

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**NORM: UM SISTEMA DE BUSCA DE DOCUMENTOS NORMATIVOS
VOLTADO À INSTITUIÇÕES DE ENSINO SUPERIOR BASEADO EM
*FULL-TEXT SEARCH***

LUCAS TAVARES DO NASCIMENTO PEREIRA

**Cajazeiras
2024**

LUCAS TAVARES DO NASCIMENTO PEREIRA

**NORM: UM SISTEMA DE BUSCA DE DOCUMENTOS NORMATIVOS VOLTADO À
INSTITUIÇÕES DE ENSINO SUPERIOR BASEADO EM *FULL-TEXT SEARCH***

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. Francisco Paulo de Freitas Neto.

**Cajazeiras
2024**

IFPB / Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

P436n Pereira, Lucas Tavares do Nascimento.
Norm : um sistema de busca de documentos normativos voltado à instituições de ensino superior baseado em *full-text search* / Lucas Tavares do Nascimento Pereira.– 2024.

52f. : il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2024.

Orientador(a): Prof. Me. Francisco Paulo de Freitas Neto.

1. Desenvolvimento de sistemas. 2. Sistema de busca Norm. 3. Documentos institucionais. 4. Instituição de ensino superior. I. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.

IFPB/CZ

CDU: 004.4(043.2)



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

LUCAS TAVARES DO NASCIMENTO PEREIRA

**NORM: UM SISTEMA DE BUSCA DE DOCUMENTOS NORMATIVOS VOLTADO À
INSTITUIÇÕES DE ENSINO SUPERIOR BASEADO EM FULL-TEXT SEARCH**

Trabalho de Conclusão de Curso apresentado junto ao
Curso Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas do Instituto Federal de
Educação, Ciência e Tecnologia da Paraíba - Campus
Cajazeiras, como requisito à obtenção do título de
Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. Francisco Paulo de Freitas Neto

Aprovada em: **29 de Fevereiro de 2024.**

Prof. Me. Francisco Paulo de Freitas Neto - Orientador

Prof. Me. João Paulo Freitas de Oliveira - Avaliador

IFPB - Campus Cajazeiras

Prof. Me. Michel da Silva - Avaliador

IFPB - Campus Cajazeiras

Documento assinado eletronicamente por:

- **Francisco Paulo de Freitas Neto, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 29/02/2024 20:33:48.
- **Joao Paulo Freitas de Oliveira, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO**, em 01/03/2024 07:46:38.
- **Michel da Silva, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 04/03/2024 17:48:42.

Este documento foi emitido pelo SUAP em 29/02/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 539262
Verificador: e2bf3ddf7f
Código de Autenticação:



Rua José Antônio da Silva, 300, Jardim Oásis, CAJAZEIRAS / PB, CEP 58.900-000
<http://ifpb.edu.br> - (83) 3532-4100

AGRADECIMENTOS

Desejo agradecer a Deus, por permitir que este momento venha a acontecer. Essa foi uma trajetória difícil e, por vezes, somente nele foi possível encontrar forças para não desistir ao enfrentar os piores desafios e pensamentos, trazendo esperanças no anseio por um dia melhor.

À minha família. Minha mãe, Francivania Bezerra Tavares, sempre me acolheu e incentivou minhas escolhas, sem nunca questionar os caminhos que decidi trilhar. Ela me mostrou minha própria capacidade e, como educadora, ressaltou a importância da educação como base fundamental para tudo na vida. Meu pai, Sandro Luiz do Nascimento Pereira, foi quem me introduziu ao mundo da tecnologia e me fez perceber que esse seria o caminho dos meus sonhos. Sempre expressou com orgulho o homem que seu filho estava se tornando. A confiança de ambos em mim sempre será meu conforto.

Ao meu orientador, Prof. Me. Francisco Paulo de Freitas Neto, que com sua experiência e atenção, esteve ao meu lado para garantir a qualidade deste trabalho e me ajudar na minha formação acadêmica.

À todos os meus amigos, em especial, David Beserra, Kauanny Bezerra, Rafael Barroso e Reyson Renan. Estes quatro nunca soltaram a minha mão e sempre estiveram presentes para me ajudar no que fosse preciso. Meus momentos no curso não seriam o mesmos sem eles.

Aos professores e funcionários do campus, expressei minha gratidão por assegurarem que a educação seja acessível a todos. Sua dedicação e profissionalismo são admiráveis e fundamentais para o sucesso de todos os alunos.

RESUMO

O fluxo de alunos ingressos em cursos superiores mostra uma tendência crescente no Brasil, para manter a organização das Instituições de ensino, normas são instauradas constantemente para manter o bom regimento. Tais documentos são disponibilizados digitalmente para todos os alunos, embora não haja garantia de que os sistemas institucionais feitos para disseminação foquem no fácil acesso desses para com o público, devido a presença de interfaces e caminhos pouco favoráveis para obter a informação presente na maioria das aplicações. Este trabalho discorre sobre o desenvolvimento de um sistema que, através de análise e lógica de dados, será capaz de fornecer pesquisas de texto completo, empregando diversas tecnologias aliadas ao *Full-Text Search (FTS)*, inicialmente pensado para atender as demandas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, porém, com a possibilidade de expandir a aplicabilidade para outras instituições. O intuito desta solução é oferecer às Instituições de Ensino Superior meios eficientes para propagar seus documentos da melhor forma possível, bem como tornar o acesso à informação pelos alunos fácil e intuitivo, além de oferecer ferramentas diversas para melhorar a experiência ao acessar alguma norma vigente. Portanto, é notório que buscas baseadas no *FTS* demonstram um grande potencial, visto que resultados concisos e precisos são seu foco principal.

Palavras-chave: Busca por texto completo. Bancos de Dados. Documentos. Ensino Superior

ABSTRACT

The flow of students entering higher education courses shows a growing trend in Brazil, to maintain the organization of educational institutions, standards are established constantly to maintain the good regiment. Such documents are made available digitally for all students, although there is no guarantee that the systems institutions designed for dissemination focus on their easy access to the public, due to the presence of unfavorable interfaces and paths to obtain the information present in most applications. This work discusses the development of a system that, through data analysis and logic, will be able to provide full-text searches, employing various technologies combined with Full-Text Search (FTS), initially designed to meet the demands of the Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, however, with the possibility of expanding applicability to other institutions. The purpose of this solution is to offer Higher Education Institutions efficient means to propagate your documents in the best possible way, as well as making access to information easy and intuitive for students, in addition to offering diverse tools to improve experience when accessing any current standard. Therefore, it is clear that searches based in FTS demonstrate great potential, as concise and accurate results are your main focus

Keywords: Full text search. Databases. Documents. Teaching Higher

LISTA DE FIGURAS

Figura 1 – Número de matrículas em Cursos de Graduação e Sequencial	13
Figura 2 – Exemplo de documento JSON	21
Figura 3 – Primeiros passos para indexação usando Full-Text Search	23
Figura 4 – Criação do índice do Full-Text Search	23
Figura 5 – Diagrama de todos os casos de uso do sistema	28
Figura 6 – Diagrama de atividades do sistema referente à um usuário comum .	30
Figura 7 – Diagrama de atividades do sistema referente à um usuário adminis- trador	31
Figura 8 – Protótipo da página principal de busca	32
Figura 9 – Arquitetura geral do sistema	33
Figura 10 – Captura de tela da página principal do sistema de busca da aplicação	39
Figura 11 – Captura de tela da página principal do administrador da aplicação .	40
Figura 12 – Formulário para adição de um novo documento no sistema	41
Figura 13 – Representação do armazenamento do documento no sistema	41
Figura 14 – Exemplo de busca efetuada na aplicação final	42

LISTA DE QUADROS

Quadro 1 – Exemplo básico do funcionamento do <i>Stemming</i>	22
Quadro 2 – Caso de uso 01: Realizar pesquisa	47
Quadro 3 – Caso de uso 02: Filtrar resultados	48
Quadro 4 – Caso de uso 03: Visualizar documentos recentes	48
Quadro 5 – Caso de uso 04: Visualizar informações dos documentos	49
Quadro 6 – Caso de uso 05: Abrir documento	49
Quadro 7 – Caso de uso 06: Fazer Login	50
Quadro 8 – Caso de uso 07: Adicionar documentos	50
Quadro 9 – Caso de uso 08: Manter documentos	51

LISTA DE CÓDIGOS

Algoritimo 1 – Exemplo de inserção de dados no <i>MongoDB</i>	24
Algoritimo 2 – Exemplo de criação do índice <i>FTS</i> no <i>MongoDB</i>	24
Algoritimo 3 – Exemplo 1 de busca através criação do índice <i>FTS</i> no <i>MongoDB</i>	25
Algoritimo 4 – Exemplo 2 de busca através criação do índice <i>FTS</i> no <i>MongoDB</i>	25
Algoritimo 5 – Criação do índice <i>FTS</i> para uso no motor de busca do sistema .	36
Algoritimo 6 – Configuração do <i>FTS</i> aplicada no sistema	37
Algoritimo 7 – Atribuição de pontuação aos índices do <i>FTS</i>	38
Algoritimo 8 – Configuração do <i>Regex</i> como alternativa para falha na busca <i>FTS</i>	38

LISTA DE ABREVIATURAS E SIGLAS

<i>API</i>	<i>Application Programming Interface</i>
<i>CAP</i>	<i>Consistency, Availability e Partition tolerance</i>
<i>FTS</i>	<i>Full-Text Search</i>
<i>GED</i>	Gestão Eletrônica de Documentos
<i>IES</i>	Instituições de Ensino Superior
<i>IFPB</i>	Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
<i>INEP</i>	Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira
<i>JSON</i>	<i>JavaScript Object Notation</i>
<i>MVC</i>	<i>Model, View, Controller</i>
<i>NoSQL</i>	<i>Not Only SQL</i>
<i>PDF</i>	<i>Portable Document Format</i>
<i>REST</i>	<i>Representational State Transfer</i>
<i>SQL</i>	<i>Structured Query Language</i>
<i>TCC</i>	Trabalho de Conclusão do Curso
<i>UI</i>	<i>User Interface</i>
<i>UML</i>	<i>Unified Modeling Language</i>
<i>XML</i>	<i>Extensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	PROBLEMÁTICA	14
1.2	DESCRIÇÃO DA SOLUÇÃO	15
1.3	OBJETIVOS	15
1.3.1	Objetivo Geral	15
1.3.2	Objetivos específicos	16
1.4	ATIVIDADES	16
1.5	ORGANIZAÇÃO DO DOCUMENTO	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	USABILIDADE	18
2.2	BANCOS DE DADOS <i>NOSQL</i>	19
2.2.1	Bancos de Dados orientado a documentos	20
2.3	<i>FULL-TEXT SEARCH</i>	21
2.3.1	<i>Full-Text Search no MongoDB</i>	23
2.4	Processos de desenvolvimento iterativos	25
3	NORM: UM SISTEMA DE BUSCA DE DOCUMENTOS NORMATIVOS VOLTADO PARA INSTITUIÇÕES DE ENSINO SUPERIOR BASEADO EM <i>FULL-TEXT SEARCH</i>	27
3.1	ANÁLISE	27
3.1.1	Casos de uso	27
3.1.2	Atividades do sistema	29
3.1.3	Protótipos	31
3.2	PROJETO	33
3.2.1	Arquitetura geral do sistema	33
3.2.2	Tecnologias utilizadas	34
4	RESULTADOS	36
4.1	Motor de busca	36

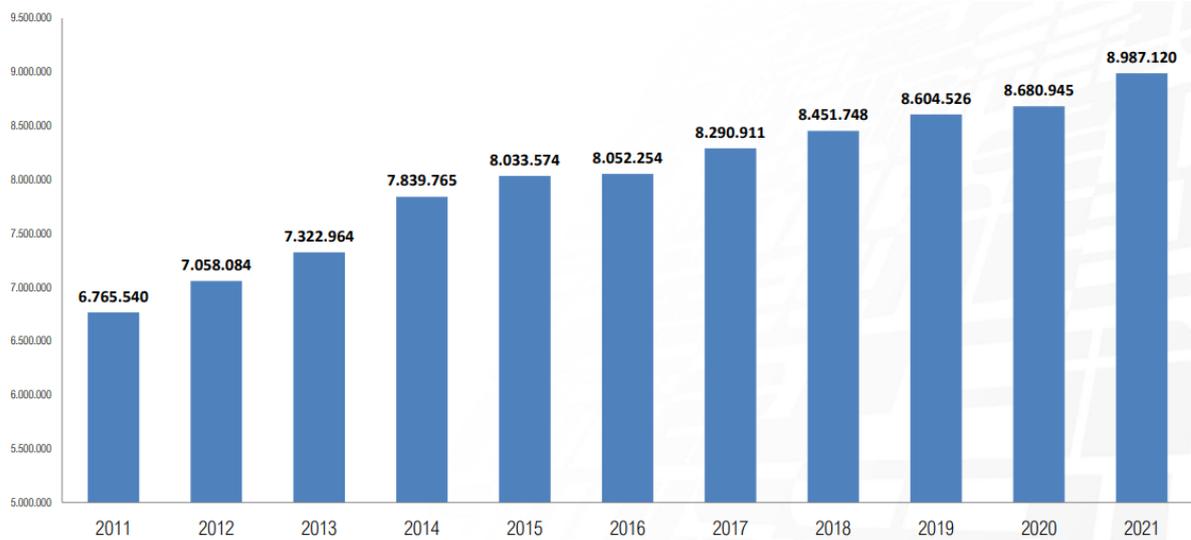
4.2	Interface do sistema	39
5	CONSIDERAÇÕES FINAIS	43
	REFERÊNCIAS	45
	APÊNDICE A – DESCRIÇÃO DE TODOS OS CASOS DE USO DO SISTEMA	47

1 INTRODUÇÃO

O ensino superior no Brasil goza de números crescentes em relação ao aumento de matrículas registrados na última década, conforme descrito nos dados mais recentes do Censo da Educação Superior de 2021 . Pode-se ressaltar o crescimento de 32,8% desde 2011, nesse período, o número de estudantes matriculados ultrapassou a marca de 8,9 milhões, evidenciando a contínua expansão dessa modalidade de ensino (TEIXEIRA, 2022). Essa tendência ascendente reflete não apenas o valor crescente atribuído ao ensino superior, mas também os esforços contínuos para promover o desenvolvimento educacional em nosso país.

