

colcom: Uma plataforma web participativa para a deliberação coletiva e desenvolvimento de ideias

João Wallace Lucena Lins



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, PB

2024

João Wallace Lucena Lins

colcom: Uma plataforma web participativa para a deliberação coletiva e desenvolvimento de ideias

Relatório Técnico apresentado ao curso Engenharia da Computação
do Centro de Informática, da Universidade Federal da Paraíba,
como requisito para a obtenção do grau de Bacharel em Engenharia da Computação

Orientador: Yuri de Almeida Malheiros Barbosa

Maio de 2024

Catálogo na publicação
Seção de Catalogação e Classificação

L759c Lins, João Wallace Lucena.

Colcom: uma plataforma web participativa para a
deliberação coletiva e desenvolvimento de ideias / João
Wallace Lucena Lins. - João Pessoa, 2024.

46 f. : il.

Orientação: Yuri de Almeida Malheiros Barbosa, Thaís
Gaudencio do Rêgo.

TCC (Graduação) - UFPB/CI.

1. Deliberação. 2. Democracia digital. 3. Redes
sociais. 4. Participação social. 5. Aplicação web. I.
Barbosa, Yuri de Almeida Malheiros. II. Rêgo, Thaís
Gaudencio do. III. Título.

UFPB/CI

CDU 004.738.5



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Engenharia da Computação intitulado ***col-com: Uma plataforma web participativa para a deliberação coletiva e desenvolvimento de ideias*** de autoria de João Wallace Lucena Lins, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Dr. Yuri de Almeida Malheiros Barbosa
Universidade Federal da Paraíba

Prof. Dra. Thaís Gaudencio do Rêgo
Universidade Federal da Paraíba

Prof. Dr. Alexandre Nóbrega Duarte
Universidade Federal da Paraíba

João Pessoa, 23 de maio de 2024

A minha família, amigos e povo do Brasil.

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a meus pais, José Wallace e Ivônia, pois sem seu apoio e influência não seria quem sou hoje e não poderia sequer sonhar em conquistar o que conquistei.

Em seguida, agradeço meu irmão Chrístian por ser uma companhia sempre presente em minha vida, e por ser sempre o primeiro a testar os limites das minhas ideias.

Não posso deixar de agradecer aos meus outros familiares, tios, tias, primos, primas, avós e avôs falecidos, e inclusive aos meus amigos — os quais considero minha segunda família —, cada um deles que me marcou e me moldou de alguma forma.

Também agradeço a Hevelyn Mafra, por ter sido uma namorada amorosa e companheira, sendo um grande apoio para mim durante a produção deste trabalho e de boa parte de minha graduação.

Agradeço ao meu orientador Dr. Yuri Malheiros, pela disponibilidade e solicitude, o que me habilitou a fazer este trabalho com segurança. Ademais, também o agradeço por ter sido um supervisor prestativo durante os projetos em que trabalhamos juntos.

Agradeço a minha professora e examinadora Dra. Thaís Gaudêncio, que, além de me ajudar enormemente na formulação do presente trabalho e a lidar com as burocracias com o processo de conclusão de curso, foi a primeira a me cativar para a área e acompanhou boa parte de minha trajetória na graduação;

Agradeço ao Dr. Alexandre Duarte, que prontamente aceitou o convite para participar da banca.

Agradeço a meus outros professores e mentores durante o curso, Anand Subramanian, que me aconselhou extensamente durante minha graduação; Marcelo Iury e Telmo Filho, que me guiaram muitas vezes em minhas experiências profissionais.

Agradeço, também, aos meus amigos que conheci durante minha graduação, principalmente João Pedro e Itamar, os quais foram influências muito positivas para meu desenvolvimento no curso e tornaram o peso do estudo um pouco menor.

Por fim, agradeço a todos que não me auxiliaram diretamente, mas que sem sua contribuição meu trabalho não seria possível: todos os trabalhadores da UFPB, todos os desenvolvedores que divulgam seus códigos e promovem o cenário *open-source* e todos aqueles que visaram ampliar o papel do povo nas decisões políticas.

RESUMO

Ferramentas de participação digital já são uma realidade no Brasil e em outros países, contudo, seu acesso ainda se restringe a uma pequena parcela dos usuários de internet; redes sociais por outro lado já são bem difundidas, mas sofrem de limitações significativas no que tange à deliberação eficaz. O objetivo principal da plataforma aqui proposta é ser um ambiente favorável para a participação e deliberação de indivíduos de uma certa comunidade em seus processos de decisão, promovendo, assim, a democracia digital em escala local. O presente relatório técnico descreve o processo de projeto, implementação e validação da aplicação de código aberto *colcom*. A plataforma utiliza dinâmicas de redes sociais e conceitos de Sistemas de Controle de Versão (SCV), particularmente o *Git*, como principais diretrizes de funcionamento.

Palavras-chave: deliberação, democracia digital, redes sociais, participação social, aplicação web.

ABSTRACT

Digital participation tools are already a reality in Brazil and other countries, however, their access is still restricted to a small portion of internet users; Social networks, on the other hand, are already widespread, but suffer from significant limitations in terms of effective deliberation. The main objective of the platform here proposed is to be a favorable environment for the participation and deliberation of individuals from a certain community in their decision-making processes, thus promoting digital democracy on a local scale. This technical report describes the design, implementation and validation process of the *colcom* open source application. The platform uses social network dynamics and Version Control Systems (VCS) concepts, particularly *Git*, as its main operating guidelines.

Key-words: deliberation, digital democracy, social networks, social participation, web application.

LISTA DE FIGURAS

1	Arquitetura da aplicação.	31
2	Diagrama entidade-relacionamento da aplicação.	32
3	Estrutura do <i>frame</i>	33
4	Estrutura do tópico.	34
5	Estrutura do post.	35
6	Tela de visualização dos tópicos.	36
7	Tela de visualização do <i>post</i>	37
8	Tela de um post com uma crítica sendo visualizada.	37
9	Tela de <i>login</i> e cadastro.	38
10	Formulário de cadastro.	38

LISTA DE TABELAS

1	Caso de teste [T01].	39
2	Caso de teste [T02].	40
3	Caso de teste [T03].	40
4	Caso de teste [T04].	40
5	Caso de teste [T05].	40
6	Caso de teste [T06].	41
7	Caso de teste [T07].	41
8	Caso de teste [T08].	41
9	Caso de teste [T09].	42
10	Caso de teste [T10].	42
11	Caso de teste [T11].	42
12	Caso de teste [T12].	43

