

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

**QUAL A BOA, DEV? UMA FERRAMENTA PARA AUXILIAR NA
DISSEMINAÇÃO DE CONHECIMENTO VOLTADO PARA O ESTUDO
COMPUTACIONAL**

DÉBORA CAMILLY QUIRINO DA SILVA

**Cajazeiras
2024**

DÉBORA CAMILLY QUIRINO DA SILVA

**QUAL A BOA, DEV? UMA FERRAMENTA PARA AUXILIAR NA DISSEMINAÇÃO
DE CONHECIMENTO VOLTADO PARA O ESTUDO COMPUTACIONAL**

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

S586q Silva, Débora Camilly Quirino da.
Qual a boa, dev? uma ferramenta para auxiliar na disseminação de conhecimento voltado para o estudo computacional / Débora Camilly Quirino da Silva.– 2024.

45f. : 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. Fórum virtual. 3. Estudo computacional. 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

DÉBORA CAMILLY QUIRINO DA SILVA

**QUAL A BOA, DEV? UMA FERRAMENTA PARA AUXILIAR NA DISSEMINAÇÃO DE
CONHECIMENTO VOLTADO PARA O ESTUDO COMPUTACIONAL**

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: **15 de Agosto de 2024.**

Prof. Me. Francisco Paulo de Freitas Neto - Orientador

Prof. Dr. Hudson Geovane de Medeiros - Avaliador

IFPB - Campus Cajazeiras

Prof. Tecgo. Antonio Ricart Jacinto de Oliveira Medeiro

IFPB - Campus Cajazeiras

Documento assinado eletronicamente por:

- **Francisco Paulo de Freitas Neto, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 15/08/2024 15:37:14.
- **Hudson Geovane de Medeiros, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 15/08/2024 15:45:11.
- **Antonio Ricart Jacinto de Oliveira Medeiros, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO**, em 18/08/2024 21:36:52.

Este documento foi emitido pelo SUAP em 15/08/2024. 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 592749
Verificador: fa6c887a21
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

Agradeço a Deus, por ter me fortalecido e me mantido perseverante ao longo dessa jornada. "Por isso não tema, pois estou com você; não tenha medo, pois sou o seu Deus. Eu o fortalecerei e o ajudarei; eu o segurarei com a minha mão direita vitoriosa."(Isaias 41:10)

Agradeço a minha família por sempre me apoiar, acreditar no meu potencial, me incentivar a persistir e correr atrás dos meus sonhos.

Agradeço ao professor Paulo Freitas por toda sua contribuição e suporte ao longo do processo transmitindo seu conhecimento com excelência.

RESUMO

Diante do avanço tecnológico crescente, faz-se necessário mudar a visão estratégica dos diversos âmbitos sociais, principalmente o educacional, para que possibilite a ocorrência de fenômenos como reparações históricas e democratização, contribuindo com a construção de um futuro mais inclusivo. Além disso, contribuir para que passe a ser mais estimulado o estudo de áreas vistas socialmente como complexas não pelo seu caráter em si mas porque os recursos para seu aprendizado são custosos, o que os torna limitados a apenas uma parcela da população. Nesse contexto, este trabalho tem o intuito de promover a disseminação de conhecimento voltado a tecnologia, através da criação de uma aplicação, que apresenta o caráter de um fórum online, no qual é possível a discussão e compartilhamento de conteúdos relacionados a essa temática. Além disso, oferece um ambiente virtual intuitivo, o qual facilita o acesso daqueles não tão inseridos no contexto, permite aos usuários trocar informações e também obtê-las de uma maneira mais otimizada.

Palavras-chave: Conhecimento. Fórum. Tecnologia. Democratização. Estudo.

ABSTRACT

Faced with increasing technological advancement, it is necessary to change the strategic vision of different social spheres, mainly educational, to enable occurrence of phenomena such as historical reparations and democratization, contributing with the construction of a more inclusive future. Furthermore, contributing to the be more encouraged to study areas seen socially as complex not because of its character in itself but because the resources for its learning are expensive, which makes it limited to just a portion of the population. In this context, this work has with the aim of promoting the dissemination of knowledge focused on technology, through of creating an application, which has the character of an online forum, in which discussion and sharing of content related to this topic is possible. Additionally, it offers an intuitive virtual environment, which facilitates access for those who are not so inserted in the context, allows users to exchange information and also obtain it in a more optimized way.

Keywords: Knowledge. Forum. Technology. Democratization. Study.

LISTA DE FIGURAS

Figura 1 – Ilustração do <i>Stack Overflow</i>	13
Figura 2 – Ilustração do <i>Code Project</i>	14
Figura 3 – Ilustração do <i>CodeRanch</i>	15
Figura 4 – Exemplo de documento do MongoDB	20
Figura 5 – Exemplo de documento do MongoDB no formato BSON	21
Figura 6 – Diagrama de casos de uso	27
Figura 7 – Protótipo da página inicial do fórum	28
Figura 8 – Arquitetura geral do sistema	29
Figura 9 – Tela de cadastro do sistema	33
Figura 10 – Tela de login do sistema	34
Figura 11 – Tela inicial da aplicação	34
Figura 12 – Tela de busca e exibição de comentários	35
Figura 13 – Tela de criação de postagens	35

LISTA DE QUADROS

Quadro 1 – Requisitos da aplicação Qual a boa,Dev?	26
Quadro 2 – Caso de uso: Fazer cadastro	27
Quadro 3 – Caso de uso: Fazer login	39
Quadro 4 – Caso de uso: Adicionar comentário	39
Quadro 5 – Caso de uso: Editar comentário	40
Quadro 6 – Caso de uso: Apagar comentário	40
Quadro 7 – Caso de uso: Adicionar um tópico	41
Quadro 8 – Caso de uso: Editar tópico	41
Quadro 9 – Caso de uso: Apagar tópico	42
Quadro 10 – Caso de uso: Buscar por conteúdo	42
Quadro 11 – Caso de uso: Ver comentários	43

LISTA DE ABREVIATURAS E SIGLAS

ACID	Atomicidade, Consistência, Isolamento e Durabilidade
API	<i>Application Programming Interface</i>
BASE	Basicamente disponível, Estado leve e Consistência Eventual
BSON	<i>Binary Json</i>
CouchDB	<i>Couch Database</i>
CRUD	<i>Create, Read, Update and Delete</i>
CSS	<i>Cascading Style Sheets</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JSON	<i>JavaScript Object Notation</i>
MongoDB	<i>Mongo Database</i>
NoSql	<i>Not Only Sql</i>
SGBDs	Sistema Gerenciador de Banco de Dados
TCC	Trabalho de Conclusão do Curso
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	11
1.1	PROBLEMÁTICA	11
1.2	DESCRIÇÃO DA SOLUÇÃO	12
1.2.1	Ferramentas relacionadas	12
1.3	OBJETIVOS	15
1.3.1	Objetivo Geral	15
1.3.2	Objetivos Específicos	15
1.4	ATIVIDADES	16
1.5	ORGANIZAÇÃO DO DOCUMENTO	16
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	API REST	18
2.2	BANCOS DE DADOS Não-Relacionais	19
2.2.1	BANCOS DE DADOS ORIENTADOS À DOCUMENTOS	19
2.2.1.1	MONGO DB	20
2.3	FULL TEXT-SEARCH	21
2.3.1	ÍNDICES	22
2.3.2	Índices textuais	23
2.4	TRABALHOS RELACIONADOS	24
3	QUAL A BOA, DEV?	25
3.1	ANÁLISE	25
3.1.1	Elicitação de requisitos	25
3.1.2	Diagrama de casos de uso	26
3.1.3	Protótipos	28
3.2	PROJETO	28
3.2.1	Arquitetura	29

3.3	TECNOLOGIAS UTILIZADAS	30
3.3.1	Javascript	30
3.3.2	Node.js	30
3.3.3	Express	31
3.3.4	MongoDB	31
3.3.5	Local Storage	32
3.4	TELAS DA APLICAÇÃO	32
4	CONSIDERAÇÕES FINAIS	36
	REFERÊNCIAS	37
	APÊNDICE A – DETALHAMENTO DOS CASOS DE USO	39

1 INTRODUÇÃO

Mesmo com os grandes avanços na expansão do sistema educacional do Brasil, não ocorreu necessariamente a superação das desigualdades entre grupos sociais. Ainda existem grandes desafios no combate aos preconceitos e outras barreiras para a garantia do direito à educação, para que se alcance acesso, trajetória regular e aprendizado adequado para todos e todas com equidade.(MACANA; BOLELLI, 2023) Conseqüentemente, o conhecimento tecnológico já foi extremamente restrito, tendo em vista que o acesso por parte da população em geral era limitado e os recursos para tal eram de custo muito elevado em comparação com os dias atuais.

A exclusão digital é o termo utilizado para sintetizar todo um contexto que impede a maior parte das pessoas de participar dos benefícios das novas tecnologias de informação. Já a inclusão digital consiste no processo de democratização do acesso às novas tecnologias e melhores condições de vida a todos os cidadãos, possibilitando a estes se inserirem na sociedade informacional(GROSSI et al., 2013).