A Figura 1 demonstra o panorama geral do número de matrículas entre os anos de 2011 a 2021, de acordo com o INEP. Ao observá-la, é notório o crescimento evidenciado nas matrículas a cada ano.

Figura 1 – Número de matrículas em Cursos de Graduação e Sequencial



Fonte: Adaptado de (TEIXEIRA, 2022)

Aliado ao fato desse crescimento contínuo, ao tratar das Instituições de Ensino Superior, é dedutível que a atualização constante dos meios de propagação das informações se mostra essencial para a evolução do sistema de ensino. Logo, quanto maior o fluxo de estudantes, maior, também, a demanda pelo acesso aos documentos providos pelas instituições. Seguindo esse pretexto, a adoção da Gestão Eletrônica de Documentos (GED) em instituições de ensino é fortemente impulsionada pela digitalização de documentos (SILVA et al., 2003).

A GED envolve o uso e a administração de informações e documentos em um ambiente virtual. Sendo assim, essa abordagem é essencial para otimizar o acesso, proporcionando melhorias significativas para as instituições de ensino. Ao digitalizar documentos, as instituições podem alcançar maior eficiência, agilidade e organização na gestão de informações, contribuindo para aprimorar processos e resultados.

Perante a isso, este Trabalho de Conclusão de Curso (TCC) promove a criação de um sistema capaz de facilitar a busca e o acesso a documentos normativos institucionais, desfrutando de tecnologias atuais para fortalecer ainda mais o alicerce necessário para uma boa educação.

1.1 PROBLEMÁTICA

Na busca por acompanhar os avanços tecnológicos da atualidade, as Instituições Brasileiras de Ensino Superior são prioridades no quesito de atualizações. Mediante a isso, o Ministério da Educação, em abril de 2018, publicou a PORTARIA 315 (BRASIL, 2018), essa, dentre todas suas obrigações, regulamenta a digitalização e a gestão de todos os documentos envolvidos nos processos de ensino-aprendizagem e pesquisa, ou seja, atividades-fim, de instituições de ensino superior, assim como descrito no seguinte:

“Art. 45. Nos termos do art. 104 do Decreto nº 9.235, de 2017 (BRASIL, 2017), os documentos e informações que compõem o acervo acadêmico, independente da fase em que se encontrem ou de sua destinação final, conforme Código e Tabela aprovados pela Portaria AN/MJ nº 92, de 2011 (BRASIL, 2011), deverão ser convertidos para o meio digital, no prazo de vinte e quatro meses, de modo que a conversão e preservação dos documentos obedeçam aos seguintes critérios: I - os métodos de digitalização devem garantir a confiabilidade, autenticidade, integridade e durabilidade de todas as informações dos processos e documentos originais; e II - a IES deverá constituir comitê gestor para elaborar, implementar e acompanhar a política de segurança da informação relativa ao acervo acadêmico, conforme definido nesta Portaria, no Marco Legal da Educação Superior e, de maneira subsidiária, em suas normas institucionais.”

No entanto, a quantidade de documentos que uma só instituição pode dispor em seus arquivos digitais é enorme, o que pode dificultar o acesso a essas informações. Esse cenário pode dificultar o acesso eficiente a dados específicos, especialmente no contexto de mudanças frequentes nas normativas que regem os processos internos da instituição. A imprevisibilidade dessas atualizações contribui para a acumulação de documentos que abordam essencialmente o mesmo assunto, resultando em uma complexidade crescente na busca por informações atualizadas. Logo, o meio digital se

torna um labirinto para aqueles que desejam fazer uma consulta corriqueira a respeito de qualquer assunto desejado.

Além disso, a hostilidade de sites institucionais é um tema recorrente na sociedade atual, estes, na maioria dos casos, não apresentam uma interface intuitiva e caminhos claros para a informação, se tornando mais uma barreira no acesso a esses documentos.

1.2 DESCRIÇÃO DA SOLUÇÃO

Considerando o exposto, torna-se necessário encontrar possibilidades e soluções para lidar com a dificuldade apresentada. Sendo assim, o caminho mais atrativo a se considerar trata da reformulação de todo o sistema de buscas de documentos digitais das IES, que apresentam discrepância em relação ao que se diz prático, porém, não se torna viável uma alteração desse porte por fatores como custos, tempo e manutenção.

Logo, a solução proposta neste trabalho trata da criação de um sistema específico e dedicado para essas buscas, chamado NORM, fornecendo suporte para armazenar os documentos digitais em sua base de dados e os apresentar de acordo com o solicitado, baseando-se na busca textual completa (*Full-Text Search*), ou seja, além do título do documento, todo seu corpo será considerado, para que assim, o resultado final seja o mais preciso possível. Tal proposta foi pensada para atender, inicialmente, o Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, mas com a real possibilidade de expansão para outras IES.

Em suma, o sistema gerencia todos os documentos cadastrados na plataforma da instituição em específico, proporcionando uma organização e acesso mais rápidos e intuitivos. Os usuários poderão realizar buscas personalizadas utilizando filtros, resultando em um sistema mais eficaz em comparação aos métodos convencionais.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Este Trabalho de Conclusão de Curso tem como objetivo a facilitação da divulgação, busca e análise de documentos institucionais através de um sistema de pesquisa eficiente voltado à Instituições de Ensino Superior, visando a utilização desses recursos pelos usuários.

1.3.2 Objetivos específicos

- Desenvolver uma aplicação que possibilite, através de serviços *Web*, o acesso a documentos digitais advindos das IES, bem como todas as configurações acerca do sistema;
- Desenvolver uma aplicação Cliente utilize serviços *Web* para consumir dados e os expor de forma interativa, por meio de uma interface de usuário (UI);
- Possibilitar ao administrador do sistema a liberdade de manipular o acesso aos documentos por outras pessoas;
- Prover aos usuários da aplicação a funcionalidade de busca textual completa, por meio da indexação dos trabalhos em um sistema de busca de documentos.

1.4 ATIVIDADES

No processo de desenvolvimento deste Trabalho de Conclusão de Curso, foram realizadas as seguintes atividades:

- **Elaboração da ideia (A1):** Nessa etapa, foi feita a escolha do tema a ser desenvolvido no projeto, além da análise e estudo dos fundamentos para definir o escopo do projeto.
- **Seleção das Tecnologias (A2):** Nesse ponto, foi realizado um estudo acerca das tecnologias escolhidas para o desenvolvimento do projeto.
- **Estudo do acervo teórico (A3):** Nesse momento, foi feito o embasamento teórico a partir dos conceitos que se relacionam com a proposta deste trabalho.
- **Análise e Projeto de Sistema (A4):** Essa fase envolveu a prototipagem do sistema, buscando compreender o domínio da aplicação e levantar casos de uso relevantes. Além disso, foram elaborados o modelo de dados da aplicação, a descrição dos casos de uso, a definição da arquitetura e a descrição das tecnologias a serem utilizadas durante a implementação.
- **Elaboração do documento (A5):** Nesse momento, foi elaborado todo o documento voltado à qualificação deste trabalho, juntamente à documentação da aplicação.
- **Atualização do Projeto do Sistema (A6):** Nessa atividade, foram feitas algumas correções e adições em relação ao projeto do sistema, a fim detalhar e refinar o projeto.

- **Implementação da aplicação (A7):** Esse momento envolveu a implementação do sistema, ou seja, todo o processo de desenvolvimento que possibilitou a usabilidade e disponibilidade da ferramenta proposta.
- **Atualização do documento (A8):** Após a implementação do código, foi indispensável a atualização deste documento para incluir todas as funcionalidades do sistema, além de documentar todas as funcionalidades implantadas durante o desenvolvimento.

1.5 ORGANIZAÇÃO DO DOCUMENTO

O restante deste trabalho está disposto em mais quatro capítulos. O Capítulo 2, referente à fundamentação teórica, é responsável por explicar todos os conceitos relacionados às tecnologias que envolvem a elaboração do trabalho, com o intuito de trazer afirmações baseadas em fundamentos científicos. O Capítulo 3, que trata dos caminhos para a implementação do sistema, através dos processos de análise de projeto, descrição da arquitetura e tecnologias que deverão ser utilizadas na concepção do sistema. O Capítulo 4, por sua vez, corresponde a demonstração da solução do problema explanado, esta corresponde a apresentação da aplicação finalizada, seus métodos, possibilidades e exemplos, além de demonstrar a funcionalidade do sistema com documentos reais. Por fim, o Capítulo 5 apresenta as considerações final deste documento, além das expectativas em relação ao desenvolvimento do TCC.

2 FUNDAMENTAÇÃO TEÓRICA

Esse capítulo expõe os conceitos e tecnologias essenciais para o entendimento das funcionalidades do sistema proposto. Ao ler este capítulo, é esperado que o leitor esteja familiarizado com os seguintes temas: Usabilidade, Bancos de dados *NoSQL*, Bancos de Dados orientado a documentos, *Full-Text search*, *MongoDB* e Processos interativos.

2.1 USABILIDADE

A facilidade proveniente do uso de uma interface define a qualidade da usabilidade presente neste, tal facilidade pode, também, ser pensada durante o processo de concepção do *design*. Em sistemas *Web*, a usabilidade é uma das principais métricas para estabelecer a sobrevivência de um sistema. Isso acontece em decorrência de alguns fatores: um sistema apresenta dificuldades desnecessárias na sua utilização, a página inicial não é de claro entendimento, os usuários se perdem facilmente, as informações são de difícil compreensão ou as perguntas dos usuários não são respondidas. Todos estes levantamentos, relacionados negativamente com a experiência do usuário, implicam no abandono do sistema, a partir da premissa de que há outras diversas aplicações disponíveis, logo, não se espera que os usuários dediquem tempo à leitura de manuais ou à tentativa e erro na utilização de uma interface (NIELSEN, 2012).

Em concordância com os conceitos citados anteriormente, Barboza et al. (2000) caracteriza a usabilidade como a extensão em que um produto pode ser utilizado em diferentes contextos e por usuários específicos com eficácia e satisfação, de acordo com o contexto de uso. Tendo isso em vista, ainda é mencionada a existência de três maneiras diretas para medir o conceito aplicado desta, assim, pode ser mensurada a partir da análise das características, do processo de interação e, por fim, da eficácia proveniente do uso do serviço. Consequentemente, é possível compreender a usabilidade como a interseção entre eficácia, eficiência e satisfação.

Através do estudo acerca de *sites* do setor público, é viável estabelecer uma conexão com os conceitos previamente mencionados. Essa análise pressupõe que é crucial que o usuário tenha acesso às informações de forma clara e intuitiva, sem se perder na navegação, ou seja, o conteúdo deve atender às necessidades do usuário, a página deve ser de fácil navegação, coerente em seu *design* e acessível a todos, assegurando, assim, a usabilidade. Logo, a preocupação ao oferecer uma interface agradável para o usuário é algo extremamente presente e necessário na atualidade

(INOVAÇÃO - SEBRAE, 2017).

Dessa maneira, um *design* mal elaborado ocasiona transtornos na satisfação do usuário, acarretando *feedbacks* negativos e rejeição do sistema. Portanto, caso uma aplicação apresente problemas de interface que dificultam seu manuseio, seu *design* deve ser repensado, assim, evitando relacionamentos hostis com os envolvidos no sistema (SOUSA, 2007).