LISTA DE ABREVIATURAS

CSS – *Cascading Style Sheets*

HTML – *Hypertext Markup Language*

HTTP – *Hypertext Transfer Protocol*

JS – *JavaScript*

JSON – *JavaScript Object Notation*

JSX – *JavaScript XML*

LGPD – Lei Geral de Proteção de Dados Pessoais

SCV – Sistema de controle de versões

SGBD – Sistema gerenciador de banco de dados

SQL – *Structured Query Language*

Sass – *Syntactically Awesome Style Sheets*

TS – *TypeScript*

Sumário

1	INTRODUÇÃO	19
1.1	Definição do Problema	20
1.2	Premissas e Hipóteses	20
1.3	Objetivo geral	20
1.4	Objetivos específicos	21
1.5	Estrutura do relatório técnico	21
2	FUNDAMENTAÇÃO TEÓRICA	22
2.1	Conceitos Gerais	22
2.1.1	<i>Front-end</i>	22
2.1.2	<i>Back-end</i>	22
2.1.3	Banco de dados	22
2.1.4	Sistema de controle de versões	23
2.2	Tecnologias utilizadas	23
2.2.1	Gerais	23
2.2.2	<i>Front-end</i>	23
2.2.3	<i>Back-end</i>	24
2.2.4	Banco de dados	25
2.3	Atores do processo	25
3	METODOLOGIA	26
3.1	Visão Geral	26
3.2	Requisitos Funcionais	26
3.2.1	[RF01] Cadastrar um usuário	26
3.2.2	[RF02] Efetuar o <i>login</i> de um usuário	26
3.2.3	[RF03] Avaliar um conteúdo como relevante ou não	27
3.2.4	[RF04] Salvar um conteúdo	27
3.2.5	[RF05] Criar um tópico	27
3.2.6	[RF06] Promover um tópico	27

3.2.7	[RF07]	Visualizar tópicos registrados	27
3.2.8	[RF08]	Visualizar tópicos salvos	27
3.2.9	[RF09]	Escrever corpo de um conteúdo	27
3.2.10	[RF10]	Criar um <i>post</i> em resposta a um tópico	27
3.2.11	[RF11]	Votar em um <i>post</i>	28
3.2.12	[RF12]	Editar um <i>post</i>	28
3.2.13	[RF13]	Criar uma sugestão de edição para um <i>post</i>	28
3.2.14	[RF14]	Criar um “ramo” de uma versão específica de um <i>post</i>	28
3.2.15	[RF15]	Visualizar uma versão específica de um <i>post</i>	28
3.2.16	[RF16]	Visualizar proposições de edição de um <i>post</i>	28
3.2.17	[RF17]	Acatar e rejeitar proposições de edição de um <i>post</i>	28
3.2.18	[RF18]	Visualizar críticas de uma versão específica de um <i>post</i> . . .	29
3.2.19	[RF19]	Realizar críticas em um <i>post</i>	29
3.3		Requisitos Não Funcionais	29
3.3.1		Usabilidade	29
3.3.1.1	[RNF01]	Intuitividade	29
3.3.1.2	[RNF02]	Responsividade	29
3.3.1.3	[RNF03]	Consistência	29
3.3.2		Desempenho	29
3.3.2.1	[RNF04]	Usuários simultâneos	29
3.3.2.2	[RNF05]	Tempo de resposta	30
3.3.3		Segurança	30
3.3.3.1	[RNF06]	Controle de acesso	30
3.3.3.2	[RNF07]	Obfuscação de dados sensíveis	30
3.3.4		Manutenibilidade	30
3.3.4.1	[RNF08]	SCV	30
3.3.4.2	[RNF09]	Código aberto	30
3.3.5		Portabilidade	30
3.3.5.1	[RNF10]	<i>Self-hosting</i>	30

3.3.5.2	[RNF11] Compatibilidade	30
3.3.6	Legalidade	31
3.3.6.1	[RNF12] Licença de Software	31
3.3.6.2	[RNF13] Lei Geral de Proteção de Dados Pessoais (LGPD)	31
3.4	Arquitetura	31
3.4.1	Banco de dados	31
3.5	Interface da aplicação	32
3.5.1	Principais componentes	33
3.5.1.1	<i>Frame</i>	33
3.5.1.2	Tópico	34
3.5.1.3	<i>Post</i>	34
3.5.1.4	Crítica	35
3.5.2	Páginas	35
3.5.2.1	Página de árvore de tópicos	35
3.5.2.2	Página de tópico	36
3.5.2.3	Página de <i>post</i>	36
3.5.2.4	Página de <i>login</i> e cadastro	36
3.5.2.5	Página de conteúdos salvos	37
3.5.2.6	Página de edição de texto	37
4	APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	39
4.1	Plano de testes	39
4.2	Casos de teste	39
4.3	Comparativo com outras plataformas	43
4.3.1	<i>Pol.is</i>	43
4.3.2	<i>Decidim</i>	44
4.3.3	<i>Consul</i>	44
4.3.4	<i>Wikilegis</i>	44
4.4	Análise dos resultados	45

5 CONCLUSÕES E TRABALHOS FUTUROS	46
REFERÊNCIAS	46

1 INTRODUÇÃO

A democracia digital é um conceito que começou a ser formulado após o advento e expansão da internet — que ocorreu concomitantemente à popularização de dispositivos que fazem uso dela, como computadores pessoais e aparelhos celulares — e que vem se modernizando e concretizando juntamente com o desenvolvimento tecnológico. Valendo-se desta difusão do acesso à internet, foram propostas, nacionalmente e internacionalmente, diversas ferramentas que visam expandir a prática da democracia.

No cenário internacional, um dos maiores exemplos é o sistema *Pol.is*¹ [1] que já foi utilizado em 9 países diferentes, sendo o seu caso de uso de maior destaque em Taiwan, com o processo *vTaiwan*², iniciado em 2014, em que, de acordo com seu próprio *website*, já foram discutidas mais de 28 pautas por mais de 200 mil participantes. 80% dessas discussões levaram à alguma ação governamental decisiva.

No cenário nacional, o projeto de maior destaque é a plataforma *Brasil Participativo*³, que tem o título de maior experiência de participação social na internet já realizada pelo governo federal [23]. O projeto contou com mais de 1,4 milhões de participantes em sua primeira iniciativa digital de participação, o Plano Plurianual Participativo 2024 - 2027, em que os ministérios tiveram a responsabilidade de analisar os conteúdos propostos. Em seguida, o Ministério do Planejamento e Orçamento elaborou um documento de devolutiva contendo o que foi incorporado e a justificativa do que não foi incorporado, e por fim, o plano foi instituído na lei 14.802/2024⁴, sancionada pelo presidente Lula.