Com o objetivo declarado de levar informatização à população em geral, os programas de inclusão lançam mão das mais variadas estratégias de disseminação das máquinas e dos conhecimentos necessários à plena integração dos menos afortunados à sociedade tecnológica. Acesso coletivo, subsídio para a compra privada, cursos profissionalizantes, softwares de baixo custo e toda uma parafernália institucional e tecnológica são meios utilizados para tentar fechar esse abismo e permitir que o computador avance por todo o tecido social.(CAZELOTO, 2019)

Diante do pressuposto, este Trabalho de Conclusão de Curso (TCC) apresenta um sistema que visa contribuir com a democratização do aprendizado tecnológico, através de um ambiente de interação, um fórum online, que possibilita a busca e troca de informações, com o objetivo de disseminar e facilitar a obtenção de conhecimento da área. Nele, os usuários podem discutir e compartilhar conteúdos fazendo postagens e interagindo por comentários.

1.1 PROBLEMÁTICA

De acordo com (BAGGIO, 2000), na era da informação atual é necessário que as ferramentas tecnológicas sejam democratizadas, não somente por estarem sendo altamente requisitadas no mercado de trabalho mas também para que não se tornem uma contribuição para a exclusão social.

Conforme (BATISTA, 2010), a quantidade de informação gerada excessivamente sem realização de nenhum tipo de filtro, fez surgir na sociedade um descontrole relacionado à disseminação e absorção destas informações, fazendo com que fenômenos como a síndrome de fadiga da informação ocorressem, caracterizada pelo sentimento de abandono causado pela sobrecarga de informação na população.

Com o passar dos anos, houve uma crescente voltada para a conscientização da importância de disseminar conhecimento, porém, essa iniciativa não foi analisada do ponto de vista de como essas informações seriam transmitidas corretamente e como poderia ocorrer a facilitação do seu acesso, acarretando em obstáculos para a obtenção de conhecimento associado a tecnologia, tendo em vista, que majoritariamente é possível somente através de sites ou documentação oficial e o distanciamento populacional cada vez maior de áreas já vistas anteriormente como complexas, a exemplo do estudo computacional.

1.2 DESCRIÇÃO DA SOLUÇÃO

A proposta de solução para essa problemática foi o desenvolvimento de uma aplicação web que auxiliasse no processo de troca de informações entre usuários da internet.

A expectativa era que essa aplicação possibilitasse que seus usuários se cadastrassem e pudessem visualizar conteúdos do âmbito tecnológico, como também pudessem interagir e compartilhar com outros, ao comentar em postagens e inserir as próprias dentro do sistema. Além disso, a aplicação conta com uma funcionalidade de busca, na qual, permite que o usuário somente visualize os materiais ligados ao tema em específico pesquisado por ele.

Com o objetivo de contribuir com os estudos ligados à tecnologia, software, programação, partindo do princípio de acelerar o processo de aprendizagem ao concentrar um grande número de informações em um único lugar, como também ajudando no cotidiano daqueles já atuantes na área.

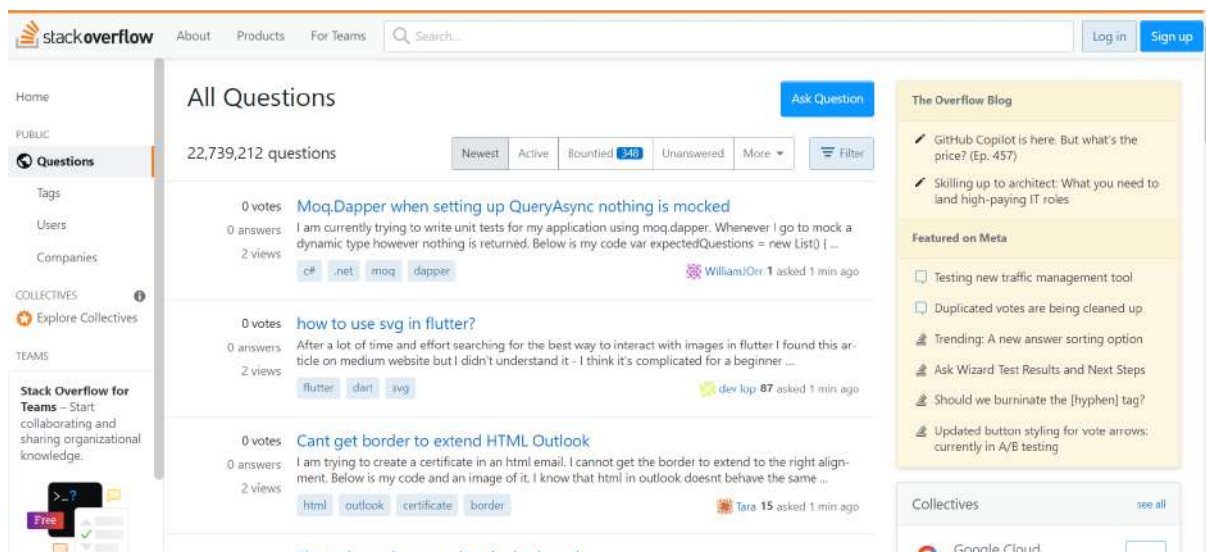
1.2.1 Ferramentas relacionadas

Nesta seção, são apresentadas ferramentas que operam de forma análoga, visando contribuir para a mesma temática abordada neste trabalho. Por meio de buscas na internet, foram escolhidas as ferramentas *Stack Overflow*, *Code Project* e *Code Ranch*.

A aplicação desenvolvida por Jeff Atwood e Joel Spolsky em 2008, intitulada

como *Stack Overflow*, é um sistema web direcionado ao campo da programação de computadores. Seu propósito é fomentar um ambiente propício para que os usuários possam tanto questionar quanto responder sobre temas, oferecendo recursos adicionais e planos de assinatura mensal voltados para empresas. Além disso, o *Stack Overflow*¹ mantém associações com renomadas plataformas, como o *Microsoft Teams*. A Figura 1 apresenta a tela inicial da ferramenta.

Figura 1 – Ilustração do *Stack Overflow*



Fonte: Wikipedia, 2022

Com isso, utilizando-se de tais estratégias, a plataforma busca promover uma comunidade *online* para desenvolvedores e interessados em tecnologia. Em contrapartida, a proposta do "Qual a boa, Dev?" contou com esses ideais, porém, não foi somente centrada nos atuantes da área em si, embora este também fosse o público-alvo, mas seu foco era as pessoas que tivessem somente o mínimo ou até mesmo nenhum domínio de informática e quisessem aprender sobre de maneira rápida. Por isso, apresenta uma interface extremamente simplificada e a forma como foram dispostos os elementos na tela foi visando a facilitação para esses usuários.

Outra aplicação é o *Code Project*², que tem sua interface apresentada na Figura 2, trata-se de um sistema web criado com o intuito de ser uma comunidade para programadores, permitindo o compartilhamento de artigos e postagens de perguntas e soluções.

¹ <https://stackoverflow.com/>

² <https://www.codeproject.com/>

Figura 2 – Ilustração do *Code Project*

The screenshot displays the Code Project website interface. At the top, it shows the site's logo, member count (15,792,842), and a sign-in option. A navigation menu includes links for home, articles, quick answers, discussions, features, community, and help, accompanied by a search bar. Below the navigation are three prominent buttons: 'POST AN ARTICLE', 'ASK A QUESTION', and 'SEARCH OUR ARTICLES'. The main content area features a large banner for 'CODEPROJECT AI' with a woman's face and text about adding AI to applications. To the right, there is a 'Current Viewers' section showing 74,389 active user sessions and a 'Questions' section with a list of recent questions.

Fonte: Code Project, 2023

Essa ferramenta diferentemente da apresentada neste documento, além de possibilitar interações entre os usuários, também oferece um sistema de inteligência artificial que pode ser usado para detecção de objetos, manipulação de imagens e reconhecimento facial.

Outra plataforma semelhante é a *CodeRanch*³, que apresenta o caráter de um fórum para aprender e discutir sobre programação. A Figura 3 exibe a tela inicial desta ferramenta.

³ <https://coderanch.com/>

Figura 3 – Ilustração do CodeRanch



Fonte: CodeRanch, 2023

Em contraste com este trabalho, o *CodeRanch*, não tem foco somente em tecnologia, trazendo a abordagem de conteúdos voltados para resenhas de livros, certificações de exames e discussões de planos de carreira.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Este Trabalho de Conclusão de Curso tem como objetivo contribuir com a disseminação de conteúdos relacionados ao estudo de computação e desenvolvimento de software, por meio do desenvolvimento de um fórum online, simplificando tal acesso ao visar a concentração e possibilidade de obtenção de muitas informações em um único lugar.

1.3.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Disponibilizar um sistema de fácil utilização com foco no público que também tenha acesso limitado a recursos computacionais.
- Permitir a consulta de conteúdos com base no interesse do usuário;
- Criar uma interface que seja intuitiva e projetada para otimizar o tempo do usuário;

- Prover opções de interação para que haja troca de informações de forma dinâmica.