2.2 BANCOS DE DADOS NOSQL

Partindo do entendimento sobre o crescimento progressivo do volume de dados, proveniente do desenvolvimento constante da *Web*, o mundo nunca lidou com uma quantidade de informações tão grande quanto atualmente, tendência essa, que vem se intensificando potencialmente ao passar do tempo. Em função disso, diversos meios de mitigar esforços para gerenciar esses dados foram desenvolvidos, dentre as abordagens, surgiram as soluções não-relacionais, sendo parte dessas, os bancos *NoSQL* (*Not Only SQL*), uma ramificação ainda mais atual que carrega diversas promessas e é capaz de armazenar grande volume de dados não estruturados ou semi-estruturados, bancos esses, que estão em constante estudo para que seja comprovada a sua eficácia em relação às outras propostas (DIANA; GEROSA, 2010).

Com isso em mente, de acordo com Lóscio et al. (2011), pode-se citar algumas características que contribuem para a evolução desse sistema e que o tornam distintos em relação aos bancos relacionais:

Uma das suas principais particularidades é a escalabilidade horizontal, que se refere à capacidade de aumentar o número de máquinas disponíveis para o armazenamento e processamento de dados. Essa alternativa surge para desbancar a dependência do poder computacional de máquinas individuais. Outra característica é a flexibilidade do esquema, que permite maior liberdade na organização dos dados, o que facilita o processo da escalabilidade.

A replicação também é suportada, isso permite um processo de recuperação das informações mais rápido, é cabível mencionar as duas abordagens da replicação, a *Master-Slave*, que envolve a escrita em um nó “mestre” e a replica em nós “escravos”, porém, isso pode gerar gargalos dependendo do volume de informações escritas, tendo isso em mente, abordagem *Multi-Master* envolve vários nós “mestres” para evitar esses gargalos, contudo, pode levar a conflitos de dados. Além disso, uma API simples para o acesso às informações é outra característica a ser citada, sendo assim, o foco é oferecer um acesso eficiente aos dados, e não em como eles são guardados, esse

meio permite que qualquer aplicação utilize o banco de forma rápida e eficiente.

Por último, ainda segundo Lóscio et al. (2011), a consistência eventual é um aspecto que está relacionada ao fato de que a consistência não é sempre absoluta entre os diversos pontos de distribuição de dados, o que é baseado pelo sistemas CAP (*Consistency, Availability e Partition tolerance*), que estabelece que, em determinado momento, só é possível garantir duas das três propriedades entre consistência, disponibilidade e tolerância à partição.

Portanto, com a capacidade de lidar com grandes volumes de dados e fornecer um desempenho considerável, os bancos de dados *NoSQL* se tornam adequados para atender às demandas de processamento, baseado na grande disponibilidade de dados oriundos da *Web* (SOARES; MATOS, 2017).

Contudo, embora todos seus aspectos positivos, é importante destacar que os Bancos de Dados *NoSQL* também possuem algumas limitações significativas. Primeiramente, eles não oferecem suporte à linguagem *Structured Query Language (SQL)*, que é amplamente reconhecida como o padrão da indústria. Além disso, há a ausência de recursos em alguns casos, como transações e relatórios, considerados fundamentais para muitas aplicações. Por fim, vale ressaltar que ainda não é suficiente para atender às demandas da maioria dos produtos que já estão consolidados no mercado (NETO et al., 2021). Esses aspectos devem ser cuidadosamente considerados ao avaliar a viabilidade e a aplicabilidade dos Bancos de Dados *NoSQL* em determinados contextos.

2.2.1 Bancos de Dados orientado a documentos

Adentrando nos bancos de dados *NoSQL*, existem alguns modelos de armazenamento, os quais possuem vantagens e desvantagens ao depender para qual finalidade será destinado seu uso, sendo eles os orientado a documentos, chave-valor, orientado a colunas e orientado a grafos (DIANA; GEROSA, 2010).

Sobretudo, o modelo orientado a documentos mostra-se ideal e foi designado para este trabalho, tendo em vista sua principal particularidade, o fato de permitir que cada documento tenha um conjunto de chaves e valores associados a seus campos, além de tolerar a modificação dinâmica da estrutura de cada documento do banco de dados o que torna esse modelo altamente flexível e eficiente, e para que tudo isso ocorra, são utilizados documentos *XML* ou *JSON*, por exemplo, como forma de armazenamento para alocar e retornar os dados (NOGUEIRA et al., 2020).

Cada documento é representado como um objeto que possui um identificador único de campos. Esses campos podem conter diferentes tipos de dados, desde *strings*

e listas, até outros documentos aninhados. Em contrapartida aos bancos relacionais, o modelo em questão não requer um esquema rígido, sendo assim, não é necessário uma estrutura fixa para os documentos armazenados. Tal flexibilidade permite atualizações na estrutura do documento, como a adição de novos campos sem comprometer o banco como um todo (LÓSCIO et al., 2011).

Na Figura 2, é apresentado um exemplo da representação de um documento, destinado a guardar editais de uma instituição, e seus valores, atribuídos individualmente:

Figura 2 – Exemplo de documento JSON

```
{
  ID: "617c9e69b8435822a45a28d4"
  Autor: "IFPB"
  Titulo: "LOREM IPSUM DOLOR SIT AMET"
  Tipo: "Edital"
  DataPostagem: "15/03/2023"
  Destinação: "Institucional"
}
```

Fonte: Elaborado pelo autor (2023)

2.3 FULL-TEXT SEARCH

A priori, o conceito de buscas na *Web* segue um modelo básico de correspondência de caracteres. Dessa maneira, os sistemas de busca mais simples procuram uma correspondência exata dos termos da utilizados na consulta para encontrar os resultados, focando, geralmente, apenas no título do documento, na base de dados. Em contraste a isso, o *Full-Text Search (FTS)* é um método que permite buscar pelo texto dentro de grandes conjuntos de dados e obter resultados que contenham algumas, ou todas, as palavras-chave da consulta, assim, o mecanismo de busca torna-se mais eficiente, por permitir encontrar informações relevantes mesmo quando há variações nos termos utilizados, abrangendo grandes quantidades de texto (MongoDB, s.d.).

À medida que esta tecnologia foi desenvolvida, foram surgindo técnicas para permitir a busca por dados derivados (*Stemming*), que leva em consideração as variações das palavras e a capacidade de desconsiderar palavras que resultam em um número excessivamente grande de correspondências dos documentos (FACHINE, 2020). Para melhor entendimento, é possível analisar o Quadro 1, que mostra um exemplo de como o a ferramenta funciona na sua forma mais básica.

Quadro 1 – Exemplo básico do funcionamento do *Stemming*

Palavra	Derivação (<i>Stemming</i>)
Estudante	Estud
Estudar	Estud
Rainy	Rain
Raining	Rain

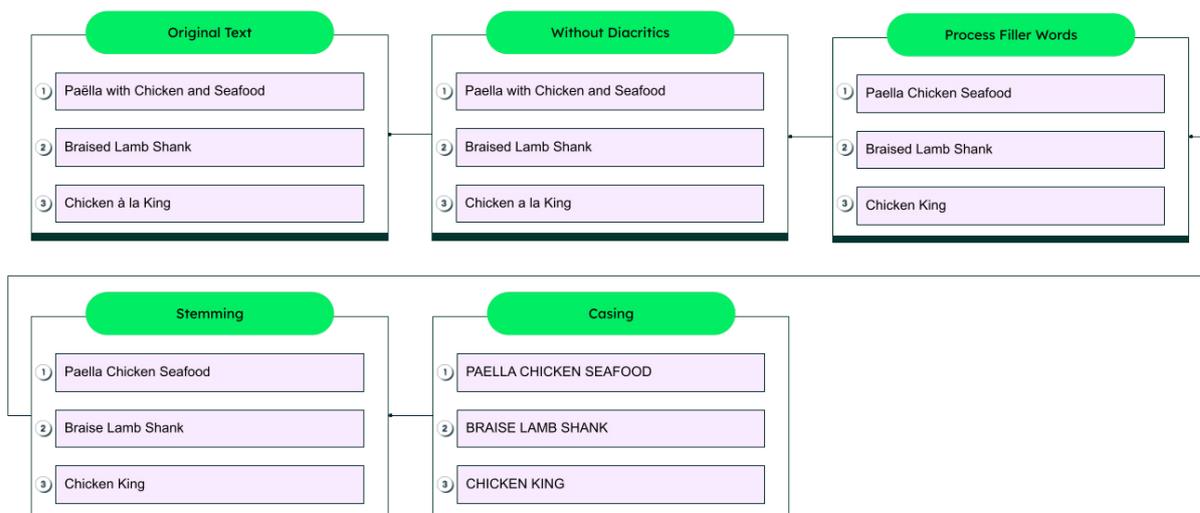
Fonte: Elaborado pelo autor (2023)

Esse processo é definido pela análise das palavras, ao depender do idioma, com o objetivo de reduzi-las a uma raiz, que trará resultados semelhantes, o que permite agrupar palavras variantes que compartilham a mesma raiz, sendo um benefício considerável ao tratar da precisão e abrangência da pesquisa.

Ainda assim, pode existir o uso de palavras consideradas indesejadas no documento, variando a cada idioma, logo, para evitar a obtenção de resultados indesejados, esses termos são ignorados no processo de busca. Essas palavras são conhecidas como *Stop-Words*, “de”, “para”, “e”, entre outros, são exemplos que podem ser incluídos nesse grupo (FACHINE, 2020).

O processo de indexação é a base para o funcionamento do *FTS*, a criação do índice é fundamental para garantir a eficiência da pesquisa. Tal processo garante que cada campo de texto seja percorrido, e para cada palavra analisada, um conjunto de etapas são aplicadas, iniciando pela remoção de sinais diacríticos, como acentos nas palavras. Em seguida, palavras de preenchimento e derivações são desconsideradas, para manter apenas a raiz dos termos, como exposto no Quadro 1. Além disso, é feita a padronização das letras, ou seja, todas as letras da expressão se tornam unicamente maiúsculas ou minúsculas (FACHINE, 2020). Todo esse processo pode ser visualizado de forma didática através da Figura 3.

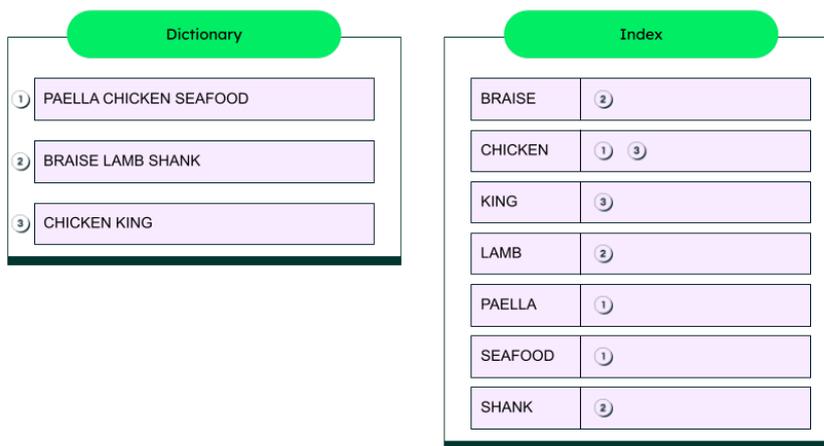
Figura 3 – Primeiros passos para indexação usando Full-Text Search



Fonte: Adaptado de MongoDB

Finalmente, a Figura 4 mostra como o índice é criado ao adicionar cada uma dessas palavras juntamente a sua referência no documento onde ela pode ser encontrada.

Figura 4 – Criação do índice do Full-Text Search



Fonte: Adaptado de MongoDB

2.3.1 Full-Text Search no MongoDB

O *MongoDB*¹ é um dos bancos de dados baseado em documentos mais conhecidos atualmente. Construído para possibilitar consultas únicas e fáceis de serem escaladas horizontalmente, assim, alavancando sua utilização, o que acarreta na sua constante atualização para superar ainda mais suas limitações em relação a

¹ <https://www.mongodb.com>

bancos de dados relacionais. Vale ressaltar a abrangência de tipos de indexação no *MongoDB*, como tipos compostos, multichave, geoespacial, texto, *hash*, entre outros, são suportados (MESUT; ÖZTÜRK, 2022).

Neste trabalho, o tipo de indexação mais explorado será o de texto, logo, para criar um índice desse tipo no *MongoDB*, na sua forma mais básica, é necessário seguir passos simples. Inicialmente, cria-se um *database*, como mostrado no Algoritmo 1, em seguida, uma *collection* para inserir os dados desejados:

Algoritmo 1 – Exemplo de inserção de dados no *MongoDB*

```

1 > use testdb // criando database
2 > db.Test.insert([
3 {string: 'a', name: 'Luis'},
4 {string: 'b', name: 'Alexandre'},
5 {string: 'c', name: 'luís'}
6 ])
```

Fonte: Elaborado pelo autor (2023)

Em seguida, o índice textual deve ser criado, assim como mostrado pelo Algoritmo 2:

Algoritmo 2 – Exemplo de criação do índice *FTS* no *MongoDB*

```

1 > db.Test.createIndex({name: 'text'}, {default_language: 'pt'})
```

Fonte: Elaborado pelo autor (2023)

Logo, o primeiro argumento consiste em um objeto que contém o campo que será indexado (*name*), enquanto o segundo argumento deve ser um objeto de configuração, como o idioma, por exemplo. Nesse caso, foi passado “*portuguese*”, possibilitando que sejam aplicadas técnicas como *Stemming* e *Stop-Words* a partir do idioma escolhido (português). Vale ressaltar que, segundo Fachine (2020), no *MongoDB*, o processo de *Stemming* acontece devido a uma linguagem de processamento de texto, chamada *Snowball*².