Entretanto, ainda há uma grande defasagem na adoção destas ferramentas pela população em geral, angariando somente uma pequena parcela dos usuários da internet, que ainda sequer constituem a totalidade da população. Considerando dados de 2023, cerca de 88% da população brasileira tinha acesso à internet [24], i.e., aproximadamente 178 milhões de pessoas, de modo que menos de 1% dos usuários de internet brasileiros foram de fato ouvidos na maior experiência de participação social online do Brasil. Afinal, como enfatizado por Gomes (2005) [2]: “[...] talvez nem toda a debilidade de participação política contemporânea se explique em termos de dificuldade de acesso, raridade de meios e escassez de oportunidades. A abundância de meios e chances não formará, per se, uma cultura da participação política.”.

Por outro lado, redes sociais já são uma forma de comunicação bem estabelecida no mundo, contando com cerca de 5 bilhões de contas ativos globalmente no começo de 2024 [25], das quais 144 milhões são brasileiras [26], o que corresponde em torno de 81%

¹Disponível em: <https://pol.is/>. Acesso em: 25 de março de 2024.

²Disponível em: <https://info.vtaiwan.tw/>. Acesso em: 25 de março de 2024.

³Disponível em: <https://brasilparticipativo.presidencia.gov.br/>. Acesso em: 25 de março de 2024.

⁴Disponível em: http://planalto.gov.br/ccivil.03/_ato2023-2026/2024/lei/L14802.htm. Acesso em: 1 abril de 2024.

de todos os usuários de internet do país. Apesar das maiores aplicações das mídias sociais não serem voltadas necessariamente para a ação política [3], o seu impacto neste setor já é extensivamente documentado pelo mundo, como por exemplo: durante a Primavera Árabe [4], nas Jornadas de Junho de 2013 [5], no *Euromaidan* [6] e no movimento *Black Lives Matter* [7].

1.1 Definição do Problema

As ferramentas de participação social na política ainda não conquistaram a adesão de uma porção significativa dos utilizadores de internet, e as redes sociais existentes, apesar de populares, ainda são ferramentas limitadas quando o assunto é democracia e prática política. Já existem diversas pesquisas que apontam os efeitos de polarização [5, 8] e de agrupamento de indivíduos com ideias semelhantes [9], fenômeno também conhecido como *echo chambers*, nas redes mais utilizadas. Estes fatos se relacionam profundamente com a falta de transparência algorítmica, por parte das empresas que desenvolvem tais plataformas, de maneira que conteúdos são fornecidos aos usuários com base em critérios não claros, ou sequer divulgados.

1.2 Premissas e Hipóteses

Levando em conta a cultura já estabelecida de uso das redes sociais, da atenção crescente que o tema da democracia digital vem recebendo no meio acadêmico [10, 11] e da crescente, mas ainda tímida, quantidade de aplicações que buscam incrementar o papel da participação popular no governo, observamos uma possibilidade de contribuir para o tema.

A hipótese central para o desenvolvimento desta aplicação baseia-se no sucesso do uso de Sistemas de Controle de Versão (SCV), particularmente o *Git*. O *Git* é mais comumente utilizado no contexto de desenvolvimento de software, principalmente de maneira colaborativa, entretanto, essa não é a única aplicação para o *Git*, sendo o mesmo já usado como uma ferramenta para auxiliar na produção de conhecimento científico [19], mostrando que há uma maior gama de aplicações possíveis com o uso do sistema. Assim, visamos aplicar conceitos de SCV para a construção coletiva de ideias, de forma a potencializar a capacidade de indivíduos de uma certa comunidade chegarem à soluções mais bem desenvolvidas e com participação de todos os seus membros.

1.3 Objetivo geral

Temos como objetivo projetar e implementar uma rede social online de código aberto, acessível por computadores e dispositivos móveis, que facilite e aprimore o processo de participação de integrantes de uma dada comunidade em seus processos de decisão.

1.4 Objetivos específicos

- Implementar um banco de dados integrando *Git* e *PostgreSQL*;
- Desenvolver uma interface de fácil compreensão e com soluções bem estabelecidas;
- Fazer um comparativo com plataformas semelhantes;
- Disponibilizar o código fonte em um repositório no *Github*;

1.5 Estrutura do relatório técnico

O presente relatório está dividido em 5 capítulos. No primeiro, é realizada a introdução ao tema e a definição do problema e dos objetivos do trabalho. No segundo, são expostos os principais conceitos e tecnologias utilizados para a construção da solução. No terceiro, são desenvolvidos os requisitos da aplicação proposta e a sua arquitetura. No quarto, são apresentados o plano de testes, um comparativo com plataformas semelhantes e uma análise sobre os resultados. No quinto e último capítulo são disponibilizadas as conclusões obtidas sobre o trabalho, bem como uma lista de sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os principais conceitos de tecnologia utilizadas para construir a aplicação, além de especificar as ferramentas escolhidas.

2.1 Conceitos Gerais

Nesta seção, estão especificados os principais conceitos em que se fundamentam a solução proposta por nós.

2.1.1 *Front-end*

O *front-end* se refere à parte da aplicação que os usuários interagem diretamente, i.e., a interface e seus elementos visuais [13]. Atua juntamente com o *back-end*, a partir da realização de requisições de dados para os exibir em sua interface. É normalmente desenvolvido com o uso de: *Hypertext Markup Language* (HTML) para a estruturação, *Cascading Style Sheets* (CSS) para a estilização e *JavaScript* (JS) para a interatividade.

2.1.2 *Back-end*

O *back-end* se refere à parte da aplicação que é responsável por fazer o processamento, armazenamento e fornecimento dos dados, dados estes que são frequentemente gerados pelos usuários e são servidos ao *front-end*, comumente através de mensagens escritas em *JavaScript Object Notation* (JSON). Geralmente é composto por três partes [13]: o servidor, que é o *hardware* em si; a aplicação, que lida com as requisições do *front-end*, recebendo, processando e enviando dados; e o banco de dados, que será discutido no próximo tópico.

2.1.3 Banco de dados

O banco de dados trata-se essencialmente da associação entre uma coleção de dados, também chamada de base de dados, e um sistema gerenciador de banco de dados (SGBD) [14], o qual é responsável por ofertar funcionalidades de manipulação sobre tal coleção, como operações de criação, leitura, atualização, deleção e manutenção. Os bancos de dados podem ser de diferentes tipos, dentre eles, os mais comuns são:

- **Bancos de dados relacionais:** Os dados são armazenados em tabelas, as quais podem ser relacionadas umas as outras. A maioria dos SGBD que lidam com bancos de dados relacionais utiliza *Structured Query Language* (SQL) para operar a manipulação dos dados;

- **Bancos de dados não relacionais:** Os dados são armazenados em qualquer outro formato, e por isto acabam por ser mais flexíveis. Podem ser usados para armazenar imagens, vídeos, documentos e outros tipos de conteúdos.