1.4 ATIVIDADES

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

- **Elaboração da ideia (A1):** Nessa etapa, foi escolhido o tema do projeto e como ele seria abordado.
- **Fundamentação Teórica (A2):** Nesse ponto, foram realizadas pesquisas sobre trabalhos semelhantes e a respeito do tema em si.
- **Seleção das Tecnologias (A3):** Nessa etapa, foram selecionadas quais tecnologias a serem utilizadas para o desenvolvimento do sistema.
- **Análise e Projeto de Sistema (A4):** Nesse momento, foram levantados os requisitos do sistema, também houve a produção dos protótipos, juntamente com os casos de uso e construção da arquitetura do sistema.
- **Elaboração do Documento (A5):** Escrita de documento de TCC.
- **Atualização do Projeto de Sistema (A6):** Conforme houve o andamento deste trabalho, foram feitos refinamentos na sua estrutura.
- **Implementação da Aplicação (A7):** Após a finalização deste trabalho, foi iniciado o processo de implementação da ferramenta proposta.
- **Atualização do documento (A8):** Considerando-se a progressão da elaboração do sistema, foram feitas atualizações no documento para garantia de que todo o processo juntamente com possíveis mudanças que viessem a surgir fossem registradas.
- **Registro de Software (A9):** Após a finalização de todo o projeto, foi realizado o registro da aplicação.

1.5 ORGANIZAÇÃO DO DOCUMENTO

Nas seções seguintes deste documento serão apresentados a fundamentação teórica, que será responsável por demonstrar quais os embasamentos teóricos que foram empregados para concepção do sistema desenvolvido. Posteriormente, será explanado como ocorreu o processo inicial de implementação, levando-se em conta,

a elicitação de requisitos, casos de uso, prévia da interface a ser apresentada para o usuário dentro da plataforma e também as tecnologias utilizadas, ademais havendo exposição da arquitetura e das telas do sistema. Ao final, são abordadas as considerações finais diante de todas as proposições feitas até o momento.

2 FUNDAMENTAÇÃO TEÓRICA

Neste tópico serão abordados os conceitos que foram utilizados como base para o desenvolvimento do sistema. Primeiro, será discutido sobre *API REST*, depois os bancos não relacionais e orientados à documentos, em seguida, sobre o banco em específico que foi utilizado, o *MongoDB*, um dos principais recursos oferecidos por ele, a busca textual completa, baseada na indexação de texto e por último trabalhos relacionados a temática deste projeto.

2.1 API REST

Conforme (LIMA, 2020) *REST* (Representational State Transfer) é um modelo de arquitetura que fornece diretrizes para que os sistemas distribuídos se comuniquem diretamente usando os princípios e protocolos existentes da *web*.

Através de solicitações *HTTP* (Hypertext Transfer Protocol) é possível executar funções padrão de banco de dados, como leitura, criação, atualização e remoção de dados. Podem ser desenvolvidas usando praticamente qualquer linguagem de programação e oferecem suporte a uma variedade de formatos de dados.

O único requisito é que devem alinhar aos seis princípios de *design* de *REST*: Interface Uniforme (os recursos precisam apresentar informações iguais para todas as requisições, não importa a origem dela), Desacoplamento do Cliente-Servidor (cliente e servidor devem ser independentes entre si), Sem estado definido (o servidor não tem permissão para armazenar dados de solicitação de cliente), Capacidade de Armazenamento em Cache (os recursos devem ser armazenados pelo cliente ou servidor para melhora de desempenho do cliente), Arquitetura de sistema em camadas (cliente e servidor não interagem diretamente, entre eles há intermediários) e Código sob demanda (conforme necessidade novos comandos podem ser usados pelo cliente) (PALHANO, 2024).

Além dessas condições anteriormente citadas, condições herdadas do *HTTP* devem ser usadas no estilo arquitetural *REST*, como os métodos e códigos de retorno próprios do protocolo *HTTP* (SILVA, 2019).

2.2 BANCOS DE DADOS NÃO-RELACIONAIS

Os bancos de dados *NoSQL* (Não apenas SQL) geralmente seguem os princípios do paradigma BASE (Basicamente disponível, Estado leve e Consistência Eventual), em detrimento das propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade), priorizando o desempenho e a disponibilidade. Esse paradigma fornece aos bancos de dados características como: Basicamente disponível, ou seja, o sistema parece estar funcionando o tempo todo, Estado leve, o sistema não precisa ser consistente o tempo todo e Consistência eventual, ou seja, o sistema torna-se consistente no momento devido.(FRIESS, 2013)

Ainda de acordo com (FRIESS, 2013) a partir desse tipo, pode-se gerenciar grande quantidade de dados sem perda de desempenho. Isso porque esses bancos prezam pela inexistência de transações e cláusulas *join*. Também possuem configuração facilitada para escalabilidade horizontal, suporte nativo à replicação dos dados e ausência parcial ou total de esquema que define estrutura de dados, garantindo assim alta disponibilidade.

Segundo (KOKAY, 2012), não utilizam o modelo de tabelas e relacionamentos e apresentam uma variedade de categorias, orientado a colunas, orientado a grafos, chave-valor e orientado a documentos.

Em contraste com os bancos relacionais que apresentam a escalabilidade como um ponto fraco, à medida que os dados aumentam, fazendo-se necessário investir em *hardware* caro para manter o desempenho. Isso pode ser um desafio em aplicativos de grande escala, onde a escalabilidade horizontal é preferível, também não possuem tanta flexibilidade de escolha, tendo um modelo fixo.(JUNIOR, 2023)

2.2.1 BANCOS DE DADOS ORIENTADOS À DOCUMENTOS

De acordo com (OLIVEIRA, 2014), bancos de dados orientados a documentos são baseados no armazenamento de pares de chave-valor, tendo um esquema altamente flexível. Esta característica torna os bancos de dados orientados à documentos ótimas opções para dados semi-estruturados, como os utilizados em ferramentas *web* colaborativas.

No modelo orientado a documentos há um conjunto de documentos e em cada documento um conjunto de campos (chaves) e o valor deste campo. Outra característica importante é que este modelo não depende de um esquema rígido, ou seja, não exige uma estrutura fixa como ocorre nos bancos relacionais. Assim, é possível que ocorra uma atualização na estrutura do documento, com a adição de novos campos, por

exemplo, sem causar problemas ao banco de dados. Como principais soluções que adotam o modelo orientado a documentos, se destacam o *CouchDB*¹ e o *MongoDB*², no último os documentos possuem a seguinte estrutura mostrada na Figura 4.

Figura 4 – Exemplo de documento do MongoDB

```
var pessoa = {
  "_id": ObjectId("64b4f1e2d52f12a3b04b93f6"),
  "nome": "Ana Maria",
  "idade": 28,
  "email": "maria.pessoa@example.com",
  "cidade": "João Pessoa",
  "estado": "PB"
}
```

Fonte: Elaborado pela autora (2023)

2.2.1.1 MONGO DB

Segundo (WANZELLER, 2013), o *MongoDB* é um banco de dados *NoSQL*, de código aberto, orientado a documentos, escrito em C++. Os dados são persistidos em coleções de dados que são representados usando o *BSON*, um formato binário similar ao *JSON*. O banco tem suporte a todos os tipos de dados JSON como *string*, inteiro, booleano, *double*, *array* e objeto.

Em vez de utilizar tabelas e linhas, ele guarda as bases num único documento, que pode conter todas as informações necessárias. Sendo assim, ele possibilita o acesso de forma eficiente, independentemente do esquema ou da estrutura utilizada (COUTINHO, 2023). A Figura 5 exibe como ocorre este processo utilizando o formato *BSON*.

¹ <https://couchdb.apache.org/>

² <https://www.mongodb.com/>

Figura 5 – Exemplo de documento do MongoDB no formato BSON

```
{
  "nome": "Débora Silva",
  "idade": 20,
  "email": "debora.silva@example.com",
  "groups": ["news", "sports"]
}
```

Fonte: Elaborado pela autora (2023)

Este banco não trabalha com uma estrutura de dados bem definida, ele usa esquemas dinâmicos. Com ele é possível criar coleções sem que a estrutura, campos ou tipos de valores dos documentos estejam definidos. Essa forma flexível de armazenar os dados permite trabalhar com estruturas.(WANZELLER, 2013)

De maneira geral, como outros bancos de dados *NoSQL*, ele oferece mais escalabilidade, rapidez no desempenho e flexibilidade. Os dados podem ser dispostos de maneiras diversas, ocupando menos espaço e tornando a informação mais acessível sempre que necessário.(AWARI, 2022)

Além disso, apresenta como benefícios o processo de compartilhamento de balanceamento de carga onde distribui conjuntos de dados de grande porte para diversas máquinas virtuais ao mesmo tempo, mantendo níveis aceitáveis de leitura e gravação. Este ajuste de escala horizontal é chamado de fragmentação e ajuda organizações a evitar os custos de escala vertical de *hardware*, sem abrir mão da expansão de implementações baseadas em nuvem, usar uma linguagem de consulta semelhante aos bancos de dados *SQL* sendo extremamente acessível tanto para desenvolvedores iniciantes quanto avançados e suporte a diversas linguagens de programação como *Python*, *PHP* e *Ruby*.(IBM, 2021)

2.3 FULL TEXT-SEARCH

De acordo com (REIS; COSTA, 2022), *Full Text Search*, ou simplesmente *FTS*, é uma ferramenta de pesquisa baseada em documentos indexados, é bastante abrangente, permitindo uma ampla variedade de configurações e mecanismos.