Para demonstrar o funcionamento da indexação, podemos realizar consultas a partir dos dados inseridos no *database*, que demonstram a aplicação das particularidades desse tipo de técnica.

No caso do Algoritmo 3, a consulta é realizada referente a “*luis*”, e retorna os objetos que acompanham “*Luis*” e “*luís*”, ou seja, é perceptível a aplicação prática do

² <https://snowballstem.org/>

processo de indexação ao notar que o *FTS* tratou letras maiúsculas e minúsculas da mesma maneira e ignorou sinais diacríticos, o que pode ser relacionado com o que foi exposto na Figura 3.

Algoritmo 3 – Exemplo 1 de busca através criação do índice *FTS* no *MongoDB*

```

1 > db.Test.find({$text: {$search: 'luis'}})
2
3 { "_id" : ObjectId("60c27378c4134d7f0b9d2ccf"), "string" : "a",
   ↪ "name" : "Luis" }
4 { "_id" : ObjectId("60c27378c4134d7f0b9d2cd0"), "string" : "c",
   ↪ "name" : "luís" }

```

Fonte: Elaborado pelo autor (2023)

Ainda é possível realizar mais uma consulta através do Algoritmo 4, que trata de pôr a prova outra aplicação prática do *Stemming*, demonstrado pela Tabela 1. Nessa situação, o *Stemming* associa a consulta que busca o termo “alexandra” com o objeto ligado a “Alexandre”, mostrando que também desconsidera o gênero da palavra.

Algoritmo 4 – Exemplo 2 de busca através criação do índice *FTS* no *MongoDB*

```

1 > db.Test.find({$text: {$search: 'alexandra'}})
2
3 { "_id" : ObjectId("60c27378c4134d7f0b9d2cd1"), "string" : "b",
   ↪ "name" : "Alexandre" }

```

Fonte: Elaborado pelo autor (2023)

Ao analisar as possibilidades advindas do uso da ferramenta descrita, é possível compreender a utilidade e todas as facilidades que podem vir a ser exploradas nos sistemas de buscas atuais, fornecendo a melhor experiência para o usuário final, tendo em vista a otimização dos resultados das possíveis consultas no banco de dados.

2.4 PROCESSOS DE DESENVOLVIMENTO ITERATIVOS

Os métodos ágeis de desenvolvimento de software adotam abordagens iterativas, nas quais o software é construído e entregue em incrementos ao cliente, ou seja, as funcionalidades que dizem respeito a esses incrementos são definidas ao decorrer do processo de desenvolvimento. A escolha do que incluir em um incremento é baseada no progresso do projeto e nas prioridades identificadas, reconhecendo a dinamicidade das suas necessidades (SOMMERVILLE, 2011).

Dessa maneira, ainda segundo Sommerville (2011), as metodologias ágeis mais amplamente utilizadas, como *Scrum* e *Extreme Programming*, seguem uma estratégia de planejamento em dois estágios, alinhados com as fases de iniciação no desenvolvimento orientado por planos e no planejamento do desenvolvimento:

- Planejamento de *release*: projeta um panorama com as características que serão incorporadas em um prazo extenso, geralmente considera-se meses de desenvolvimento;
- Planejamento de interação: adota uma perspectiva de curto prazo, concentrando-se na definição do próximo incremento do sistema.

Em paralelo a isto, para o desenvolvimento deste trabalho, esses estágios foram adaptados da seguinte forma: Novas incrementos eram realizados de acordo com a necessidade do sistema e um conjunto de incrementos definiam uma *release*, ocorrendo, majoritariamente, no intervalo de 15 dias. Logo, a partir da conclusão de uma *release*, uma nova interação definia as próximas metas.

3 NORM: UM SISTEMA DE BUSCA DE DOCUMENTOS NORMATIVOS VOLTADO PARA INSTITUIÇÕES DE ENSINO SUPERIOR BASEADO EM *FULL-TEXT SEARCH*

Considerando o que foi mencionado no Capítulo 1, devido ao aumento no fluxo de alunos nas Instituições de Ensino Superior do Brasil, surgiu a necessidade, por lei, de que essas entidades disponibilizassem os documentos normativos de forma digital, visando facilitar as buscas e torná-las mais precisas. Com base nessa necessidade, foi considerada e realizada a implementação de um sistema online, permitindo que essas instituições e o público possam utilizá-lo para esse fim.

Além disso, é cabível mencionar a semelhança desta aplicação com sistemas de Gerenciamento de Conteúdo, visto que este fornecerá funcionalidades destinadas a alguns usuários específicos que determinam o conteúdo que será exibido ao público, sem a necessidade de conhecimentos concretos de desenvolvimento de *software*.

3.1 ANÁLISE

A elaboração deste trabalho vai além da construção do sistema proposto, tendo em vista a necessidade da aplicação de técnicas de análise, para garantir o melhor desempenho e atendimento de todos os requisitos do sistema, baseado nos fundamentos da análise e projeto de *software*. Assim, técnicas associadas a sistemas de busca, por exemplo, podem ser melhor estudadas e empregadas na construção da aplicação. Portanto, foi de suma importância identificar todos os requisitos do sistema para que as principais funcionalidades sejam determinadas.

3.1.1 Casos de uso

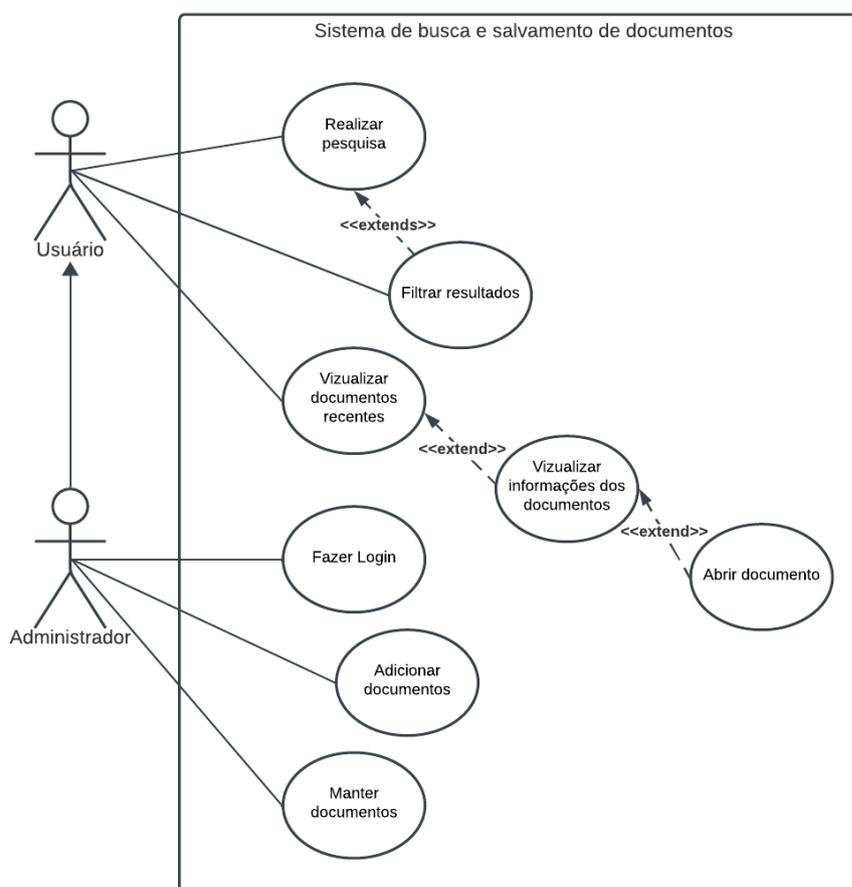
De acordo com o livro "*Writing Effective Use Cases*", de Alistair Cockburn, os casos de uso são descritos como um episódio completo de uso do sistema, visível para um ator do sistema, para realizar um objetivo. Sendo assim, trata-se de uma ferramenta importante na engenharia de requisitos de *software*, visto que ajudam a garantir que este seja desenvolvido de acordo com as necessidades do usuário. Eles são representados graficamente em modelos de casos de uso do *Unified Modeling Language (UML)*, linguagem padrão de modelagem de *software*. Portanto, estes modelos incluem atores, casos de uso e relacionamentos entre eles, que descrevem como os usuários interagem com o sistema para realizar um objetivo (COCKBURN, 2001).

No sistema em questão, existem dois atores: o Usuário e o Administrador. O

Usuário é aquele que desfruta do sistema, ou seja, goza de todas as funcionalidades direcionadas ao público. Por outro lado, o Administrador, além de conseguir desfrutar de todas as funcionalidades de um usuário comum, possui algumas regalias, estas são referentes a manipulação dos documentos disponíveis para o público, mantendo as informações e as atualizando dentro do sistema quando necessário. Tendo isso em mente, o sistema permite a visualização de todo o acervo digital para satisfazer o interesse de qualquer um. É possível observar a ausência de uma autenticação para o usuário comum da aplicação, com o objetivo de facilitar o acesso ao sistema. Portanto, ao visar documentos públicos da instituição, o usuário não precisa estar validado, tendo em vista a obrigatoriedade da disposição dos documentos da instituição para todos, discutido na Seção 1.1

Na Figura 5 pode-se observar o diagrama de casos de uso do sistema e, também, para auxiliar na interpretação e entendimento do esquema, segue abaixo uma breve descrição para cada caso de uso. Porém, se necessário, o leitor pode ter acesso ao detalhamento de cada caso de uso através do Apêndice A.

Figura 5 – Diagrama de todos os casos de uso do sistema



Fonte: Elaborado pelo autor (2023)

Caso de uso 01: Realizar pesquisa

Descrição: O sistema permitirá a pesquisa por termos específicos de acordo com as necessidades do usuário.

Caso de uso 02: Filtrar resultados

Descrição: O usuário terá a opção de filtrar os resultados que serão exibidos de acordo com a disponibilidade no sistema.

Caso de uso 03: Visualizar documentos recentes

Descrição: O usuário terá acesso aos documentos ordenados a partir da sua adição na base de dados do sistema.

Caso de uso 04: Visualizar informações dos documentos

Descrição: O usuário, antes de visualizar o corpo do documento, terá uma breve noção do seu conteúdo através de breves informações, por exemplo, uma breve descrição sobre o que se trata o documento e sua data de publicação.

Caso de uso 05: Abrir documento

Descrição: Ao encontrar o documento desejado em meio ao acervo digital, o usuário poderá abri-lo para melhorar a visualização do seu conteúdo.

Caso de uso 06: Fazer Login

Descrição: O administrador deverá ser autenticado pelo sistema para conseguir acessar e manipular dados sensíveis.

Caso de uso 07: Adicionar documentos

Descrição: O administrador terá acesso a função de adicionar documentos para manter o sistema sempre atualizado, em relação ao acervo da Instituição.

Caso de uso 08: Manter documentos

Descrição: O administrador terá total liberdade para gerenciar e manipular os documentos existentes (atualizar e excluir dados) para manter o sistema em ordem de acordo com o acervo da Instituição.

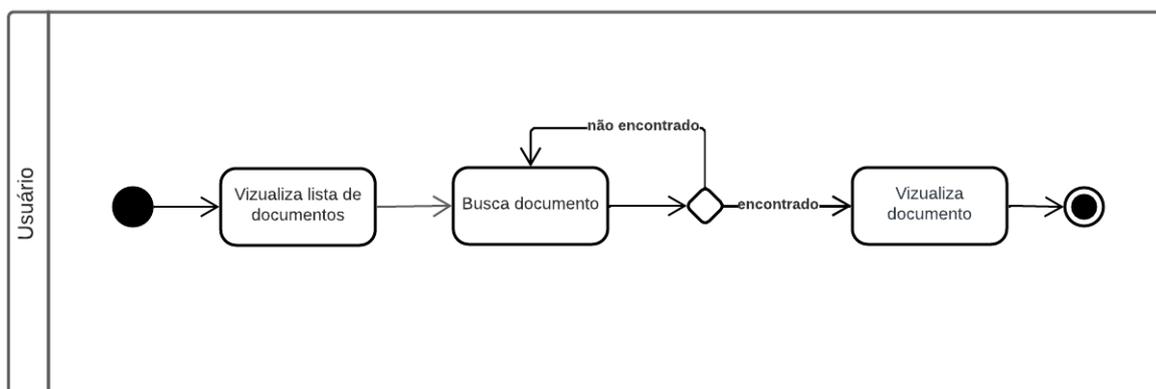
3.1.2 Atividades do sistema

Ainda na perspectiva da análise do projeto, a criação de um diagrama de atividades UML é crucial para definir a ordem das atividades modeladas, proporcionando uma representação dinâmica dos processos na ferramenta. Nesse contexto, esse modelo descreve a sequência de ações executadas, oferecendo uma visão abrangente do comportamento das operações no sistema (FERREIRA; MARTINS, 2010). Essa abor-

dagem contribui para a compreensão detalhada do fluxo de atividades e das interações no ambiente do sistema.