2.1.4 Sistema de controle de versões

Um sistema de controle de versões (SCV) é um sistema que gerencia o desenvolvimento de um objeto em evolução [15]. É uma ferramenta muito utilizada no ramo de desenvolvimento de software, pois ele fornece as funcionalidades de rastreamento de alterações em arquivos e a possibilidade de gerenciar diferentes versões de um projeto, além de outros utilitários que facilitam a colaboração entre várias pessoas em um projeto. Nós faremos uso de SCV neste trabalho como um SGBD para um banco de dados local. Há dois conceitos importantes que foram introduzidos por estes sistemas que usaremos em nossa proposta:

- **Branching:** Consiste em criar uma nova “ramificação” do projeto a partir de uma certa versão, assim permitindo novas versões serem criadas sem causar alterações no “ramo” original, o que facilita o trabalho de várias pessoas em paralelo no mesmo projeto sem causar conflitos;
- **Merging:** É o ato de incorporar as mudanças de um “ramo” em outro, o que facilita que a colaboração entre diferentes pessoas envolvidas no projeto.

2.2 Tecnologias utilizadas

Nesta seção, estão especificadas as tecnologias utilizadas para o desenvolvimento de nossa solução, separadas de acordo com a área conceitual em que foi utilizada.

2.2.1 Gerais

- **Node.js**⁵: O *Node* é um ambiente de código aberto para execução de JS. A partir dele, é possível utilizar o JS fora do navegador, o que abre a possibilidade de se desenvolver diversas aplicações. Em nosso caso, o utilizamos para executar tanto o *front-end*, como o *back-end*.

2.2.2 Front-end

- **JavaScript**⁶: O JS é uma linguagem de programação interpretada de tipagem fraca. Utilizada amplamente no cenário de desenvolvimento Web, principalmente

⁵Website oficial disponível em: <https://nodejs.org/>. Acesso em: 1 de abril de 2024

⁶Documentação disponível em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. Acesso em 8 de abril de 2024.

na construção de páginas interativas, também é usada para construir aplicações fora do navegador, como citado no tópico do *Node.js*. Faremos uso do JS para implementar a lógica de funcionamento da interface;

- **Vite**⁷: É um pacote de ferramentas para desenvolvimento de aplicações *front-end*. O *Vite* inclui comandos para criação de um projeto pré-configurado em vários *frameworks* diferentes, scripts de inicialização e para o processo de *build*, suporte à pré-processadores de CSS, além de fornecer a funcionalidade de *Hot Module Replacement*, que possibilita o recarregamento da aplicação à medida que alterações no código são feitas, facilitando o processo de desenvolvimento;
- **React**⁸: O *React* é uma biblioteca *front-end* para JS de código aberto desenvolvida pela *Meta*, então *Facebook*. Com o *React* é possível construir interfaces de usuário, a partir de partes chamadas “componentes”, que podem ser reutilizadas várias vezes pela aplicação, a fim de facilitar a codificação da aplicação. O *React* também introduz uma nova extensão de arquivo, o *JavaScript XML* (JSX), que permite a combinação de HTML e JS em um único arquivo;
- **Sass**⁹: O *Syntactically Awesome Style Sheets* (Sass) é uma linguagem de extensão e pré-processador do CSS e é de código aberto. O *Sass* estende o CSS ao ofertar alguns mecanismos disponíveis em linguagens de programação convencionais, como variáveis e operadores, e outras ferramentas como o aninhamento. Desta forma, o *Sass* acaba por acelerar o processo de escrita das regras de estilo da aplicação, além de conferir uma melhor legibilidade ao código.

2.2.3 Back-end

- **TypeScript**¹⁰: O *TypeScript* é uma linguagem de programação de código aberto desenvolvida pela Microsoft. Esta linguagem é um superconjunto do JS, i.e., adiciona funcionalidades às já existentes do JS. Como indica seu nome, é de tipagem forte, o que possibilita uma maior previsibilidade sobre o funcionamento do código, de modo a facilitar a detecção de possíveis erros;
- **Express**¹¹: É um *framework* minimalista para *Node.js* que possibilita a criação de aplicações que gerenciam requisições em servidores. Através do *Express*, é possível criar rotas que ofertam funcionalidades em certos *Uniform Resource Locators* (URLs) para os principais métodos HTTP, sendo alguns deles: GET, para a solicitação de

⁷Website oficial disponível em: <https://vitejs.dev/>. Acesso em: 1 de abril de 2024

⁸Website oficial disponível em: <https://react.dev/>. Acesso em: 1 de abril de 2024

⁹Website oficial disponível em: <https://sass-lang.com/>. Acesso em: 1 de abril de 2024

¹⁰Website oficial disponível em: <https://typescriptlang.org/>. Acesso em: 1 de abril de 2024

¹¹Website oficial disponível em: <https://expressjs.com/>. Acesso em: 1 de abril de 2024

recursos; POST, para submissão de recursos; PATCH, para aplicação de alterações em um recurso; e DELETE, para a deleção de um recurso.

2.2.4 Banco de dados

- ***PostgreSQL***¹²: O *PostgreSQL* é um SGBD de código aberto para bancos de dados relacionais. O utilizamos para criar as tabelas que irão armazenar os dados dos usuários, das interações entre os usuários e alguns metadados dos conteúdos gerados;
- ***Git***¹³: Trata-se de um dos SCV mais utilizados no desenvolvimento de software. Teve seu desenvolvimento iniciado por Linus Torvalds em 2005 para ser o novo SVC do código fonte do kernel *Linux* [12]. Nós faremos uso do *Git* como um SGBD, em que o banco de dados serão documentos que representam os posts. Ao utilizar o *Git* para armazenar os *posts*, conseguimos manter um histórico de alterações de cada *post*, além de nos possibilitar realizar as operações de *branching* e *merging*, para fazer o controle de alterações ente vários usuários em um mesmo *post*.

2.3 Atores do processo

Os únicos atuantes no processo são os próprios membros da comunidade que optarem por participar na plataforma. Nesta plataforma, não há moderadores nem administradores, o conteúdo é gerido totalmente pelos próprios usuários.