Ela difere das pesquisas tradicionais com *like* ou outros operadores que buscam os dados realizando um escaneamento de cada caractere, cada coluna, cada tabela.

No *FTS* os dados são agrupados em documentos, que podem conter informações de diferentes colunas e tabelas.

Como afirma (VITOR, 2010), para o melhor entendimento do processo de criação, utilização e manutenção da estrutura de índices *full-text*, é preciso também conhecer o significado de alguns conceitos importantes. São eles: Termo: a palavra, frase ou caractere utilizado na pesquisa textual, Catálogo *Full-Text*, um grupo de índices *full-text*, *Word breaker*: o processo que encontra os limitadores de cada palavra em uma frase, baseando-se nas regras gramaticais do idioma selecionado para a criação do índice, *Token*: uma palavra, frase ou caractere definido pelo processo *word breaker*, *Stemmer*: o processo que gera diferentes formas verbais para as palavras, baseando-se nas regras gramaticais do idioma selecionado para a criação do índice e Filtro: componente responsável por extrair informação textual de documentos armazenados com o tipo de dados *varbinary(max)* e enviar essas informações para o processo *word breaker*.

Considerando-se um contexto de sistema de *e-commerce* para busca de produtos, com uma pesquisa normal, as buscas aconteceriam com base em correspondências exatas de texto usando operadores de comparação ou regex, já com a *Full Text Search*, seriam considerados também sinônimos, variações de palavras, relevância e suportaria diferentes idiomas, o que torna o processo mais otimizado.

2.3.1 ÍNDICES

Índices são estruturas de dados que funcionam como um índice de um livro, ajudando a localizar rapidamente informações específicas em um grande conjunto de dados. (J, 2023)

São fundamentais em bancos de dados pois agem como um catálogo de busca rápida sobre os registros de uma tabela, facilitando a localização de consultas e melhorando a performance da aplicação. (REIS; COSTA, 2022)

Uma técnica eficiente é catalogar as informações do banco de dados em chaves. Desta forma, a busca é efetuada pelas chaves, para recuperar os dados correspondentes em disco. O resultado é um único acesso ao disco rígido. Essa técnica é chamada de *index*. Um *index* em banco de dados funciona como índice de um livro. O sistema pesquisa no *index* e traz o bloco de disco contendo as informações. Os *index* são estruturas ágeis, pois são armazenados em memória principal *RAM* (*Random Access Memory*), tornando a pesquisa rápida e eficiente. Os *indexes* no *MongoDB*³ são

³ <https://www.mongodb.com/basics/full-text-search>

armazenados utilizando a estrutura de dados Árvore B. Essa técnica foi desenvolvida em 1971, utilizando o conceito de estrutura em árvore.(SCHUERZOSKI, 2019)

Ainda conforme (SCHUERZOSKI, 2019), é amplamente aplicada em sistemas de banco de dados, sendo projetada inicialmente para ser manipulada em disco rígido. Porém, atualmente utiliza-se também em memória *RAM*. A principal vantagem da utilização de estruturas Árvore B é manter as chaves classificadas em ordem crescente ou decrescente. Essa classificação torna a consulta mais ágil, por isso é a estrutura mais utilizada para melhoria de desempenho em consultas.

No *MongoDB* os *indexs* são definidos na coleção, sendo suportados em qualquer campo ou subcampo. Quando se cria uma coleção é gerado automaticamente um campo *id* para cada novo documento. Tal índice *id* identifica os documentos como exclusivo, não permitindo duplicidade. O campo *id* se assemelha ao conceito de chave primária dos SGBDRs.

2.3.2 Índices textuais

Os índices desempenham um papel crucial na otimização do desempenho das consultas em bancos de dados. Estas são algumas razões pelas quais os índices são importantes: Melhor desempenho das consultas, os índices permitem que o banco de dados localize rapidamente os registros que correspondem aos critérios de busca, resultando em consultas mais rápidas e eficientes, Redução da carga de trabalho do servidor, com índices adequados, o banco de dados pode evitar a necessidade de ler todas as linhas de uma tabela para executar uma consulta, o que reduz a carga de trabalho do servidor e melhora o tempo de resposta, Suporte a restrições de chave única, os índices podem ser usados para garantir a unicidade dos valores em uma coluna ou conjunto de colunas, permitindo a definição de restrições de chave única, Ordenação eficiente, os índices permitem que os dados sejam armazenados de forma ordenada, facilitando a execução de consultas com base em ordem crescente ou decrescente e Melhor desempenho de junções, os índices também podem melhorar o desempenho de operações de junção entre tabelas, ajudando o banco de dados a encontrar rapidamente os registros relacionados (CETESC, 2022).

A arquitetura de índice contemplada dentro do *SQL Server* (Servidor SQL) 2000 por exemplo, compreende-se em torno de índices e pilhas de dados. Existem três tipos de índices: Índices de agrupamento ou ordenados, os dados são armazenados em uma página de dados, em ordem crescente. A ordem dos valores nas páginas de índice também é crescente; Índice sem agrupamento e de *hash*, criado sobre uma pilha, quando um índice sem agrupamento é criado sobre a pilha, o *SQL Server* usa os

identificadores de registros das páginas de índice que indicam os registros das páginas de dados; Índices sem agrupamento ou de *hash* criados sobre um índice agrupado ou ordenado, quando um índice sem agrupamento é criado sobre uma tabela com um índice de agrupamento, o *SQL Server* usa uma chave de agrupamento nas páginas de índice que indicam o índice de agrupamento. A chave de agrupamento armazena informações sobre a localização dos dados (WAGNER, 2007).

2.4 TRABALHOS RELACIONADOS

Para além dos pontos previamente destacados, alguns trabalhos com a temática de contribuições de ambientes digitais para a disseminação de conhecimento, estão relacionados com o "Qual a boa, Dev?". Dentre eles, destaca-se o livro "Internet e democratização do conhecimento: repensando o processo de exclusão social" do autor (TEIXEIRA, 2001) em que são trazidas as reflexões sobre a sociedade da informação que possui como característica principal, a crescente valorização da informação enquanto recurso estratégico, chegando até mesmo a superar a importância do capital, o papel das máquinas como potencializadoras, principalmente computadores, das possibilidades de atuação e interferência do homem na sociedade e a ideia de conhecimento estratégico que pode fornecer a população meios de superar suas próprias condições sociais.

Outro trabalho importante é o de (FARIA, 2009), denominado "A tecnologia da informação e da comunicação como ferramenta para a construção e democratização do conhecimento", em seu trabalho acadêmico o autor enfatiza a melhoria na qualidade do processo de ensinar e aprender ao utilizar ambientes de aprendizagem que apresentam caráter ativo e como estimular isso a partir de um computador gera um processo construcionista, no qual o aluno explora seu desenvolvimento de ideias, mediante ações sucessivas.

Considerando-se tais abordagens, podem ser destacadas algumas diferenças principais, ambos tem foco na disseminação do conhecimento de maneira geral, como o uso de tecnologia de forma mais incisiva poderia aperfeiçoar o ensino escolar e quais seriam as etapas necessárias para essa inserção, enquanto o "Qual a boa, dev?" está direcionado para o conhecimento especificamente tecnológico e de programação. Outro ponto é que o "Qual a boa, dev?" não só contribui para a reflexão desses tópicos como também em si já se trata de uma solução, por se tratar de um ambiente digital que possibilita a troca de informações, nesse caso um fórum online.

3 QUAL A BOA, DEV?

Neste capítulo serão abordadas as etapas principais que foram necessárias para concepção da aplicação, como o levantamento de requisitos. Em seguida, criação do diagrama de casos de uso e protótipos, definição da arquitetura do sistema, seleção das tecnologias utilizadas e efetuação das telas do sistema.

3.1 ANÁLISE

Nesta seção estão contidos os artefatos de análise de *software* desenvolvidos durante o planejamento do escopo da ferramenta.

3.1.1 Elicitação de requisitos

Como afirma Filho et al. (2017), a elicitação de requisitos é uma etapa crucial para o sucesso de um projeto de desenvolvimento de *software*, pois é responsável por extrair as necessidades e os desejos dos envolvidos que serão traduzidas em funcionalidades implementadas na solução final. Os requisitos do sistema apresentados no Quadro 1 foram levantados juntamente com o professor orientador Francisco Paulo Freitas.

Quadro 1 – Requisitos da aplicação Qual a boa,Dev?