A partir dessa premissa, foram desenvolvidos dois diagramas de atividade, concebidos para atender à ambos os perfis de usuários do sistema. Na Figura 6, é representado o diagrama referente à busca de um documento por um usuário comum da aplicação.

Figura 6 – Diagrama de atividades do sistema referente à um usuário comum

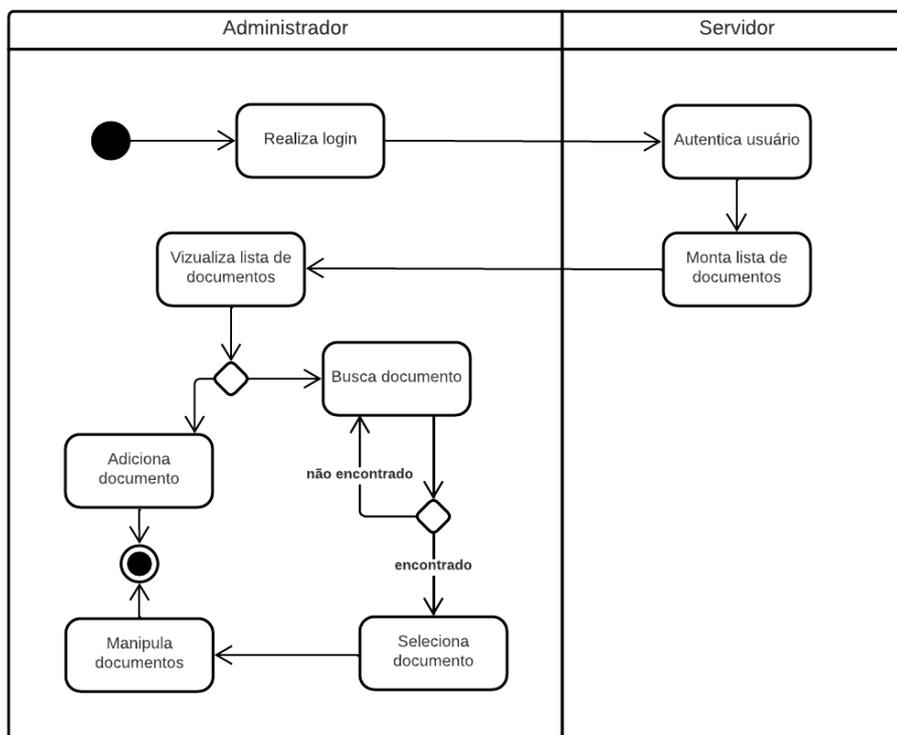


Fonte: Elaborado pelo autor (2023)

Nesta situação, o usuário inicia a interação com o sistema ao visualizar a lista de documentos disponíveis. A partir desse ponto, ele realiza uma busca pelo documento desejado, porém, caso não seja encontrado, pode ser feita a busca novamente até que o usuário esteja satisfeito. Após localizar o documento desejado, o usuário procede à visualização do conteúdo e, finalizada a atividade.

Por outro lado, na perspectiva do administrador, apresentado pela Figura 7, a interação com o sistema se desdobra em algumas novas atividades, tendo em vista a possibilidade de tal usuário ser encarregado de tudo o que tange a gestão e administração das publicações.

Figura 7 – Diagrama de atividades do sistema referente à um usuário administrador



Fonte: Elaborado pelo autor (2023)

Neste cenário, a interação do administrador inicia-se com o processo de autenticação, após isto, o servidor do sistema monta e exibe a lista com todos os documentos disponíveis em seu acervo, para que, assim, as operações de gerenciamento de documentos possam ser realizadas. Em seguida há possibilidade de adicionar um novo documento, por outro lado, pode ser realizada uma busca semelhante à que acontece no caso de um usuário comum, e, a partir disso a seleção de um documento possibilita a manipulação deste, o que gira em torno de editar e excluir o documento selecionado do sistema.

Logo, pode-se notar, através da figura, que ambas operações, “Adiciona documentos” e “Manipula documentos”, culminam no fim desta atividade, porém, o término dessa atividade não implica o encerramento do acesso do administrador, tendo em vista que as operações podem ser realizadas novamente quantas vezes forem necessárias.

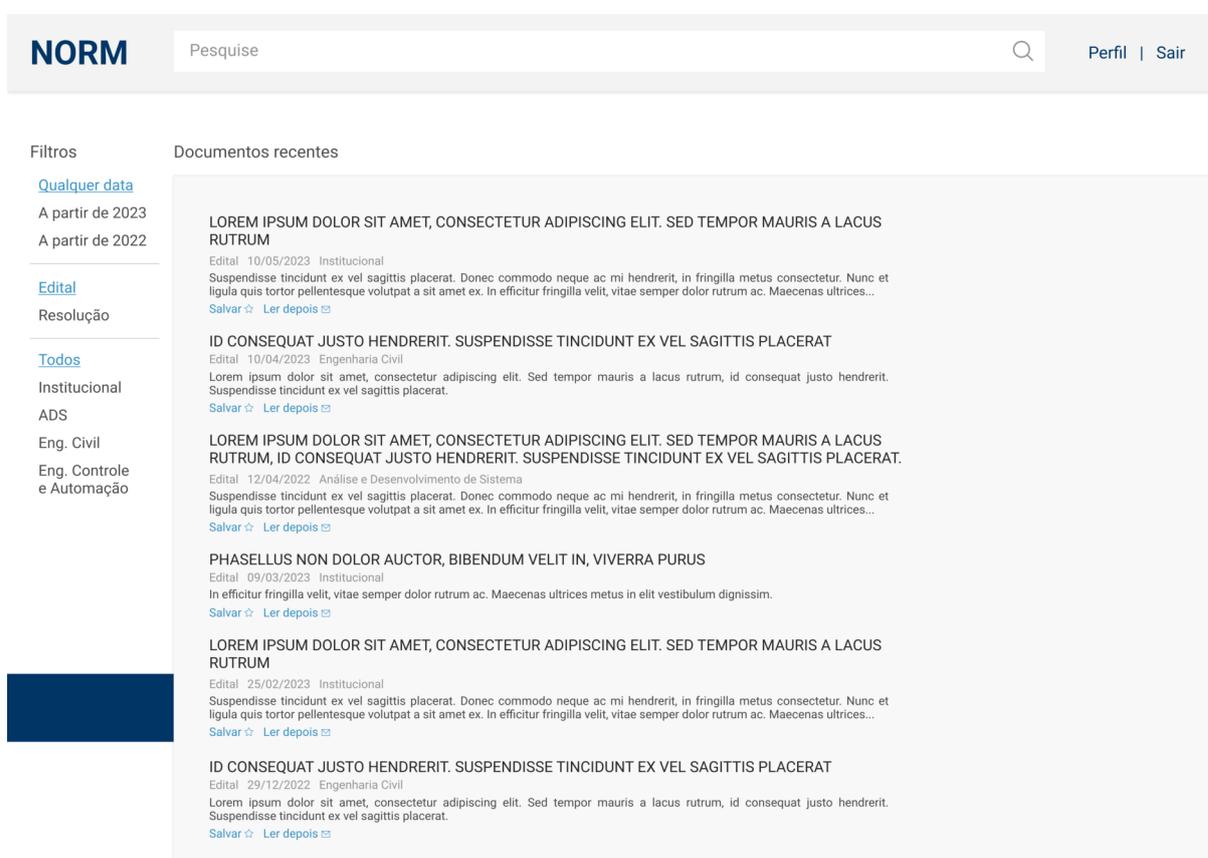
3.1.3 Protótipos

No campo da Engenharia Web, o *Web Design* é uma prática que se concentra na criação de páginas com foco na perspectiva do usuário. Tal prática envolve a combinação de habilidades técnicas e artísticas para melhorar a usabilidade, aparência e visibilidade das páginas (SANTOS, 2009). Sendo assim, almejando a concretização

deste trabalho, o protótipo do sistema foi desenvolvido mediante os requisitos definidos no processo de levantamento, para que, assim, seja possível ter um vislumbre da criação de uma aplicação que atenda todos as necessidades do usuário, ao mesmo tempo em que possui uma aparência visualmente agradável.

A seguir, na Figura 8, é apresentada a página inicial da aplicação, que poderá ser acessada pelo público. Dito isso, para a elaboração do protótipo do sistema, foi utilizada a ferramenta de edição vetorial e prototipagem *Figma*¹, amplamente conhecida no meio do *design*, que permite a construção de telas de um sistema de forma interativa e intuitiva.

Figura 8 – Protótipo da página principal de busca



Fonte: Elaborado pelo autor (2023)

A maioria das telas do sistema passaram pelo processo de prototipagem, porém, esta subseção está destinada a demonstrar que a parte de análise esteve presente durante esta fase do projeto, todavia, caso necessário, o leitor pode visualizar todas as telas elaboradas direcionando-se ao Protótipo do sistema².

¹ <https://www.figma.com/>

² <https://www.figma.com/file/VckJ3vJkVWtqmYt5m3Eb8P/TCC—NORM?type=design&node-id=0%3A1&mode=design&t=IV3YD8mgBSPcInCk-1>

3.2 PROJETO

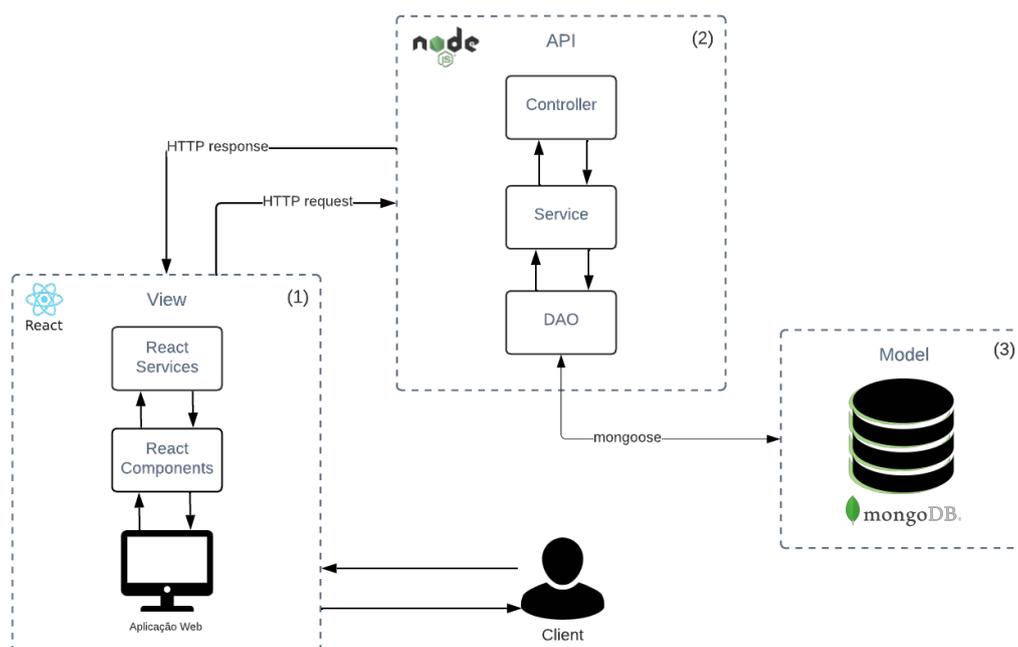
3.2.1 Arquitetura geral do sistema

O desenvolvimento desse sistema envolveu a comunicação e integração de três módulos, para que, assim, possa fornecer um serviço *Web* para o usuário final. Estes módulos, por sua vez, consistem em uma aplicação que fornece os serviços usando protocolos *REST* (*Representational State Transfer*), outro módulo é responsável por consumir os dados destes serviços e oferecer uma representação *UI* para o usuário e, por fim, uma aplicação responsável por analisar e prover o acesso aos documentos, além de manter seu estado no sistema.

Além disso, esse modelo contribuiu positivamente para a efetividade na construção da aplicação. Tendo em vista que, há separação de responsabilidades através dos módulos da aplicação, ajudando na organização do código, o incentivo a boas práticas na construção do sistema e a facilitação da manutenção, já que podem existir vários módulos responsáveis pela representação do sistema, porém, todas as suas características serão tratadas por um único controlador (VILHEGAS, 2011).

Para melhor entendimento dos conceitos descritos, a Figura 9 demonstra a arquitetura geral do sistema e o funcionamento detalhado de cada um dos seus componentes:

Figura 9 – Arquitetura geral do sistema



Fonte: Elaborado pelo autor (2023)

A aplicação atua baseada nestes três módulos da estrutura *MVC* (*Model*, *View*, *Controller*), tendo isso em vista, pode-se desenvolver o entendimento da arquitetura ao analisar o padrão da seguinte forma:

No módulo 3 (*Model*) é representada a camada referente aos dados da aplicação e suas particularidades, sendo responsável por encapsular a lógica de manipulação e pelo armazenamento de dados, bem como a interação com a fonte destes, que serão visualizados na *View* (1) (LEMOS et al., 2013). O sistema em questão irá utilizar o *MongoDB* como banco de dados e motor para realizar as requisições necessárias com base na funcionalidade de pesquisa de texto completo (*FTS*).