¹²Website oficial disponível em: <https://postgresql.org/>. Acesso em: 1 de abril de 2024

¹³Documentação disponível em: <https://git-scm.com/>. Acesso em 8 de abril de 2024

3 METODOLOGIA

3.1 Visão Geral

A plataforma colcom se propõe a ser um ambiente de debates justo, transparente, participativo e deliberativo. O nome colcom é um acrônimo para “colaboração e competição”, que são as principais formas de atuação na plataforma. A lógica de operação da plataforma pode ser resumizada nos seguintes passos:

- Um usuário inicia um tópico;
- Usuários submetem *posts* em resposta ao tópico;
- Usuários colaboram no aprimoramento de *posts*, através de sugestões de edição e críticas das falhas encontradas;
- Usuários competem para que sua resposta seja a mais votada do tópico.

Além disso, a plataforma possibilita aos usuários avaliar a relevância percebida de cada conteúdo, seja ele um tópico, *post* ou crítica, e também confere aos usuários a possibilidade de promoverem um tópico, aumentando assim a sua visibilidade no dia corrente. Optamos por permitir apenas a promoção de tópicos a fim de evitar efeitos de polarização, dado que são os tópicos que agregam todas as respostas dos usuários.

3.2 Requisitos Funcionais

Nesta seção serão descritos os requisitos funcionais da aplicação, que se referem às declarações de serviços que o sistema deve fornecer, especificando inclusive as condições sobre as quais devem ser fornecidos [16].

3.2.1 [RF01] Cadastrar um usuário

A aplicação deve possibilitar o cadastro de novos usuários por meio do fornecimento de um nome de usuário, e-mail e senha.

3.2.2 [RF02] Efetuar o *login* de um usuário

A aplicação deve possibilitar que usuários previamente cadastrados consigam acessar sua conta ao fornecer seu nome de usuário ou e-mail e sua senha.

3.2.3 [RF03] Avaliar um conteúdo como relevante ou não

Usuários que estejam logados devem ser capazes de avaliar se um conteúdo como relevante ou não por meio de botões de voto.

3.2.4 [RF04] Salvar um conteúdo

Usuários que estejam logados devem ser capazes de salvar um conteúdo em sua coleção de favoritos.

3.2.5 [RF05] Criar um tópico

Usuários que estejam logados devem ser capazes de criar novos tópicos através de um botão. O botão ao ser acionado deve abrir uma caixa de diálogo para a inserção dos dados do tópico, que incluem: título, possibilidade de usuários submeterem múltiplas respostas, e as possíveis respostas, caso o tópico se trate de uma enquete.

3.2.6 [RF06] Promover um tópico

Usuários que estejam logados devem ser capazes de promover um tópico, i.e., conferir um voto de visibilidade para o tópico até o fim do dia corrente.

3.2.7 [RF07] Visualizar tópicos registrados

Todos os usuários devem ser capazes de visualizar os tópicos registrados, ordenados pelo número de promoções, ou data de criação.

3.2.8 [RF08] Visualizar tópicos salvos

Usuários que estejam logados devem ser capazes de visualizar os tópicos salvos anteriormente, ordenados pela data de salvamento.

3.2.9 [RF09] Escrever corpo de um conteúdo

A aplicação deve fornecer um editor de texto com várias funcionalidades e capacidades de estilização para viabilizar o uso da plataforma para diversos contextos diferentes.

3.2.10 [RF10] Criar um *post* em resposta a um tópico

Usuários que estejam logados devem ser capazes de responder um dado tópico através da criação de um *post*. Ao criar o *post*, o usuário deve fornecer: um título, um corpo do *post* e, caso o tópico seja uma enquete, uma opção de resposta.

3.2.11 [RF11] Votar em um *post*

Usuários que estejam logados devem ser capazes de votar em um *post* de um tópico para representar sua escolha.

3.2.12 [RF12] Editar um *post*

O autor de um *post* deve ser capaz de realizar edições em um *post* e salvá-las, criando assim uma nova versão do *post*. Uma mensagem contendo uma explicação das alterações deve ser fornecida pelo usuário.

3.2.13 [RF13] Criar uma sugestão de edição para um *post*

Usuários logados que não sejam o autor de um *post* devem ser capazes de submeter uma sugestão de alteração para o *post*. Uma mensagem contendo uma explicação das alterações deve ser fornecida pelo usuário.

3.2.14 [RF14] Criar um “ramo” de uma versão específica de um *post*

Usuários logados que não sejam o autor de um *post* devem ser capazes de fazer uma *branch* de um *post* em qualquer versão, i.e., criar uma cópia do *post* em que a autoria passa a ser do usuário solicitante.

3.2.15 [RF15] Visualizar uma versão específica de um *post*

Usuários que estejam logados devem ser capazes de visualizar uma versão específica de um *post* através da manipulação de uma linha do tempo que exiba todas versões da aplicação.

3.2.16 [RF16] Visualizar proposições de edição de um *post*

O autor de um *post* deve ser capaz de visualizar todas as proposições de edição feitas por outros usuários para este *post*.

3.2.17 [RF17] Acatar e rejeitar proposições de edição de um *post*

O autor de um *post* deve ser capaz de rejeitar ou incorporar proposições, através de um *merge*, em um *post* de sua autoria.

3.2.18 [RF18] Visualizar críticas de uma versão específica de um *post*

A aplicação deve possibilitar que todas as críticas contidas em cada trecho, de cada versão de um *post*, sejam visualizáveis para todos os usuários da aplicação.

3.2.19 [RF19] Realizar críticas em um *post*

Usuários que estejam logados devem ser capazes de criticar qualquer trecho de um *post*, fornecendo um título e corpo para a crítica.

3.3 Requisitos Não Funcionais

Nesta sessão, estão descritos os requisitos não funcionais. Contrariamente aos requisitos funcionais, não há um consenso bem estabelecido sobre o que são, havendo diversas definições na literatura [16, 17]. Aqui, usaremos a definição comum de que são requisitos que se referem a outros aspectos qualitativos que não funcionalidades em si. Para nosso projeto, separamos os requisitos não funcionais de acordo com os seguintes aspectos: usabilidade, desempenho, segurança, manutenibilidade, portabilidade e legalidade.

3.3.1 Usabilidade

3.3.1.1 [RNF01] Intuitividade

A aplicação deve ter um design intuitivo, aproveitando-se de símbolos comumente já usados em redes sociais populares.

3.3.1.2 [RNF02] Responsividade

A aplicação deverá ter uma boa adaptação à diferentes resoluções de tela, de modo que funcione bem em dispositivos móveis e em computadores pessoais.

3.3.1.3 [RNF03] Consistência

A aplicação deve ter um design consistente, de modo que usuários possam inferir o funcionamento de outras partes da aplicação a partir do que já foi visto.