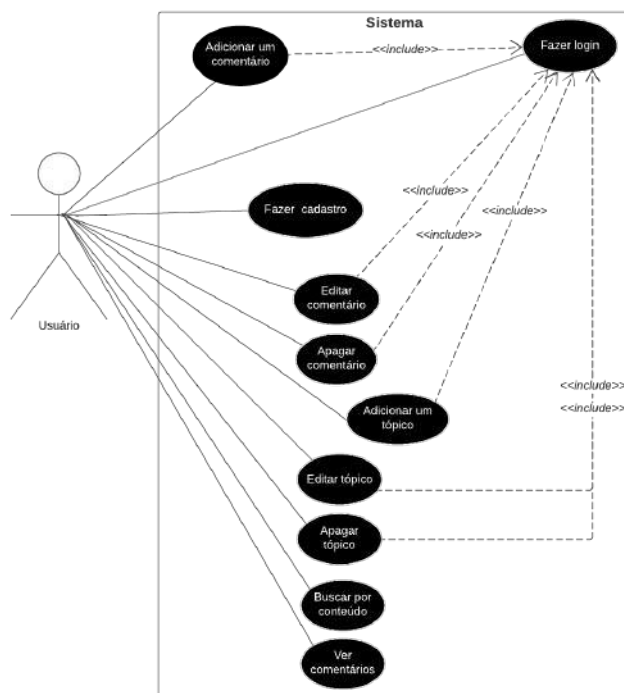
ID	REQUISITO	DESCRIÇÃO
RF01	Fazer cadastro	Permite que os usuários realizassem seu cadastro para poder utilizar o sistema
RF02	Fazer login	Permite que o usuário realizasse login para que possa acessar o sistema.
RF03	Exibir postagens	Exibe o conteúdo inserido pelos usuários no fórum.
RF04	Adicionar comentário	Permite que o usuário adicione comentários nas postagens.
RF05	Editar comentário	Permite que o usuário atualize comentários adicionados por ele em postagens.
RF06	Apagar comentário	Permite que o usuário apague comentários que ele inseriu no fórum.
RF07	Criar novo tópico	Permite que o usuário insira novos conteúdos no fórum.
RF08	Editar tópico	Permite que o usuário edite conteúdos inseridos por ele no fórum.
RF09	Apagar tópico	Permite que o usuário apague conteúdos inseridos por ele no fórum.
RF10	Buscar por conteúdo	Permite que o usuário realize buscas por conteúdo dentro da aplicação.
RF11	Ver comentários	Permite que o usuário visualize os comentários das postagens.

Fonte: Elaborado pela autora (2023)

3.1.2 Diagrama de casos de uso

Casos de uso são descrições textuais da funcionalidade do sistema a partir da perspectiva do usuário, usados para mostrar a funcionalidade que o sistema oferecerá e quais usuários de alguma forma se comunicarão com o sistema quando prover essa funcionalidade(CASTRO, 2001). A Figura 6 apresenta o diagrama de casos de uso da ferramenta proposta que exhibe os papéis exercidos por cada entidade no sistema.

Figura 6 – Diagrama de casos de uso



Fonte: Elaborado pela autora (2023)

O usuário é o único responsável por realizar as interações dentro do sistema, principalmente através de comentários nas postagens de outros e adição dos próprios tópicos dentro do fórum.

O Quadro 2 apresenta o detalhamento do caso de uso Fazer cadastro. Nele, são detalhados os atores, fluxos e condições. Os demais casos de usos podem ser visualizados no Apêndice A deste documento.

Quadro 2 – Caso de uso: Fazer cadastro

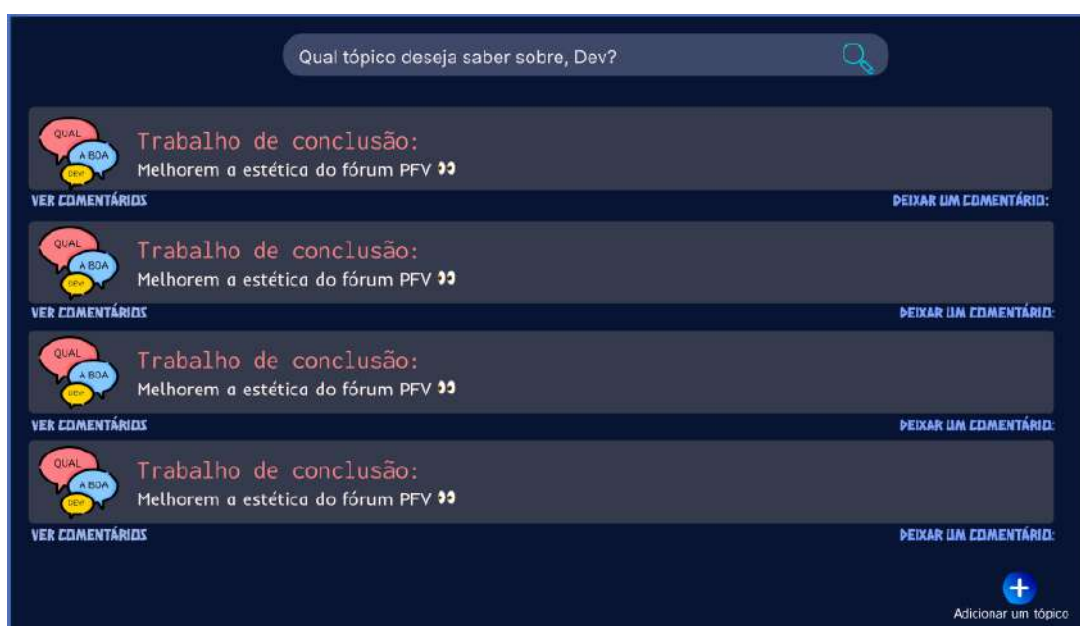
UC-01 Fazer cadastro	
Ator principal	Usuário
Atores secundários	Nenhum
Resumo	O usuário insere seu nome, email e senha para se cadastrar no sistema
Pré-condições	Nenhuma
Pós-condições	O usuário é cadastrado no sistema
Ações do Ator	Ações do Sistema
1. O usuário insere seu nome, email e senha.	2. O sistema cadastra o usuário.
	3. O sistema redireciona o usuário para tela de login.

Fonte: Elaborado pela autora (2023)

3.1.3 Protótipos

Protótipo é um produto unitário usado para testes feitos antes da fabricação em grande escala. Versão de um sistema que antecede à principal, normalmente reduzida, para ser aperfeiçoada (PALHAIS, 2015). Na Figura 7 é mostrada a tela inicial do fórum, é por ela que o usuário tem o contato inicial direto com a proposta da aplicação em si. Para exemplificação de tal interface foi usada a plataforma de *design* de protótipos *Figma*.¹

Figura 7 – Protótipo da página inicial do fórum



Fonte: Elaborado pela autora (2023)

Todas as telas pertinentes a plataforma em questão já foram submetidas ao processo de prototipagem e implementação. Para uma visualização mais detalhada e completa desses elementos, é possível sua visualização por meio do *link*² que apresenta o sistema no Figma.

3.2 PROJETO

Nesta seção estão contidos os artefatos de projeto de *software* desenvolvidos durante o planejamento do escopo da aplicação.

¹ <https://www.figma.com/>

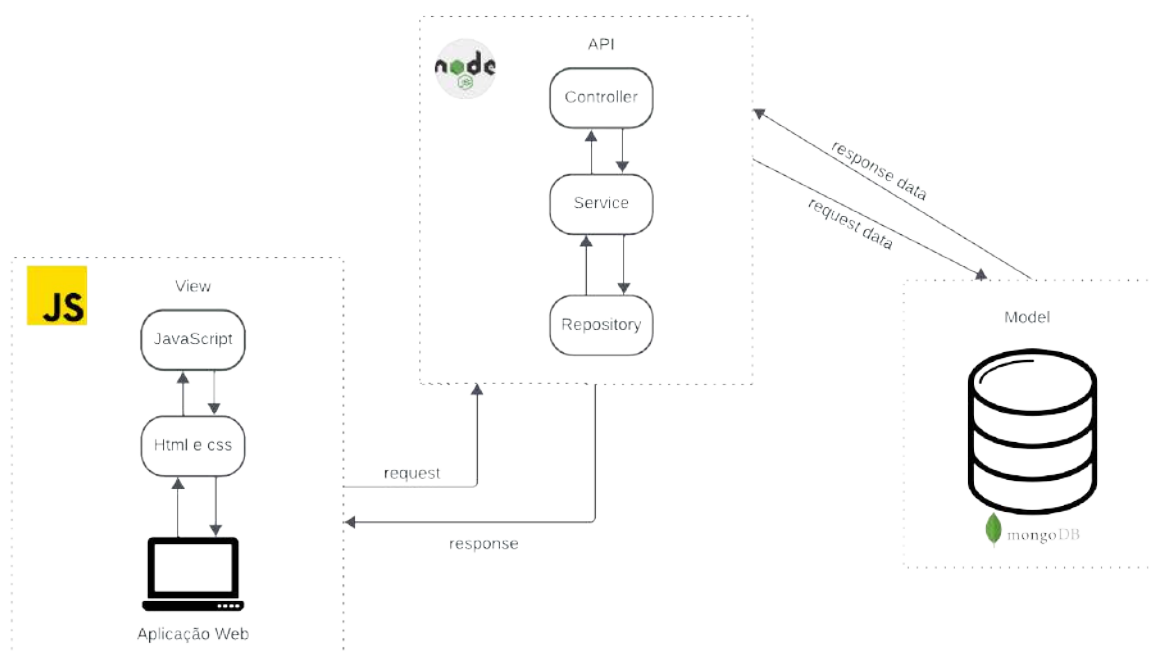
² <https://www.figma.com/proto/XxTPa4zXiuUsjlSC5YJxug/TCC-I?t=MWpp9cuoOXAWeLWr-1>

3.2.1 Arquitetura

A arquitetura do *software* foi efetuada a partir da junção do protocolo *REST* (Representational State Transfer) e modelo *MVC* (Model-View-Controller), o qual, auxilia os desenvolvedores a construir aplicações separando seus principais componentes, a manipulação e armazenamento dos dados, as funções que irão trabalhar com as entradas dos dados e a visualização do usuário. O componente *Model* (Modelo) é o objeto de aplicação, a *View* (Visualização) é a interface visualizada pelo usuário e o *Controller* (Controlador) trabalha em relação às entradas de uma *View* e como as mesmas reagirão. A arquitetura *MVC* especifica onde cada tipo de lógica deve estar localizada na aplicação (LEMOS et al., 2013).