O acesso aos dados são provenientes da Interface de Programação de Aplicativos, descritos no módulo 2 (*API*), que foi desenvolvida com base no *Node.js*³ para oferecer os serviços *back-end*, sendo que este oferece um ambiente de execução *JavaScript* do lado do servidor, permitindo a criação de servidores *Web* de maneira eficiente. Nesse contexto, segundo Lemos et al. (2013), o *Controller* recebe as requisições dos clientes e executa a lógica necessária para atendê-las. Ele pode acessar o *Model* (3) para recuperar ou atualizar os dados conforme necessário. No cenário da aplicação para este projeto, tal comunicação acontece através do *Mongoose*⁴, que é responsável por modelar os dados e conectar o *Node.js* ao *MongoDB*.

No módulo 1 (*View*), o usuário tem acesso às informações através da apresentação dos dados. É nesse módulo que ocorre a interação com o sistema, ou seja, a exibição visual da interface e a renderização dos dados fornecidos pelo *Model* (3). Em suma, este módulo não possui lógicas de negócios e é encarregado de oferecer, exclusivamente, a melhor experiência para o usuário final (LEMOS et al., 2013). A par deste conceito, foram utilizadas as bibliotecas *React*⁵ para a componentização da aplicação e *styled-components*⁶ para a estilização do sistema. Tais ferramentas foram escolhidas para compor a interface do sistema, por oferecer recursos avançados de desenvolvimento e um conjunto de meios eficientes para a criação de uma interface visualmente atraente e funcional.

3.2.2 Tecnologias utilizadas

A seguir, nesta seção, serão detalhados alguns aspectos relacionados às tecnologias que foram utilizadas neste trabalho.

³ <https://nodejs.org/en/docs>

⁴ <https://mongoosejs.com/>

⁵ <https://react.dev/>

⁶ <https://styled-components.com/>

A priori, vale ressaltar o uso do *Node.js* como ambiente de execução *Javascript* do lado do servidor. Tal tecnologia é executada sobre o *Google V8*⁷, que interpreta e compila o código para linguagem de máquina, buscando a otimização. Foi através do *Node.js* que os *frameworks* necessários foram gerenciados para a construção e funcionamento da aplicação. Os serviços fornecidos são de suma importância para seguir com a comunicação dos módulos descritos pela arquitetura do sistema.

Seguindo disso, a documentação oficial do *React* propõe que a tecnologia fornece meios de construir interfaces de usuário reutilizáveis seguindo o modelo de peças individuais, chamadas de componentes. Sendo assim essa abordagem simplifica o código repetitivo e torna o processo de desenvolvimento mais simples, em comparação com as técnicas tradicionais de ligação de dados. Logo, o *React* permite que as alterações sejam propagadas de forma estruturada e previsível, o que resulta em um código conciso, organizado e de fácil manutenção, melhorando a produtividade do desenvolvedor e a qualidade da aplicação final.

Além disso, aliado ao *React*, foi utilizado o *styled-components* como principal método para estilizar o sistema como um todo. Essa ferramenta segue, basicamente, o mesmo princípio de componentes do *React*, ou seja, além de se comunicar da melhor forma possível, a legibilidade e aplicação são simplificadas, seguindo a premissa de componentizar o *CSS*. Logo, novas possibilidades e conceitos puderam ser aplicados para melhorar o desenvolvimento e a usabilidade.

⁷ <https://v8.dev/>

4 RESULTADOS

A fim de demonstrar o uso da aplicação, bem como os resultados obtidos, foram utilizados documentos provenientes do portal digital do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras. Os documentos apresentados são uma pequena parte de todo o regimento da Instituição, porém, evidenciam a aplicabilidade do sistema de uma forma prática. Vale ressaltar, também, que todos os meios elencados para realizar a busca são provenientes do *MongoDB*.

Para efetuar a solução proposta neste trabalho, foi elaborado um motor de busca eficaz, baseado no *Full-Text Search*, seguindo os seguintes propósitos:

- Efetuar buscas textuais completas, baseadas na indexação do título e do corpo do documento inseridos no sistema;
- Apresentar os resultados da busca em ordem de correspondência com o termo desejado;
- Apresentar as informações das publicações que correspondem a pesquisa ao usuário, bem como dar acesso ao corpo do documento anexado;
- Construir um esquema de filtros que complementam a busca;
- Ter uma segunda alternativa caso o termo desejado não satisfaça os critérios do *Full-Text Search*.

4.1 MOTOR DE BUSCA

Inicialmente, para implementar o motor de busca, foi preciso elaborar um modelo de dados para as inserções na aplicação. Em seguida, o seguinte esquema de indexação, apresentado pelo Algoritmo 5, foi pensado:

Algoritmo 5 – Criação do índice *FTS* para uso no motor de busca do sistema

```
1 NormSchema.index(  
2   { title: 'text', pdf: 'text' },  
3   { default_language: 'pt', weights: { title: 10, pdf: 5 } }  
4 );
```

Fonte: Elaborado pelo autor (2023)

Neste cenário, seguindo a sintaxe do Mongoose, o título (*title*) e o corpo do documento (*pdf*) foram indexados, logo após, a linguagem foi definida como “pt”, para aplicar as estratégias de busca considerando palavras em português. Além disso, foram definidos os pesos (*weights*) para os campos indexados. Esses pesos foram determinados com o objetivo de indicar a prioridade na ordem dos resultados.

Dessa maneira, pode-se perceber, então, que o sistema dará preferência ao campo do título, de modo que quanto maior a correspondência do termo pesquisado com o este, mais acima na lista de publicações o documento relacionado estará. Contudo, sem deixar de considerar o corpo do texto, já que ele também está definido para agregar precisão à busca, afinal, seu peso também está estabelecido. Essas definições são feitas internamente pelo *MongoDB*, de forma com que os pesos sejam considerados nos cálculos, feitos para estabelecer a posição dos documentos na lista de resultados, assim, exibindo a lista em ordem de correspondência.

Diante disso, deu-se início a implementação do que viria ser o motor de busca elaborado para as situações elencadas. Para obtenção dos resultados desejados, inicialmente, foi implantado a busca textual completa através do operador *\$text*¹, conforme o Algoritmo 6:

Algoritmo 6 – Configuração do *FTS* aplicada no sistema

```
1 $text: {  
2   $search: 'Edital',  
3   $caseSensitive: false,  
4   $diacriticSensitive: false,  
5 };
```

Fonte: Elaborado pelo autor (2023)

À vista disso, o trecho acima tem a finalidade de configurar a aplicação do *FTS* com base na indexação mencionada anteriormente. Logo, os seguintes parâmetros foram definidos: *\$search* recebe o termo pesquisado pelo usuário, por exemplo "Edital", em seguida, *\$caseSensitive* e *\$diacriticSensitive* são configurados como *false*, para indicar que a pesquisa não deve diferenciar letras maiúsculas e minúsculas, bem como não considerar sinais diacríticos. Por fim, obtém-se uma lista de publicações relacionadas ao termo desejado.

Além disso, para atender ao objetivo de listar os resultados de acordo com a correspondência, foi adotada a seguinte estratégia, exposta pelo Algoritmo 7:

¹ <https://www.mongodb.com/docs/manual/reference/operator/query/text/>

Algoritmo 7 – Atribuição de pontuação aos índices do FTS

```
1 score: { $meta: 'textScore' };
```

Fonte: Elaborado pelo autor (2023)

Esta configuração, por sua vez, determina que a ordem dos resultados da pesquisa deve ser organizada com base na pontuação (*score*).

Porém, ao realizar testes após o desenvolvimento desta estratégia, observou-se que, seria interessante incrementar o sistema com uma busca parcial dos termos, tendo em vista que o *Full-Text Search* opera a partir dos radicais das palavras, ou seja, em decorrência de buscas que não se enquadram nesse aspecto, como palavras cortadas, a lista de resultados obtidos era vazia. Assim, este comportamento, mesmo que faça sentido, considerando o comportamento do *FTS*, ainda não era o esperado, um exemplo deste cenário se dá caso um usuário efetuasse a busca pelo termo “Edi”, a fim de obter resultados correspondentes à “Edital”. Para contornar situações semelhantes, uma alternativa foi incorporada ao sistema, como uma válvula de escape, para evitar listas vazias nestes casos específicos.

Assim, a criação de outra lista de resultados foi pensada, esta, por sua vez, recebe outra configuração, conforme exposta pelo Algoritmo 8:

Algoritmo 8 – Configuração do *Regex* como alternativa para falha na busca *FTS*

```
1 title: {  
2   $regex: 'Edi',  
3   $options: 'i',  
4 };
```

Fonte: Elaborado pelo autor (2023)

Sendo assim, o resultado obtido é estabelecido pelo seguinte padrão: *title* representa o campo de título armazenado no banco de dados, este é relacionado aos operadores *\$regex*² e *\$options* que, respectivamente, tratam de verificar a correspondência literal do termo pesquisado e garantir que não haja diferenças entre letras maiúsculas e minúsculas. Todavia, vale ressaltar que, esta estratégia só é consolidada em último caso, devido a relação ser feita somente ao título da publicação, diferente do *FTS*, que também considera o corpo documento, por isso, deve ser vista como um incremento do sistema.

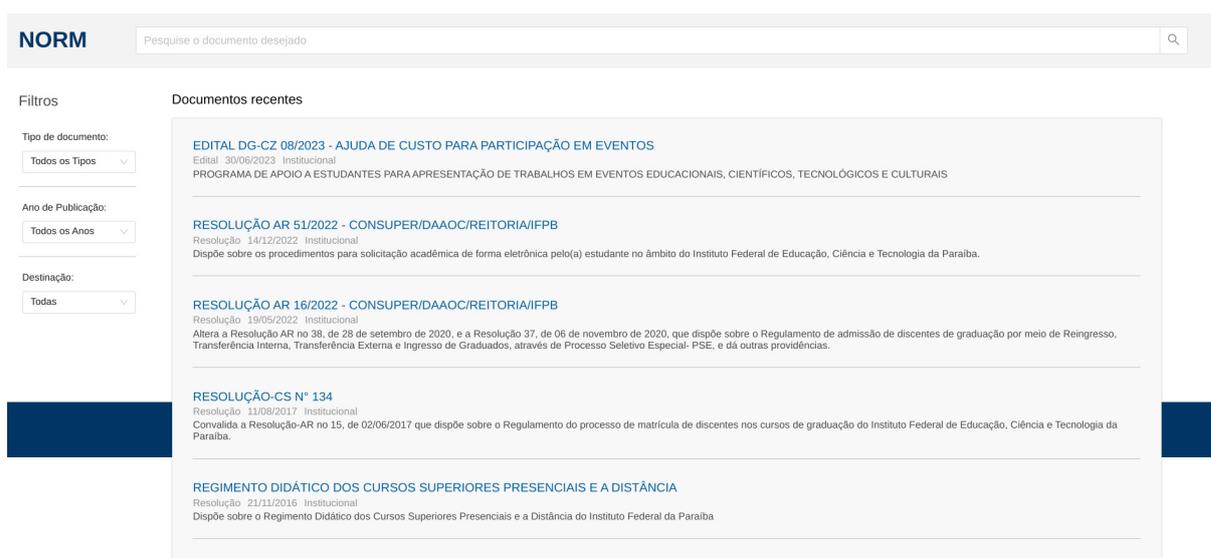
² <https://www.mongodb.com/docs/manual/reference/operator/query/regex/>

Portanto, para estabelecer o cenário ideal, a aplicação monta duas listas, uma referente a pesquisa *FTS* e outra utilizando *Regex*. Em seguida, é realizada uma verificação que estabelece um paralelo entre ambas, onde a lista do *FTS* é a principal e caso esteja vazia, em decorrência ao caso de palavras incompletas, a outra é invocada e exibida para o usuário.

4.2 INTERFACE DO SISTEMA

A seguir, na Figura 10, é apresentada a tela inicial do sistema. Praticamente toda a aplicação foi pensada para funcionar baseada nesta tela, com a finalidade de simplificar a usabilidade, busca e a obtenção de resultados. Nela, é apresentada a lista de documentos disponíveis e os principais métodos de pesquisa (campo de texto e filtros).

Figura 10 – Captura de tela da página principal do sistema de busca da aplicação



Fonte: Elaborado pelo autor (2023)

A visualização do usuário é detalhada e composta por três campos interativos que se complementam para apresentar os resultados desejados. O cabeçalho abriga o campo de texto destinado às pesquisas, enquanto a lateral esquerda disponibiliza filtros para refinar a busca, ambos os elementos funcionam de forma isolada ou simultânea, a depender da preferência do usuário. Na parte central da tela, enfim, os resultados da pesquisa são exibidos de forma organizada em uma lista. Ao encontrar o documento desejado e clicar sobre ele, o usuário é redirecionado para o *PDF* correspondente. Essa abordagem visa proporcionar uma experiência intuitiva e eficaz na navegação e localização de documentos no sistema.