3.3.2 Desempenho

3.3.2.1 [RNF04] Usuários simultâneos

A aplicação deve suportar o uso de mais de 100 usuários simultâneos.

3.3.2.2 [RNF05] Tempo de resposta

A aplicação deve demorar não mais que 1 segundo para retornar uma resposta.

3.3.3 Segurança

3.3.3.1 [RNF06] Controle de acesso

A aplicação deve garantir que somente usuários autorizados publiquem e editem conteúdos.

3.3.3.2 [RNF07] Obfuscação de dados sensíveis

A aplicação deve criptografar os dados sensíveis a serem armazenados.

3.3.4 Manutenibilidade

3.3.4.1 [RNF08] SCV

O código fonte deve ser escrito fazendo-se uso de um SCV para registrar as alterações realizadas.

3.3.4.2 [RNF09] Código aberto

O código fonte deve ser disponibilizado gratuitamente na plataforma *GitHub*.

3.3.5 Portabilidade

3.3.5.1 [RNF10] *Self-hosting*

A aplicação deve ser auto-contida, de modo a permitir que qualquer um seja capaz de hospedar uma instância do colcom.

3.3.5.2 [RNF11] Compatibilidade

A aplicação deverá funcionar nos navegadores *Chrome* versão 66+, *Edge* versão 16+, *Firefox* versão 57+, *Opera* versão 53+ e *Safari* versão 12.1+.

3.3.6 Legalidade

3.3.6.1 [RNF12] Licença de Software

O código fonte deve ser disponibilizado sobre a Licença Pública Geral Affero GNU versão 3 (AGPLv3).

3.3.6.2 [RNF13] Lei Geral de Proteção de Dados Pessoais (LGPD)

A aplicação deve ser desenvolvida em conformidade com a LGPD.

3.4 Arquitetura

Em sua arquitetura, o colcom segue a Figura 1. A aplicação *front-end*, desenvolvida com *React*, recebe diretamente as interações do usuário e faz as requisições necessárias para o *back-end*, através dos métodos *HTTP* e mensagens em codificadas em *JSON*. A aplicação *Express* do *back-end*, ofertando diversas rotas com funcionalidades específicas, recebe tais requisições e faz o devido processamento correspondente à rota requisitada, realizando as operações de manipulação necessárias sobre o banco de dados *PostgreSQL* e *Git*, e, por fim, recebe os dados provenientes do banco de dados e os envia, em *JSON*, de volta para a aplicação *React*.

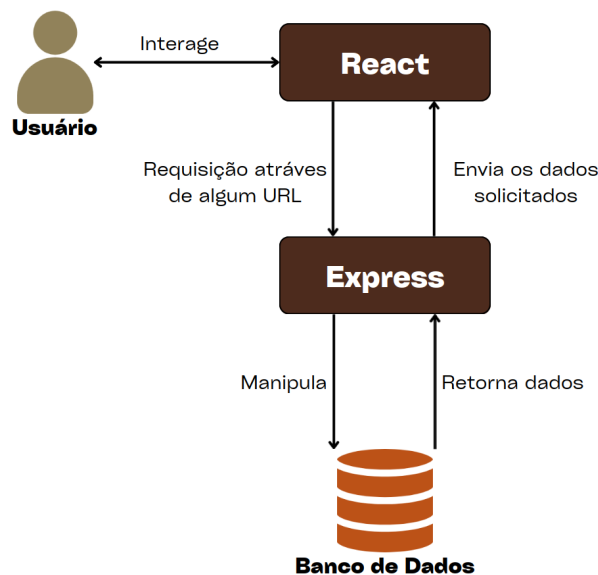


Figura 1: Arquitetura da aplicação.

Fonte: própria do autor

3.4.1 Banco de dados

O banco de dados foi implementado baseando-se no diagrama entidade-relacionamento da Figura 2.

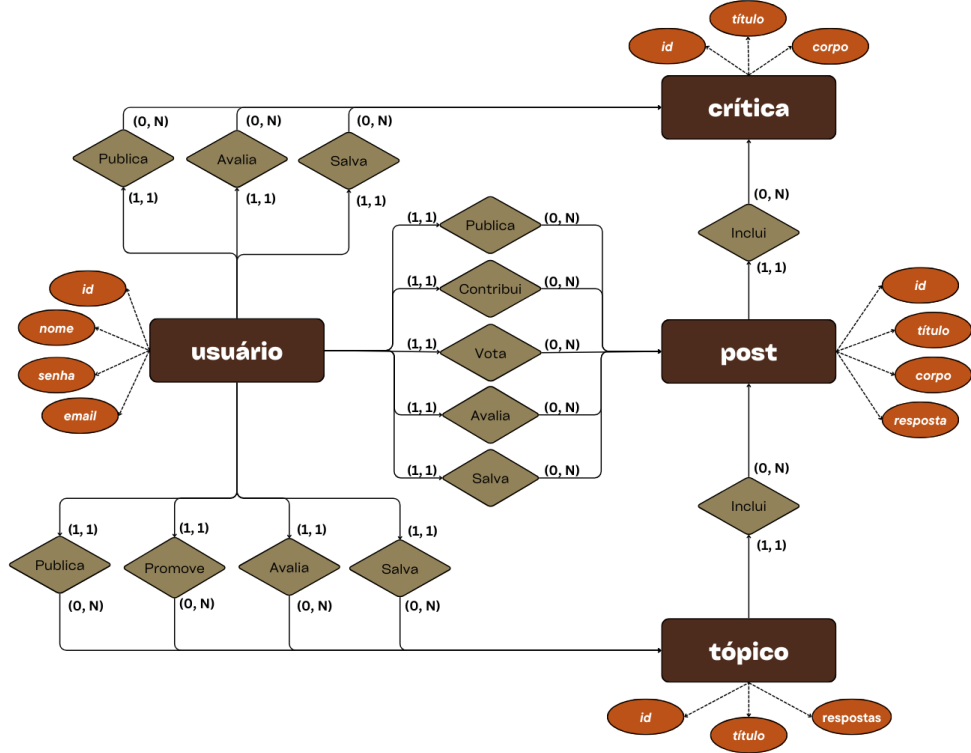


Figura 2: Diagrama entidade-relacionamento da aplicação.

Fonte: própria do autor

As entidades *tópico*, *post* e *crítica* foram unificados em uma tabela de conteúdos *contents*, onde a relação entre os conteúdos se dá através de um campo *parent_id*, que denota o identificador (*id*) do conteúdo “pai”. Para as relações de avaliação, voto, promoção, contribuição e salvamento foram todas consideradas em uma tabela de interações *interactions*, na qual o tipo da interação é especificado por uma coluna *type*.