A Figura 8 demonstra como este método foi aplicado dentro do sistema desenvolvido.

Figura 8 – Arquitetura geral do sistema



Fonte: Elaborado pela autora (2023)

Conforme (LEMOS et al., 2013), a camada *Model* contém a comunicação com os dados armazenados que serão visualizados na *View*, podendo estar armazenado em um banco de dados, em um arquivo *XML* ou em qualquer meio para tal funcionalidade. É somente na *Model* que as operações de criar, ler, atualizar e remover (*CRUD*), operações básicas em um banco de dados, podem ocorrer. Na aplicação em questão foi utilizado o banco *MongoDB* para manipulação dos dados e *Local Storage* (Armazenamento Local) para persistência da sessão do usuário.

A *View* é a camada que especifica exatamente como a *Model* deve ser apresentada. É a interface do usuário. A visão é dinâmica se adequando a qualquer modificação da *Model* (LUCIANO; ALVES, 2011). Diante disso, foram utilizadas as linguagens de marcação *Html* e *Css*, com complementação da linguagem de programação *Javascript* para permitir a criação de uma aplicação *web* interativa e dinâmica, tornando a experiência do usuário mais agradável e atraente.

A *Controller* traduz interações com a *View* em ações que devem ser desempenhadas em cima da *Model*. Tem como funções: Ativar processos de negócio, mudar o estado da *Model* e escolher a *View* apropriada de acordo com as ações dos usuários e com os resultados das mudanças na *Model*. (DUARTE, 2011). Mediante este conceito, o sistema foi elaborado partindo de uma *API* baseada no *Node.js* e *Express* para oferecer os serviços *back-end* e enviar os dados para a *View*.

3.3 TECNOLOGIAS UTILIZADAS

Nesta seção são explanadas as tecnologias utilizadas durante o processo de desenvolvimento do sistema.

3.3.1 Javascript

Consoante o autor (ALMEIDA, 2017), o *JavaScript*¹ surgiu em 1995 para rodar no navegador *Netscape* e posteriormente foi adaptado para a maioria dos navegadores *web*. Idealizado inicialmente para rodar nos navegadores atualizando o conteúdo dinamicamente. Esse contexto contribuiu para que evoluísse de uma forma diferente das outras linguagens focado em performance, criação de interfaces e dar uma melhor experiência para o usuário.

Conforme aconteceu a evolução dos navegadores o *JavaScript* acompanhou essa evolução. Com o surgimento do conceito de nuvem as aplicações necessariamente precisam ser escaláveis e com esse novo contexto foi necessário tirar maior proveito das máquinas e utilizar o mínimo de recurso possível. Houveram também outras propostas como divisão de carga e micro-serviços.

3.3.2 Node.js

Segundo (NUNES, 2018) o *Node.js*², lançado em 2009, é uma plataforma que chama a atenção por permitir a execução de programas escritos em *Javascript* fora de um navegador, permitindo o desenvolvimento completo de aplicações na mesma

¹ <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/>

² <https://nodejs.org/>

linguagem. A plataforma foi desenvolvida a partir do interpretador de *Javascript* do *Google Chrome*: *V8* com a finalidade de interpretar códigos em *Javascript* no *backend*, possibilitando criar aplicações apenas com uma única linguagem.

Além disso, utilizando *Node.js* é possível criar aplicações *REST* com o *framework open source* de desenvolvimento *Express*. O próprio *framework* tem uma série de *features* para trabalhar com o desenvolvimento *web* e com os métodos de requisição *API*.

Possui uma estrutura orientada a eventos e uma forma de entrada e saída que faz com que ele seja leve e eficiente. Isso torna o *Node.js* “poderoso”, porque essas características são essenciais para um melhor desempenho de intenso tráfego de rede e para aplicações em tempo real, que hoje, são considerados grandes empecilhos da *web* (BARSOTI; GIBERTONI, 2020).

3.3.3 Express

De acordo com (BARSOTI; GIBERTONI, 2020), o *Express*³ atua como uma camada no topo do *Node.js*, facilitando e deixando o desenvolvimento de *APIs* em *Node* mais prático. Além disso, com ele é mais fácil organizar as funcionalidades da aplicação usando *middleware* e roteamento, facilita a renderização de páginas *HTML* dinâmicas. Ele define um padrão de extensibilidade facilmente implementado e adiciona utilitários para os objetos *HTTP* do *Node.js*.

O *framework* ajuda a resolver alguns problemas que os desenvolvedores têm com o *Node.js*, como gerenciar corpos de solicitações *HTTP*, gerenciar *cookies*, gerenciar sessões, organização de rotas, determinar cabeçalhos de respostas adequados. O *Express.js* é considerado minimalista, porém, permite aos desenvolvedores uma liberdade para criar pacotes *middleware* específicos com o objetivo de resolver problemas que surgem durante o desenvolvimento de uma aplicação. Além disso, é um *framework* não opinativo, ou seja, o desenvolvedor pode inserir qualquer *middleware* que prefira para o manuseio das solicitações e pode estruturar a aplicação em um arquivo ou em vários, usar qualquer estrutura de pastas dentro do diretório.

3.3.4 MongoDB

De acordo com (GERALDO, 2022), o *MongoDB*⁴ foi desenvolvido como um componente de serviço, em outubro do ano de 2007 e em 2009 passou a ser um *software open source* ou um *document store* onde não é reconhecido um esquema

³ <https://expressjs.com/pt-br/>

⁴ <https://www.mongodb.com/>

relacional; os documentos semelhantes ao formato *JSON* (Java Script Object Notation) para armazenar dados. O documento é semelhante a um registro com campos e valores.

Ademais, fornece recursos para ambientes de produção como balanceamento de cargas, replicação, consultas a sistemas de arquivos e conectividade com diversos *softwares*, algumas funcionalidades são a conexão com o serviço de computação em nuvem, sincronia com dispositivos móveis e a criação de funções e gatilhos para automações.

As unidades de dados no *MongoDB* são os documentos tais quais comparáveis a estrutura *JSON*, no entanto no *MongoDB* são denominados *BSON*, sigla em inglês (*Binary encoded Javascript Object Notation*), devido à maneira como os dados são armazenados, é possível fazer a analogia de que o *MongoDB* é considerado um banco de dados orientado a documentos, sendo esta uma das características mais marcantes nos bancos *NoSQL*, a forma simples como as consultas são executadas tendo alta eficiência (GERALDO, 2022).

3.3.5 Local Storage

O *local storage* (armazenamento local) é uma forma de armazenamento de dados persistente, ou seja, os dados não serão deletados se o navegador for fechado, que permite que dados sejam salvos diretamente no navegador.(GARCEZ, 2023)

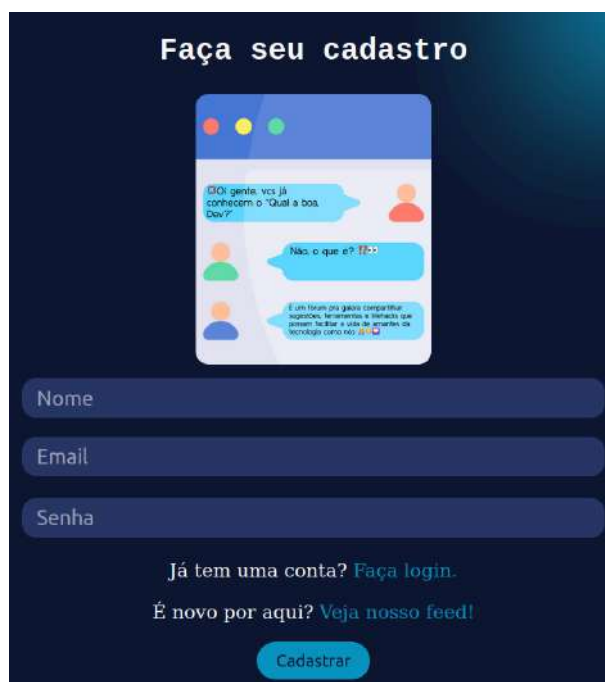
Os dados são armazenados em um formato de chave-valor, sendo que o valor sempre será armazenado no formato *string*, o que significa que na hora de acessar os dados armazenados pode ser preciso fazer algum tipo de conversão.

Ainda segundo (GARCEZ, 2023), deve ser usado em casos onde se deseja armazenar alguma informação do lado do cliente por tempo indeterminado, como, por exemplo, preferências relacionadas ao tema de uma interface.

3.4 TELAS DA APLICAÇÃO

Nesta seção, são mostradas as telas produzidas para a concretização do aplicativo, acompanhadas de descrições de suas funcionalidades e características. A Figura 9 apresenta a primeira tela que é exibida para o usuário para que através dela ele possa inserir suas informações e efetuar seu cadastro no sistema, ir diretamente para a tela principal, utilizando o botão "Veja nosso feed!"ou ser redirecionado para a tela de login.