A Figura 11, por outro lado, demonstra a visualização do sistema pela parte do administrador. É possível perceber a semelhança entre as telas do usuário comum e do usuário administrador, porém, as diferenças sutis entre elas são suficientes para definir e assegurar as responsabilidades exclusivas do administrador, em relação à manipulação abrangente da listagem de documentos disponíveis no sistema.

Figura 11 – Captura de tela da página principal do administrador da aplicação

NORM

Filtros

Tipo de documento:

Ano de Publicação:

Destinação:

Documentos recentes

EDITAL DG-CZ 08/2023 - AJUDA DE CUSTO PARA PARTICIPAÇÃO EM EVENTOS
Edital 30/06/2023 Institucional
PROGRAMA DE APOIO A ESTUDANTES PARA APRESENTAÇÃO DE TRABALHOS EM EVENTOS EDUCACIONAIS, CIENTÍFICOS, TECNOLÓGICOS E CULTURAIS
[Editar](#) [Excluir](#)

RESOLUÇÃO AR 51/2022 - CONSUPER/DAAOC/REITORIA/IFPB
Resolução 14/12/2022 Institucional
Dispõe sobre os procedimentos para solicitação acadêmica de forma eletrônica pelo(a) estudante no âmbito do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba.
[Editar](#) [Excluir](#)

RESOLUÇÃO AR 16/2022 - CONSUPER/DAAOC/REITORIA/IFPB
Resolução 19/05/2022 Institucional
Altera a Resolução AR no 38, de 28 de setembro de 2020, e a Resolução 37, de 06 de novembro de 2020, que dispõe sobre o Regulamento de admissão de discentes de graduação por meio de Reingresso, Transferência Interna, Transferência Externa e Ingresso de Graduados, através de Processo Seletivo Especial- PSE, e dá outras providências.
[Editar](#) [Excluir](#)

RESOLUÇÃO-CS N° 134
Resolução 11/08/2017 Institucional
Convulsa a Resolução-AR no 15, de 02/06/2017 que dispõe sobre o Regulamento do processo de matrícula de discentes nos cursos de graduação do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba.
[Editar](#) [Excluir](#)

REGIMENTO DIDÁTICO DOS CURSOS SUPERIORES PRESENCIAIS E A DISTÂNCIA

Fonte: Elaborado pelo autor (2023)

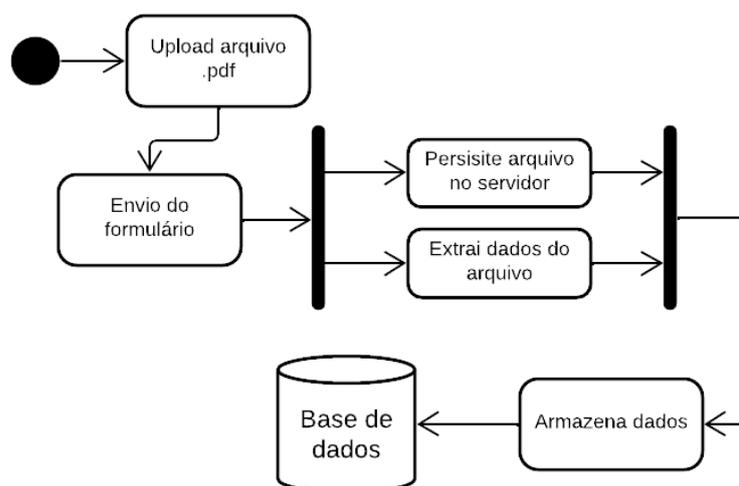
Na Figura 12, pode-se visualizar a interação do administrador para adicionar um novo documento. Essa tela apresenta um formulário estruturado para que sejam inseridos os dados necessários e realizar a publicação do documento. Esse formulário visa facilitar e orientar a inclusão das informações relevantes para a efetivação do processo de publicação no sistema.

Figura 12 – Formulário para adição de um novo documento no sistema

Fonte: Elaborado pelo autor (2023)

Logo, no momento da publicação, uma das principais operações do sistema é realizada. O *Upload* do arquivo *PDF* marca uma etapa crucial para viabilizar a aplicação do *Full-Text Search*, este processo pode ser observado pela Figura 13.

Figura 13 – Representação do armazenamento do documento no sistema



Fonte: Elaborado pelo autor (2023)

Vale destacar que a técnica desenvolvida para essa aplicação atua exclusivamente sobre arquivos *PDF*. A partir da seleção deste arquivo e da inserção dos dados necessários o sistema processa o arquivo da seguinte forma: duas operações são

efetuadas de forma simultânea. A persistência do arquivo no servidor marca o momento em que o documento é salvo, de fato, em uma pasta interna da aplicação, tal pasta guarda todos os documentos inseridos, por outro lado, ocorre a extração dos dados do arquivo, um processo interno do servidor é responsável por converter todo o conteúdo do *PDF* em um *JSON*, possibilitando a legibilidade do documento pelo o *FTS*.

A partir desta conversão, é armazenado no banco de dados o caminho que aponta para o documento específico dentro do servidor e uma *string* que corresponde a todo o texto do *PDF* em questão. Vale ressaltar, também, que nenhum documento é, de fato, salvo no banco, essa estratégia foi adotada para não haver sobrecarga de dados, proveniente do tamanho do arquivo, uma vez que armazenar somente o caminho para o documento é consideravelmente mais leve.

Tendo em vista todos estes aspectos do sistema, foi possível conceber, enfim, pesquisas concretas, que resultam em dados consistentes, dentro da aplicação. Esta percepção é evidente ao conduzir uma busca, como exposto na Figura 14.

Figura 14 – Exemplo de busca efetuada na aplicação final

The screenshot shows a search interface for a system named "NORM". At the top, there is a search bar with the text "trancamento" and a magnifying glass icon. Below the search bar, there are three filter sections: "Tipo de documento:" with a dropdown menu set to "Todos os Tipos"; "Ano de Publicação:" with a dropdown menu set to "2022"; and "Destinação:" with a dropdown menu set to "Institucionais". To the right of the filters, under the heading "Documentos recentes", there are two search results. The first result is "RESOLUÇÃO AR 51/2022 - CONSUPER/DAOC/REITORIA/IFPB" with a sub-heading "Resolução 14/12/2022 Institucional" and a description "Dispõe sobre os procedimentos para solicitação acadêmica de forma eletrônica pelo(a) estudante no âmbito do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba." The second result is "RESOLUÇÃO AR 16/2022 - CONSUPER/DAOC/REITORIA/IFPB" with a sub-heading "Resolução 19/05/2022 Institucional" and a description "Altera a Resolução AR no 38, de 28 de setembro de 2020, e a Resolução 37, de 06 de novembro de 2020, que dispõe sobre o Regulamento de admissão de discentes de graduação por meio de Reingresso, Transferência Interna, Transferência Externa e Ingresso de Graduados, através de Processo Seletivo Especial- PSE, e dá outras providências." Below the search results, there is a solid dark blue horizontal bar.

Fonte: Elaborado pelo autor (2023)

Nesta situação, foi realizada a pesquisa por todos os documentos presentes na base de dados do sistema do ano de 2022, destinados à Instituição que contenham trechos acerca do termo “trancamento”. Destaca-se, assim, a eficácia da abordagem de busca textual completa proposta para a aplicação, evidenciando resultados notáveis.

5 CONSIDERAÇÕES FINAIS

Após a leitura deste trabalho, é nítido o aumento gradativo do fluxo de estudantes nas Instituições de Ensino Superior do Brasil. Nesse contexto, é fundamental que os meios de propagação das informações acerca das IES sejam facilmente acessíveis. Portanto, o projeto deste documento provê a criação de uma ferramenta chamada NORM, um sistema destinado à disseminação e busca de documentos normativos acadêmicos. Dito isso, é válido salientar que o uso desta ferramenta deverá otimizar todo o processo referente ao acesso a esses documentos, trazendo benefícios notáveis para as instituições e usuários do sistema, tendo em vista a facilitação dos caminhos para o acesso a tais normas.

Para tornar o NORM uma realidade, foi utilizado uma série de ferramentas especializadas na construção de aplicações *Web*, sendo o maior destaque do sistema o uso do *Full-Text Search*, que permite com que a busca dos documentos seja realizada da forma mais concisa e precisa possível.

As fases de análise, projeto do sistema e levantamento teórico foram de suma importância para a compreensão da aplicação e estabeleceram uma base sólida para dar continuidade ao projeto. Logo, o aprofundamento acerca de bancos de dados se mostrou fundamental para compreender os meios de manipulação e armazenamento de um grande volume de documentos gerados pelas IES. Com isso em mente, foram concebidas as lógicas necessárias para viabilizar a implementação desse sistema, além de uma arquitetura ideal ter sido elencada.

Além disso, o motor de busca desenvolvido para este projeto oferece uma ampla gama de aplicações. A forma com que o *FTS* foi empregada no NORM, funciona, praticamente, para qualquer outra abordagem, desde que aplicada da maneira ideal, baseando-se apenas na criação de um índice textual no *MongoDB*, visto a maneira com que os documentos são elaborados e dispostos na base de dados. Logo, a criação de uma *API* que não foque em um escopo específico, mas sim em casos mais genéricos, para facilitar ainda mais a aplicação em usos gerais, bem como a disponibilização como uma biblioteca acessível através dos gerenciadores de pacotes se tornam uma ideia promissora para ampliar a disseminação dos métodos e resultados obtidos ao longo do desenvolvimento deste trabalho.

É possível pensar, também, na adaptação deste serviço para possibilitar a aplicação em outras instituições além do IFPB, fazendo com que a disseminação de

aplicações como o NORM auxiliem a maior quantidade de pessoas presentes nos ciclos de ingressos nessas instituições.

Ainda mais, é válido pensar sobre a adição de novas funcionalidades para o NORM, visando atualizações futuras. A adição de um perfil para o usuário é um exemplo, para que possam ser efetuadas operações que dizem respeito apenas à um perfil em específico. Assim, é razoável a criação de um método que permita designar um documento como favorito, o salvando no perfil do usuário, bem como, outras funcionalidade que possam se relacionar a este novo perfil.

Pode-se, então, concluir que a facilitação promovida por esse sistema é de grande relevância, visto que os usuários finais conseguirão ser inteirados de qualquer atualização acerca do regimento que lhe abriga, estando livres das dificuldades para o acesso a sistemas institucionais, além da própria instituição ser capaz de gozar dos meios mais atuais para a propagação da informação.

REFERÊNCIAS

BARBOZA, E. M. F.; NUNES, E. M. d. A.; SENA, N. K. Web sites governamentais, uma esplanada à parte. **Ciência da Informação**, SciELO Brasil, v. 29, p. 118–125, 2000.

BRASIL. Portaria nº 92, de 23 de setembro de 2011. **Diário Oficial da União**, Brasília, DF, 2011. ISSN 1677-7042. Disponível em: <<https://www.gov.br/conarq/pt-br/legislacao-arquivistica/portarias-federais/portaria-no-92-de-23-de-setembro-de-2011>>.

_____. Decreto nº 9.235, de 15 de dezembro de 2017. **Diário Oficial da União**, Brasília, DF, 2017. ISSN 1677-7042. Disponível em: <<https://www.in.gov.br/en/web/dou/-/decreto-n-9-235-de-15-de-dezembro-de-2017-1101286-1101286>>.

_____. Portaria nº 315, de 4 de abril de 2018. **Diário Oficial da União**, Brasília, DF, 2018. ISSN 1677-7042. Disponível em: <<https://www.in.gov.br/en/web/dou/-/portaria-n-315-de-4-de-abril-de-2018-9177556>>.

COCKBURN, A. **Writing Effective Use Cases**. Addison-Wesley, 2001. (Crystal collection for software professionals). ISBN 9780201702255. Disponível em: <<https://books.google.com.br/books?id=VKJQAAAAMAAJ>>.

DIANA, M. D.; GEROSA, M. A. Nosql na web 2.0: Um estudo comparativo de bancos não-relacionais para armazenamento de dados na web 2.0. In: **Workshop de Teses e Dissertações de Bancos de Dados do Simpósio Brasileiro de Bancos de Dados WTDBD2010**. [S.l.: s.n.], 2010.

FACHINE, B. **MongoDB - Entendendo full-text search**. 2020. Acesso em 15 abr. 2023. Disponível em: <<https://www.redspark.io/mongodb-entendendo-full-text-search/>>.

FERREIRA, J.; MARTINS, E. Fluxo de exceções intraprocedimentais a partir do diagrama de atividades da uml 2.0. **Instituto de Computação Universidade Estadual de Campinas, Campinas, São Paulo**, 2010.