Todos os dados da aplicação foram armazenados utilizando o *PostgreSQL*, com exceção dos dados de corpo dos *posts*, que foram armazenados em arquivos locais gerenciados pelo *Git*, como um SGBD. Cada *tópico* criado corresponde a uma pasta que contém um repositório *Git*, a partir do qual, para cada *post* criado, cria-se uma nova *branch* correspondente.

3.5 Interface da aplicação

A seguir, seguem imagens do design desenvolvido para a aplicação. Como o design depende da existência de conteúdos publicados, fizemos uso de dados *mock*, i.e., dados simulados, para o preenchimento dos campos. Para a paleta de cores utilizada no design da aplicação, foram escolhidas cores visando minimizar a emissão de luz azul, dado que existem evidências que apontam que a luz azul artificial pode prejudicar o sono [21], apesar de que existem outros estudos que ainda apontam inconclusividade em relação a este ponto e outros supostos malefícios da luz azul artificial [20].

3.5.1 Principais componentes

Nesta seção, iremos especificar os principais componentes implementados para compor a interface da aplicação.

3.5.1.1 *Frame*

Um *frame*, ou quadro (Figura 3), é uma unidade básica de composição, em que outros componentes serão montados em cima. Ele fornece:

- **Botões de voto e avaliação:** São compostos pelas setas e o círculo antes do título, as setas correspondem aos botões de avaliação de relevância, com a seta para cima indicando um conteúdo relevante e, para baixo, um conteúdo não relevante. O botão de voto pode ser usado para múltiplas finalidades que serão descritas nas seções dos próximos componentes;
- **Campo de botões utilitários:** Representado pelos três pontos no canto superior direito, podem ser fornecidos botões com texto e ícones que realizam alguma função quando clicados. Vale ressaltar que no design para *desktop* os ícones são mostrados diretamente, enquanto para *mobile* somente são visíveis ao clicar no ícone de três pontos, abrindo um menu;
- **Campo de corpo:** Pode ser texto ou outros componentes;
- **Campo de rodapé:** Pode ter texto ou uma lista de itens;
- **Campo de título.**

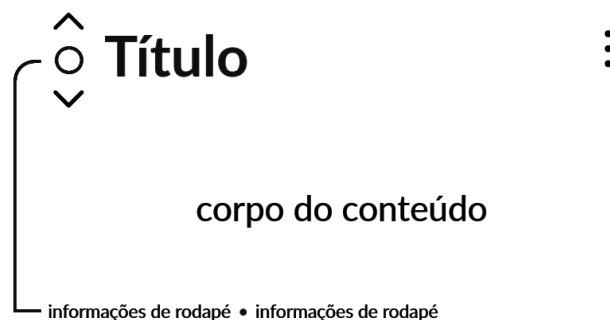


Figura 3: Estrutura do *frame*.
Fonte: própria do autor

3.5.1.2 Tópico

O tópico contém os seus *posts* de resposta dispostos em ordem decrescente de votos. Os *posts* de cada tópico são expostos em duas partes (Figura 4): uma barra com o título do *post* de tamanho proporcional à sua porcentagem de votos e os primeiros 280 caracteres do texto do *post*. *Posts* escritos pelo usuário tem um símbolo de um lápis antes de seu título, e o *post* votado pelo usuário contém um símbolo de uma estrela e uma cor de fundo diferente para a barra. O usuário pode promover o tópico ao clicar no botão de voto, além disso, pode salvar o tópico ao clicar no símbolo de *bookmark* e responder o tópico criando um *post*, ao clicar na seta apontada para esquerda.

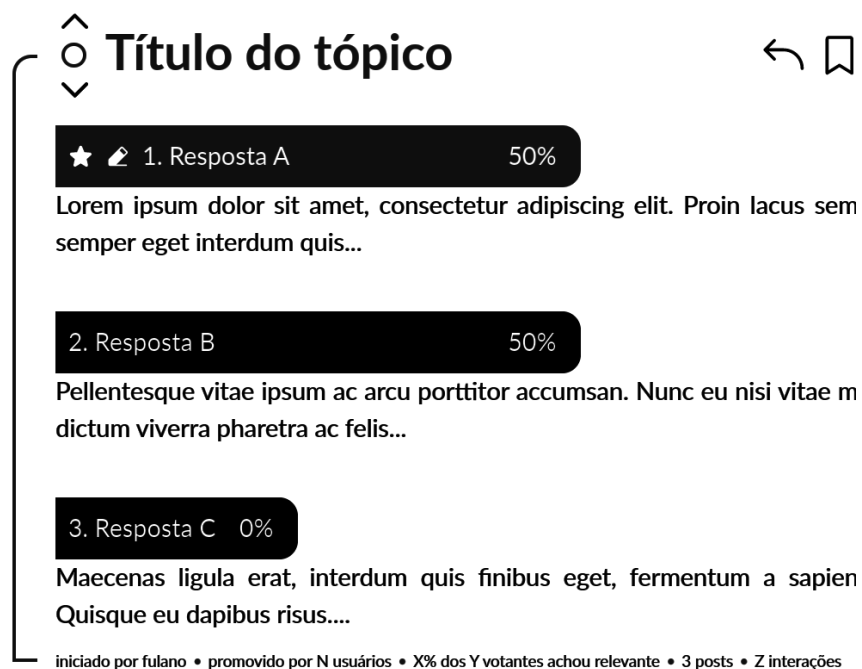


Figura 4: Estrutura do tópico.

Fonte: própria do autor

3.5.1.3 Post

O *post* pode ter, como parte de seu corpo, texto, com as principais funcionalidades de formatação do *Markdown*, além de gráficos e tabelas. O seu botão de voto se destina a realizar o voto no *post corrente* como resposta ao tópico, já os botões utilitários têm as seguintes funções, seguindo da esquerda para a direita:

- **Criar uma *branch*:** Atende ao [RF14], somente visível para usuários que não sejam o autor do *post*;
- **Realizar *merge* de uma *branch*:** Atende ao [RF17], somente visível para o autor do *post*. Há um contador indicando quantas sugestões foram recebidas;

- Ocultar/expor críticas;
- **Editar o conteúdo do *post*:** Permite o usuário editar o conteúdo do *post*, quando o usuário terminar, ele pode clicar novamente no ícone e então uma caixa de diálogo será aberta para que ele escolha o que fazer com suas edições. Caso seja o autor, atualiza diretamente o *post*, atendendo ao [RF12], caso não, cria uma nova sugestão de *post*, atendendo ao [RF13];
- Salvar o *post*.