Figura 9 – Tela de cadastro do sistema



A imagem mostra a tela de cadastro do sistema, intitulada "Faça seu cadastro". No topo, há um ícone de janela com três pontos coloridos (vermelho, amarelo, verde). Abaixo, há uma simulação de uma conversa em uma interface de mensagens. O primeiro usuário pergunta: "Oi! gente, vcs já conhecem o 'Qual a boa Dev?'". O segundo usuário responde: "Não, o que é?!!?". O terceiro usuário responde: "É um feed pra gente compartilhar sugestões, lembranças e links que podem ajudar a vida de quem tá trabalhando como nós! 🙌". Abaixo da simulação, há três campos de entrada de texto: "Nome", "Email" e "Senha". No final da tela, há o texto "Já tem uma conta? [Faça login.](#)" e "É novo por aqui? [Veja nosso feed!](#)". Um botão "Cadastrar" está localizado no rodapé da tela.

Fonte: Elaborado pela autora (2023)

A Figura 10 exibe a tela de *login* da plataforma, ela pode ser acessada a partir da tela de cadastro pelo botão "Faça login", nela o usuário insere seus dados e acessa sua conta ou pode ter uma prévia experiência de uso ao acessar a tela inicial diretamente, ao utilizar o botão "Veja nosso feed!".

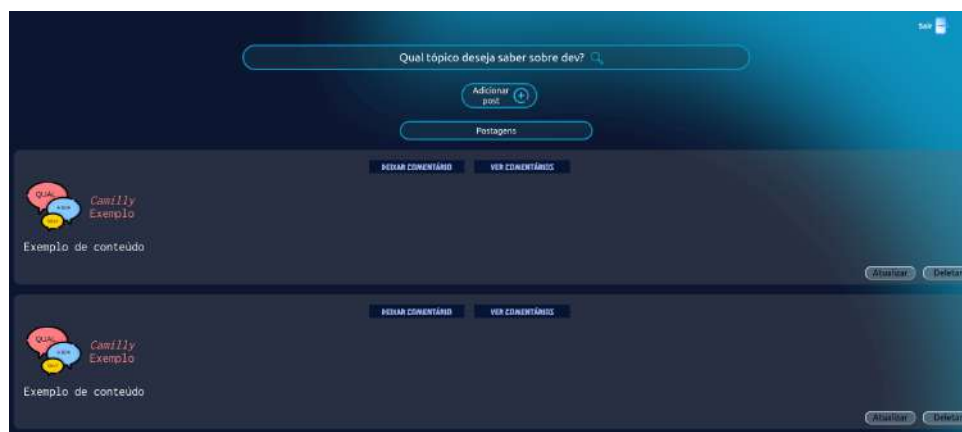
Figura 10 – Tela de login do sistema



Fonte: Elaborado pela autora (2023)

A Figura 11 apresenta a tela principal do fórum, onde é possível visualizar os conteúdos já existentes, interagir com os de outros usuários pelos comentários, ao clicar no botão "Deixar comentário", visualizar os comentários já existentes com o "Ver comentários", atualizar ou remover postagens de própria autoria, ser redirecionado para a tela de adição de postagens com o botão "Adicionar Post" e realizar pesquisas através do "Qual tópico deseja saber sobre, Dev?".

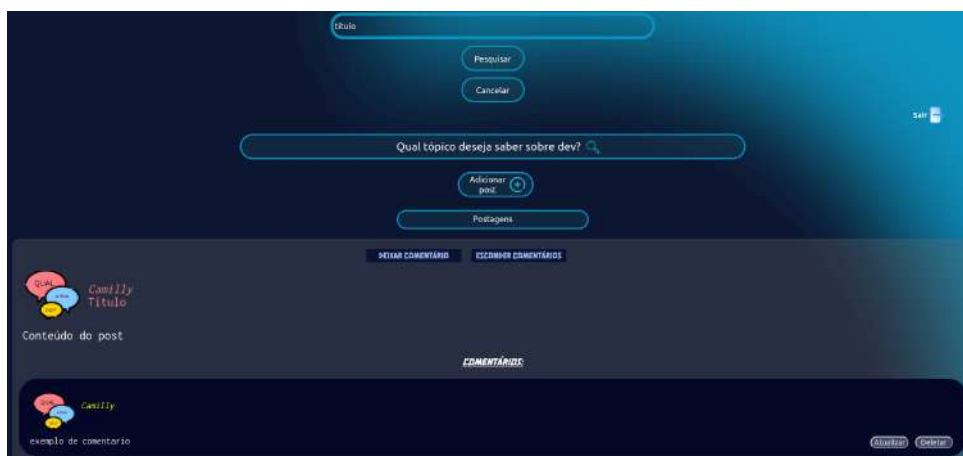
Figura 11 – Tela inicial da aplicação



Fonte: Elaborado pela autora (2023)

A Figura 12 apresenta como o usuário pode realizar a busca por conteúdo dentro do fórum e também como é a visualização dos comentários de uma postagem, permitindo juntamente com isso a possibilidade de atualização ou remoção do comentário se o autor for o usuário logado no momento.

Figura 12 – Tela de busca e exibição de comentários



Fonte: Elaborado pela autora (2023)

A Figura 13 apresenta como o usuário pode adicionar suas postagens dentro do fórum e retornar à principal, tal tela pode ser acessada a partir da principal.

Figura 13 – Tela de criação de postagens



Fonte: Elaborado pela autora (2023)

4 CONSIDERAÇÕES FINAIS

É imprescindível destacar que a proposta do "Qual a boa, Dev", representa não somente um recurso de acesso ao conhecimento, mas também uma ferramenta que visa reduzir as barreiras existentes no aprendizado de tecnologia. Ao tornar disponíveis informações relevantes de maneira envolvente e de fácil compreensão, essa iniciativa buscou contribuir para a diminuição das disparidades no acesso ao conhecimento técnico, promovendo, assim, maior inclusão e democratização do saber na área de tecnologia.

De modo geral, os objetivos principais deste projeto foram alcançados, sendo encontradas somente dificuldades relacionadas a oferta de trabalhos acadêmicos que também oferecessem como solução o desenvolvimento de uma aplicação e que focassem especificamente no setor tecnológico.

Atualmente ele se encontra em processo de registro junto ao IFPB e em relação a trabalhos futuros, poderiam ser consideradas melhorias de segurança, ampliação das funcionalidades, como adesão de recursos premium, visando possibilidade de oferta de assinaturas, criação de relatórios para análise e integração com outras plataformas.

REFERÊNCIAS

ALMEIDA, A. R. R. de Sousa e W. H. C. Desenvolvendo aplicações restful utilizando node.js. 2017.

AWARI. 2022. Acesso em 28/11/2023.

BAGGIO, R. A sociedade da informacao e a infoexclusao. **Ciencia da Informacao**, SciELO Brasil, v. 29, p. 16–21, 2000.

BARSOTI, N.; GIBERTONI, D. Impacto que o sequelize traz para o desenvolvimento de uma api construida em node. js com express. js. **Revista Interface Tecnologica**, v. 17, n. 2, p. 231–243, 2020.

BATISTA, C. L. As dimensoes da informacao publica: transparencia, acesso e comunicacao. **Transinformacao**, SciELO Brasil, v. 22, p. 225–231, 2010.

CASTRO, J. 2001.

CAZELOTO, E. Inclusao digital: uma visao critica. Editora Senac Sao Paulo, 2019.

CETESC. **4.2. Índices e suas importâncias - Aprendendo Programar**. 2022. <https://aprendendoprogramar.com.br/tutoriais/postgresql/04_restricoes_indices/02_indices_importancia/>. (Accessed on 08/02/2024).

COUTINHO, T. **MongoDB: o que é, pra que serve e por que utilizá-lo?** 2023. <<https://www.voitto.com.br/blog/artigo/o-que-e-mongodb>>. (Accessed on 11/28/2023).

DUARTE, A. R. Metodologia rails: Analise da arquitetura model view controller aplicada. 2011.

FARIA, E. V. de. A tecnologia da informação e da comunicação como ferramenta para a construção e democratização do conhecimento. 2009.

FILHO, P. G. R.; SILVA, L. G. da; TEIXEIRA, C. A. A.; VASCONCELOS, A. P. V. de; HORA, H. R. M. da. Uma proposta de elicitaÇÃO de requisitos de software auxiliada pela modelagem da visÃO futura dos processos de negócios. 2017.

FRIESS, I. I. Analise de bancos de dados nosql e desenvolvimento de uma aplicacao. Universidade Federal de Santa Maria, 2013.

GARCEZ, L. 2023. (Accessed on 02/09/2024).

GERALDO, E. H. Pymongo com mongo db, uma opcao atual aos sgbds relacionais. 004, 2022.

GROSSI, M. G. R.; COSTA, J. W. da; SANTOS, A. J. dos. A exclusao digital: o reflexo da desigualdade social no brasil. **Nuances: estudos sobre Educacao**, v. 24, n. 2, p. 68–85, 2013.

IBM. O que é mongodb? | ibm. 2021.

