristics*. [S.l.]: Springer, 2004.

GUDIVADA, V. N.; RAO, D. L.; GUDIVADA, A. R. Information retrieval: Concepts, models, and systems. *Expert Systems with Applications*, Elsevier, v. 38, p. 331–401, 2018.

HIEMSTRA, D. Information retrieval models. *Information Retrieval: searching in the 21st Century*, Wiley Online Library, p. 1–19, 2009.

JORDAN, S.; ZABUKOVŠEK, S. S.; KLANČNIK, I. Š. Document management system–a way to digital transformation. *Naše gospodarstvo/Our economy*, v. 68, n. 2, p. 43–54, 2022.

KOWALSKI, G. *Information Retrieval: Architecture and Algorithms*. [S.l.]: Springer, 2011.

MANNING, C.; NAYAK, P. *Introduction to Information Retrieval*. Acesso em 30/01/2024. Disponível em: <https://web.stanford.edu/class/cs276/handouts/lecture2-dictionary-handout-6-per.pdf>.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. *An Introduction to Information Retrieval*. [S.l.]: Cambridge University, 2009.

MITRA, M.; CHAUDHURI, B. Information retrieval from documents: A survey. *Information retrieval*, Springer, v. 2, p. 141–163, 2000.

RIDI. *Koha*. Acesso em 27/04/2024. Disponível em: <https://ridi.ibict.br/bitstream/123456789/1180/1/Conhecendo_o_software_Koha__a_cartilha_simplificada_sobre_o_Sistema_Integrado_de_Gestao_de_Biblioteca_2020.pdf>.

SAVOY, J.; GAUSSIER, E. Information retrieval. 01 2010.

SINGHAL, A. et al. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, v. 24, n. 4, p. 35–43, 2001.

STANLEY, O. E. Information overload: Causes, symptoms, consequences and solutions. *Asian Journal of Information Science and Technology*, v. 11, n. 2, p. 1–6, 2021.

TEAM, S. C. *Inverse Document Frequency (IDF)*. Acesso em 30/01/2024. Disponível em: <https://seo.ai/faq/inverse-document-frequency-idf>.

WILLINSKY, J. *The access principle: The case for open access to research and scholarship*. [S.l.]: Cambridge, Mass.: MIT Press, 2006.