INOVAÇÃO - SEBRAE. **Técnicas de usabilidade para sites do setor público**. 2017. Disponível em: <<https://inovacaoebraeminas.com.br/tecnicas-de-usabilidade-para-sites-do-setor-publico/>>.

LE MOS, M. F. de; OLIVEIRA, P. C.; RUELA, L. C.; SANTOS, M. da S.; SLVEIRA, T. C.; REIS, J. C. de S. Aplicabilidade da arquitetura mvc em uma aplicação web (webapps). **RE3C-Revista Eletrônica Científica de Ciência da Computação**, v. 8, n. 1, 2013.

LÓSCIO, B. F.; OLIVEIRA, H. d.; PONTES, J. d. S. Nosql no desenvolvimento de aplicações web colaborativas. **VIII Simpósio Brasileiro de Sistemas Colaborativos**, sn, v. 10, n. 1, p. 11, 2011.

MESUT, A.; ÖZTÜRK, E. A method to improve full-text search performance of mongodb. **Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi**, v. 28, n. 5, p. 720–729, 2022.

MongoDB. **Full-Text Search: What Is It and How It Works**. s.d. <<https://www.mongodb.com/basics/full-text-search>>. Acesso em 15 abr. 2023.

NETO, J. C. F. A.; ARAÚJO, F. L. N.; NETO, M. M.; PAILLARD, G. A.; MOREIRA, L. O. Um estudo exploratório entre banco de dados nosql e armazenamento de dados em blockchain. **Revista Sistemas e Mídias Digitais (RSMD)**, v. 6, n. 1, 2021.

NIELSEN, J. **Usability 101: Introduction to usability**. 2012. Disponível em: <<https://www.nngroup.com/articles/usability-101-introduction-to-usability/>>.

NOGUEIRA, L. B. do N.; CURA, L. M. D. V.; PAULISTA, C. U. C. L. Um estudo para desenvolvimento de consultas com lógica nebulosa em bancos de dados de documentos. 2020.

SANTOS, E. Web design: uma reflexão conceptual. Universidade Aberta, 2009.

SILVA, D. P. da; BARRETO, F. F.; MENDES, J. A.; SOUZA, M. A. de; SILVA, W. F. da. Ged–gerenciamento eletrônico de documentos a tecnologia que está mudando o mundo. **INICIA**, v. 37, p. 38, 2003.

SOARES, A. S. S.; MATOS, P. F. Uma análise comparativa entre sistemas gerenciadores de bancos de dados nosql no contexto de internet das coisas. In: **SBB (Short Papers)**. [S.l.: s.n.], 2017. p. 306–311.

SOMMERVILLE, I. **Engenharia de Software, 9ª Edição**. [S.l.]: Pearson Education–BR, 2011.

SOUSA, R. R. de. Universidade federal rural do semi-árido–ufersa. 2007.

TEIXEIRA. Censo da educação superior. **Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira**, 2022. Disponível em: <https://download.inep.gov.br/educacao_superior/ceenso_superior/documentos/2021/apresentacao_censo_da_educacao_superior_2021.pdf>.

VILHEGAS, V. Utilização de arquitetura em camadas baseada no model view controller, em aplicações web. **ETIC-ENCONTRO DE INICIAÇÃO CIENTÍFICA-ISSN 21-76-8498**, v. 7, n. 7, 2011.

APÊNDICE A – DESCRIÇÃO DE TODOS OS CASOS DE USO DO SISTEMA

Quadro 2 – Caso de uso 01: Realizar pesquisa

Nome do cenário	Realizar pesquisa
Descrição	O sistema permitirá a pesquisa por termos específicos de acordo com as necessidades do usuário.
Atores envolvidos	Público e Administrador
Pré-condições	Nenhuma.
Fluxo básico	<ol style="list-style-type: none">1. O sistema possui uma tela inicial por onde são feitas as pesquisas.2. O usuário decide realizar uma pesquisa.3. O sistema solicita o valor a ser buscado.4. O usuário informa os termos desejados e clica na opção de busca.5. O sistema atribui os filtros básicos à pesquisa.6. O sistema oferece todos os documentos referentes aos termos da busca em forma de lista e o caso de uso termina.
Fluxos alternativos	Nenhum.
Fluxos de exceção	<p>E1: Nenhum termo informado</p> <ol style="list-style-type: none">1. No passo 4, o usuário não informa os termos da busca e tenta pesquisar.2. O sistema informa ao usuário sobre a ausência dos termos da busca.3. O caso de uso retorna ao passo 3.
Pós-condições	O usuário visualiza a lista de documentos e pode abrir qualquer um, como desejar.

Fonte: Elaborado pelo autor (2023)

Quadro 3 – Caso de uso 02: Filtrar resultados

Nome do cenário	Filtrar resultados
Descrição	O usuário terá a opção de filtrar os resultados que serão exibidos de acordo com a disponibilidade no sistema.
Atores envolvidos	Público e Administrador
Pré-condições	O usuário deve ter realizado uma pesquisa.
Fluxo básico	<ol style="list-style-type: none"> 1. O sistema possui opções de filtragem distintas disponíveis. 2. O usuário determina quais filtros deseja aplicar na pesquisa. 3. O sistema aplica os filtros nos resultados e os atualiza, a fim de exibir apenas os correspondentes, e o caso de uso termina.
Fluxos alternativos	<p>A1: Filtragem temporal</p> <ol style="list-style-type: none"> 1. No passo 2, o usuário seleciona a opção de filtro temporal. 2. O usuário seleciona qual período de tempo se adequa mais à sua necessidade. 3. O caso de uso retorna ao passo 3. <p>A2: Filtragem por tipo de documento</p> <ol style="list-style-type: none"> 1. No passo 2, o usuário seleciona a opção de filtro por tipo de documento. 2. O usuário seleciona qual o tipo de documento se adequa mais à sua necessidade. 3. O caso de uso retorna ao passo 3. <p>A3: Filtragem por direcionamento de área</p> <ol style="list-style-type: none"> 1. No passo 2, o usuário seleciona a opção de filtragem de direcionamento por área. 2. O usuário seleciona para qual área em específico deseja que o documento seja direcionado. 3. O caso de uso retorna ao passo 3.
Fluxos de exceção	<p>E1: Nenhum resultado correspondente</p> <ol style="list-style-type: none"> 1. No passo 3, o sistema verifica se não há nenhum documento que corresponda ao filtro aplicado. 2. O sistema informa ao usuário que nenhum documento foi encontrado e o caso de uso termina.
Pós-condições	O usuário visualiza a lista de documentos e pode abrir qualquer um, como desejar.

Fonte: Elaborado pelo autor (2023)

Quadro 4 – Caso de uso 03: Visualizar documentos recentes

Nome do cenário	Visualizar documentos recentes
Descrição	O usuário terá acesso aos documentos por ordem de adição na página inicial.
Atores envolvidos	Público e Administrador
Pré-condições	Nenhuma.
Fluxo básico	<ol style="list-style-type: none"> 1. O sistema possui uma ferramenta que lista os documentos na tela inicial da aplicação. 2. O usuário acessa a tela inicial da aplicação e o caso de uso termina.
Fluxos alternativos	Nenhum.
Fluxos de exceção	Nenhum.
Pós-condições	O usuário visualiza a lista de documentos disponível no sistema.

Fonte: Elaborado pelo autor (2023)

Quadro 5 – Caso de uso 04: Visualizar informações dos documentos

Nome do cenário	Visualizar informações dos documentos
Descrição	O usuário, antes de visualizar o corpo do documento, terá uma breve noção do seu conteúdo através de breves informações, por exemplo, uma breve descrição sobre o que se trata o documento e sua data de publicação.
Atores envolvidos	Público e Administrador
Pré-condições	Nenhuma.
Fluxo básico	1. O sistema possui uma ferramenta que lista os documentos na tela inicial da aplicação. 2. O usuário visualiza as informações dispostas no cabeçalho do documento e o caso de uso termina.
Fluxos alternativos	Nenhum.
Fluxos de exceção	Nenhum.
Pós-condições	O usuário tem um vislumbre do conteúdo do documento sem que seja preciso abri-lo.

Fonte: Elaborado pelo autor (2023)

Quadro 6 – Caso de uso 05: Abrir documento

Nome do cenário	Abrir documento
Descrição	Ao encontrar o documento desejado em meio ao acervo digital, o usuário poderá abri-lo para melhorar a visualização do seu conteúdo.
Atores envolvidos	Público e Administrador
Pré-condições	Nenhuma.
Fluxo básico	1. O sistema possui uma ferramenta que possibilita que o usuário acesse o corpo de um documento. 2. O usuário decide abrir um documento. 3. O sistema direciona o usuário ao documento e o caso de uso acaba.
Fluxos alternativos	Nenhum.
Fluxos de exceção	Nenhum.
Pós-condições	O usuário visualiza o corpo do documento.

Fonte: Elaborado pelo autor (2023)

Quadro 7 – Caso de uso 06: Fazer Login

Nome do cenário	Fazer Login
Descrição	O administrador deverá ser autenticado pelo sistema para conseguir acessar e manipular dados sensíveis.
Atores envolvidos	Administrador
Pré-condições	O usuário deve possuir um registro no sistema.
Fluxo básico	<ol style="list-style-type: none"> 1. O sistema possui uma ferramenta que possibilita o usuário entrar na sua conta. 2. O usuário decide entrar na sua conta. 3. O usuário opta pela opção de realizar login e insere todos os dados necessários. 4. O sistema valida os dados informados, permite o acesso ao usuário e o caso de uso termina.
Fluxos alternativos	Nenhum.
Fluxos de exceção	<p>E1: Conta não existente</p> <ol style="list-style-type: none"> 1. No passo 3, o usuário insere dados que não estão cadastrados no sistema. 2. O sistema informa sobre o erro, impede o acesso do usuário e o caso de uso termina.
Pós-condições	O usuário tem acesso ao sistema e a todas as funcionalidades relacionadas ao seu perfil.

Fonte: Elaborado pelo autor (2023)

Quadro 8 – Caso de uso 07: Adicionar documentos

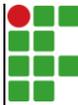
Nome do cenário	Adicionar documentos
Descrição	O administrador terá acesso a função de adicionar documentos para manter o sistema sempre atualizado, em relação ao acervo da Instituição.
Atores envolvidos	Administrador
Pré-condições	O usuário administrador deve estar autenticado no sistema.
Fluxo básico	<ol style="list-style-type: none"> 1. O sistema possui uma ferramenta que possibilita a adição de documentos. 2. O usuário decide adicionar um novo documento. 3. O sistema solicita o novo documento e seus respectivos dados em forma de formulário. 4. O usuário adiciona o documento, informa os dados e o caso de uso termina.
Fluxos alternativos	Nenhum.
Fluxos de exceção	Nenhum.
Pós-condições	O novo documento fica disponível para o público.

Fonte: Elaborado pelo autor (2023)

Quadro 9 – Caso de uso 08: Manter documentos

Nome do cenário	Manter documentos
Descrição	O administrador terá total liberdade para gerenciar e manipular os documentos existentes (atualizar e excluir dados) para manter o sistema em ordem de acordo com o acervo da Instituição.
Atores envolvidos	Administrador
Pré-condições	O usuário administrador deve estar autenticado no sistema
Fluxo básico	<ol style="list-style-type: none"> 1. O sistema possui uma ferramenta que possibilita o manuseio dos documentos adicionados. 2. O usuário decide gerenciar os documentos disponíveis no sistema. 3. O sistema exibe as opções de gerenciamento de cada documento. 4. O usuário decide qual opção atende a sua necessidade. 5. O sistema exige uma verificação para confirmar a alteração da lista de documentos. 6. O usuário confirma a alteração e o caso de uso termina.
Fluxos alternativos	<p>A1: Editar documento</p> <ol style="list-style-type: none"> 1. No passo 3, o usuário opta por editar as informações de um documento em específico. 2. O sistema solicita as novas informações ao usuário. 3. O usuário preenche os dados solicitados e realiza a confirmação para o sistema. 4. O sistema salva o documento em seu novo estado e atualiza a lista de documentos do sistema. 5. O caso de uso retorna ao passo 4. <p>A2: Excluir documento</p> <ol style="list-style-type: none"> 1. No passo 3, o usuário opta por excluir um documento em específico. 2. O sistema solicita as confirmações necessárias para realizar a exclusão. 3. O usuário realiza a confirmação para o sistema. 4. O sistema exclui o documento e atualiza a lista de documentos do sistema. 5. O caso de uso retorna ao passo 4.
Fluxos de exceção	Nenhum.
Pós-condições	O sistema mantém a atualização na lista de documentos disponível no sistema.

Fonte: Elaborado pelo autor (2023)

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus Cajazeiras - Código INEP: 25008978
	Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB)
	CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100

Documento Digitalizado Ostensivo (Público)

TCC

Assunto:	TCC
Assinado por:	Lucas Pereira
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Lucas Tavares do Nascimento Pereira, ALUNO (202012010018) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 08/03/2024 10:55:06.

Este documento foi armazenado no SUAP em 08/03/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1108322

Código de Autenticação: bc007f859f