J, G. **O que são indexes de bancos de dados e quando pode ser útil em nossa aplicação?** 2023. <https://dev.to/gabrielgcj/o-que-sao-indexes-de-bancos-de-dados-e-quando-pode-ser-util-em-nossa-aplicacao-4kmg>. (Accessed on 19/08/2024).

JUNIOR, E. Bancos de dados relacionais vs não relacionais: Qual é a melhor opção para engenheiros de dados? **DIO**, 2023.

KOKAY, M. C. **Banco de dados NoSQL: um novo paradigma**. [S.l.]: Revista SQL Magazine 102, 2012.

LEMOS, 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 aplicacao web (webapps). **RE3C-Revista Eletronica Cientifica de Ciencia da Computacao**, v. 8, n. 1, 2013.

LIMA, G. Rest: Conceito e fundamentos. 2020.

LUCIANO, J.; ALVES, W. J. B. Padrao de arquitetura mvc: model-view-controller. **Revista EPeQ Fafibe**, 2011.

MACANA, E. C.; BOLELLI, R. 2023. Acesso em 28/11/2023.

NUNES, G. N. As vantagens do node. js. **REFAQI-REVISTA DE GESTAO EDUCACAO E TECNOLOGIA**, v. 4, n. 2, p. 2–2, 2018.

OLIVEIRA, S. S. de. Bancos de dados nao-relacionais: um novo paradigma para armazenamento de dados em sistemas de ensino colaborativo. **Revista da Escola de Administracao Publica do Amapa**, v. 2, n. 1, p. 184–194, 2014.

PALHAIS, C. B. C. **PROTOTIPAGEM Uma abordagem ao processo de desenvolvimento de um produto**. 2015.

PALHANO, V. **O que é uma API de REST?** 2024. (Accessed on 20/02/2024).

REIS, E. C. dos; COSTA, R. M. de R. Pesquisas textuais: Como acelerar respostas e cruzar dados baseados em um mesmo contexto. **Caderno de Estudos em Sistemas de Informacao**, v. 6, n. 2, 2022.

SCHUERZOSKI, M. R. Estudo e aplicacao de tuning em sistema gerenciador de banco de dados orientado a documentos. 2019.

SILVA, H. S. I. L. Api rest na plataforma a-cdm. 2019.

TEIXEIRA, A. C. **Internet be democratização do conhecimento: repensando o processo de exclusão social**. Dissertação (Mestrado) — Universidade de Passo Fundo, 2001.

VITOR. Utilizando full-text search - sql magazine 82. **DEV MEDIA**, 2010.

WAGNER. **Entendendo e usando índices no SQL**. 2007. <<https://www.devmedia.com.br/entendendo-e-usando-indices/6567>>. (Accessed on 08/02/2024).

WANZELLER, D. A. P. Investigando o uso de bancos de dados orientados a documentos para gerenciar informacoes da administracao publica. 2013.

APÊNDICE A – DETALHAMENTO DOS CASOS DE USO

Quadro 3 – Caso de uso: Fazer login

UC-02 Fazer login	
Ator principal	Usuário
Atores secundários	Nenhum
Resumo	O usuário insere seu email e senha para acessar o sistema.
Pré-condições	O usuário estar cadastrado.
Pós-condições	O usuário acessa o sistema
Ações do Ator	Ações do Sistema
	1. O sistema redireciona o usuário para tela de login.
2. O usuário insere seu email e senha.	3. O sistema permite que o usuário visualize a tela de fórum.

Fonte: Elaborado pela autora (2023)

Quadro 4 – Caso de uso: Adicionar comentário

UC-03 Adicionar comentário	
Ator principal	Usuário
Atores secundários	Nenhum
Resumo	O usuário interage com a postagem de outros inserindo comentários.
Pré-condições	O usuário estar logado no sistema.
Pós-condições	O usuário acessa o sistema e consegue adicionar comentários em postagens
Ações do Ator	Ações do Sistema
	1. O sistema permite que o usuário visualize o fórum.
2. O usuário clica no botão "deixar um comentário" em uma postagem e insere seu comentário.	3. O sistema exibe o comentário adicionado.

Fonte: Elaborado pela autora (2023)

Quadro 5 – Caso de uso: Editar comentário

UC-04 Editar comentário	
Ator principal	Usuário
Atores secundários	Nenhum
Resumo	O usuário edita os comentários feitos por ele em postagens.
Pré-condições	O usuário estar logado no sistema.
Pós-condições	O usuário acessa o sistema e consegue editar comentários feitos por ele em postagens
Ações do Ator	Ações do Sistema
	1. O sistema permite que o usuário visualize o fórum.
2. O usuário visualiza o comentário inserido por ele em postagens e clica no botão de editar.	3. O sistema exibe o estado atual do comentário após a edição.

Fonte: Elaborado pela autora (2023)

Quadro 6 – Caso de uso: Apagar comentário

UC-05 Apagar comentário	
Ator principal	Usuário
Atores secundários	Nenhum
Resumo	O usuário apaga os comentários feitos por ele em postagens.
Pré-condições	O usuário estar logado no sistema.
Pós-condições	O usuário acessa o sistema e consegue apagar comentários feitos em postagens
Ações do Ator	Ações do Sistema
Resumo	O usuário edita os comentários feitos por ele em postagens.
Pré-condições	O usuário realizar cadastro e login
Pós-condições	O usuário acessa o sistema e consegue editar comentários feitos por ele em postagens
Ações do Ator	Ações do Sistema
	1. O sistema permite que o usuário visualize o fórum.
2. O usuário visualiza o comentário inserido por ele na postagem e clica no botão de apagar.	3. O sistema remove o comentário e exibe o estado atual da postagem após a remoção.

Fonte: Elaborado pela autora (2023)

Quadro 7 – Caso de uso: Adicionar um tópico

UC-06 Adicionar um tópico	
Ator principal	Usuário
Atores secundários	Nenhum
Resumo	O usuário consegue adicionar tópicos sobre temas que se interesse.
Pré-condições	O usuário estar logado no sistema
Pós-condições	O usuário acessa o sistema e consegue fazer as próprias postagens
Ações do Ator	Ações do Sistema
	1. O sistema permite que o usuário visualize o fórum.
2. O usuário clica no botão "adicionar um tópico" e insere o seu.	3. O sistema exibe o tópico adicionado.

Fonte: Elaborado pela autora (2023)

Quadro 8 – Caso de uso: Editar tópico

UC-07 Editar tópico	
Ator principal	Usuário
Atores secundários	Nenhum
Resumo	O usuário consegue editar os tópicos adicionados por ele na aplicação.
Pré-condições	O usuário estar logado no sistema
Pós-condições	O usuário acessa o sistema e consegue editar as próprias postagens
Ações do Ator	Ações do Sistema
	1. O sistema permite que o usuário visualize o fórum.
2. O usuário visualiza um tópico inserido por ele no fórum e clica no botão de editar.	3. O sistema exibe o estado atual do tópico após a edição.

Fonte: Elaborado pela autora (2023)

Quadro 9 – Caso de uso: Apagar tópico

UC-08 Apagar tópico	
Ator principal	Usuário
Atores secundários	Nenhum
Resumo	O usuário consegue apagar tópicos adicionados por ele na aplicação.
Pré-condições	O usuário estar logado no sistema.
Pós-condições	O usuário acessa o sistema e consegue apagar as próprias postagens
Ações do Ator	Ações do Sistema
	1. O sistema permite que o usuário visualize o fórum.
2. O usuário visualiza um tópico inserido por ele no fórum e clica no botão de apagar.	3. O sistema remove o tópico e exibe o estado atual do fórum após a remoção.

Fonte: Elaborado pela autora (2023)

Quadro 10 – Caso de uso: Buscar por conteúdo


UC-09 Buscar por conteúdo	
Ator principal	Usuário
Atores secundários	Nenhum
Resumo	O usuário consegue buscar tópicos por meio de conteúdo na aplicação.
Pré-condições	O usuário estar logado no sistema
Pós-condições	O usuário acessa o sistema e consegue pesquisar tópicos por conteúdo na aplicação
Ações do Ator	Ações do Sistema
	1. O sistema permite que o usuário visualize o fórum.
2. O usuário digita o conteúdo que deseja visualizar sobre na barra de pesquisa.	3. O sistema exibe as postagens relacionadas com a pesquisa que o usuário fez.

Fonte: Elaborado pela autora (2023)

Quadro 11 – Caso de uso: Ver comentários

UC-10 Ver comentários	
Ator principal	Usuário
Atores secundários	Nenhum
Resumo	O usuário visualiza os comentários das postagens do fórum.
Pré-condições	O usuário estar logado no sistema
Pós-condições	O usuário acessa o sistema e consegue visualizar os comentários das postagens
Ações do Ator	Ações do Sistema
	1. O sistema permite que o usuário visualize o fórum.
2. O usuário clica no botão "ver comentários" em uma postagem e visualiza os comentários feitos nela por ele e outros.	3. O sistema exibe os comentários do tópico.

Fonte: Elaborado pela autora (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:	Debora Camilly
Tipo do Documento:	Tese
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Debora Camilly Quirino da Silva, DISCENTE (202112010014) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 04/09/2024 14:52:50.

Este documento foi armazenado no SUAP em 04/09/2024. 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: 1239212

Código de Autenticação: a86e3ba2df