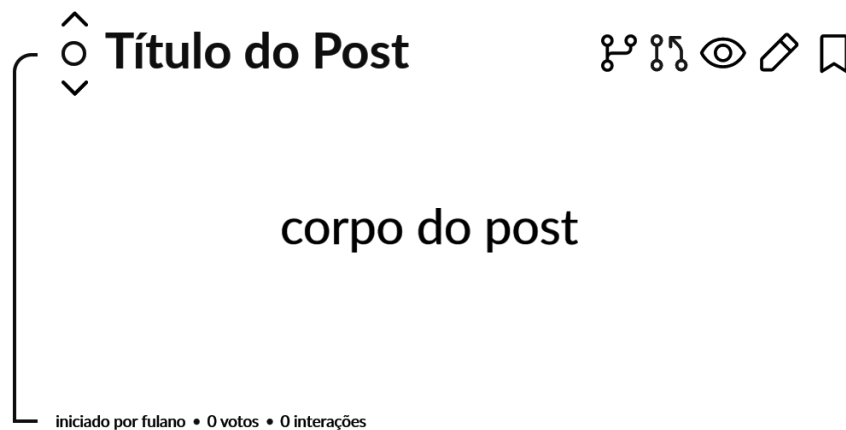


Figura 5: Estrutura do post.

Fonte: própria do autor

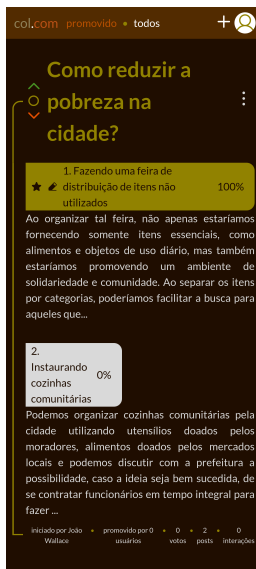
3.5.1.4 Crítica

A crítica não contém um botão de voto, e só tem um botão para salvá-la e um botão para fechá-la. Em seu rodapé consta o nome do autor e a quantidade e divisão dos votos de avaliação de relevância.

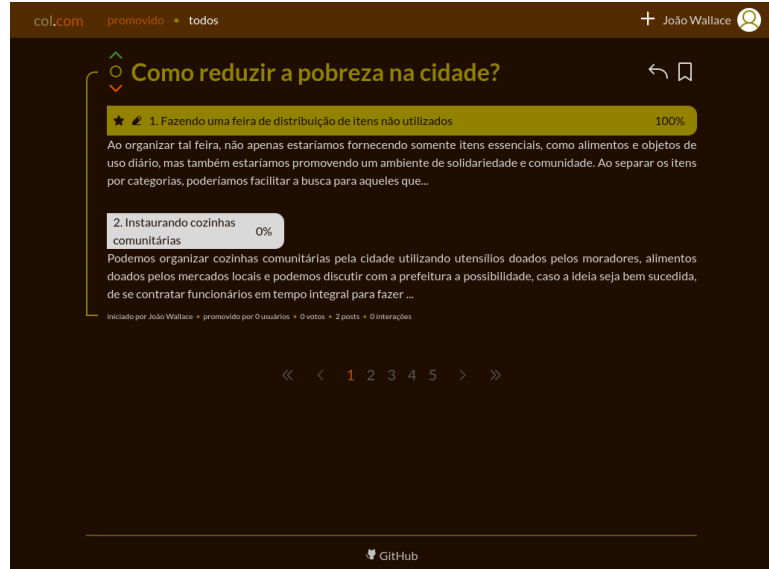
3.5.2 Páginas

3.5.2.1 Página de árvore de tópicos

A tela de visualização de tópicos (Figura 6) é a tela principal da aplicação e contém até 5 tópicos, sendo o restante acessível por meio do uso de um componente de paginação. Cada tópico exibe os 3 *posts* mais votados, podendo o restante ser visualizado ao entrar na página do tópico através de um clique no título do tópico, ou em reticências que aparecem em seu canto inferior, caso hajam mais de 3 *posts*. Por fim, o conteúdo pode ser ordenado por número de promoções, na aba “promovido” e por ordem de criação, na aba “todos”.



(a) Versão mobile



(b) Versão desktop

Figura 6: Tela de visualização dos tópicos.

3.5.2.2 Página de tópico

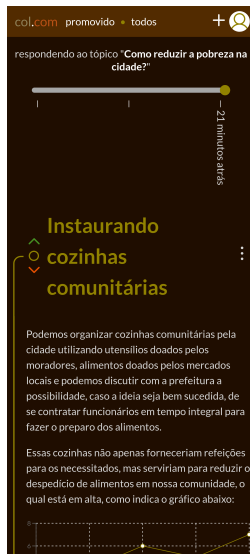
A página de tópico é semelhante à página de árvore de tópicos, entretanto, só contém um componente de tópico e todos os seus *posts* associados.

3.5.2.3 Página de *post*

Como exposto na Figura 7, a tela de um *post* é composta por três elementos, sendo eles de cima para baixo: um *link* para seu tópico pai, uma barra *slider*, para seleção da versão e o *post* em si. Através da tela de *post*, também é possível visualizar as críticas realizadas ao *post* em cada uma de suas versões, como demonstrado na Figura 8. Críticas feitas em trechos que se sobrepõem são unidas em um mesmo bloco que, ao ser clicado, exhibe a lista de todas as críticas contidas nele; para se identificar o trecho específico de uma crítica contida em um bloco, o usuário pode clicar em seu título, assim marcando o trecho condizente.

3.5.2.4 Página de *login* e cadastro

O *login* e cadastro do usuário são realizados na mesma página, esta exposta na Figura 9, sendo possível alternar entre duas operações ao clicar no texto “crie uma agora!”, para se direcionar ao cadastro, e “entre agora!” para o *login*. O formulário de cadastro é exposto na Figura 10.

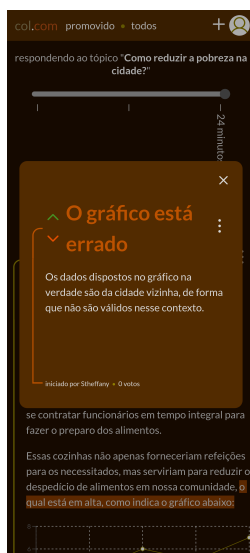


(a) Versão mobile



(b) Versão desktop

Figura 7: Tela de visualização do *post*.



(a) Versão mobile



(b) Versão desktop

Figura 8: Tela de um *post* com uma crítica sendo visualizada.

3.5.2.5 Página de conteúdos salvos

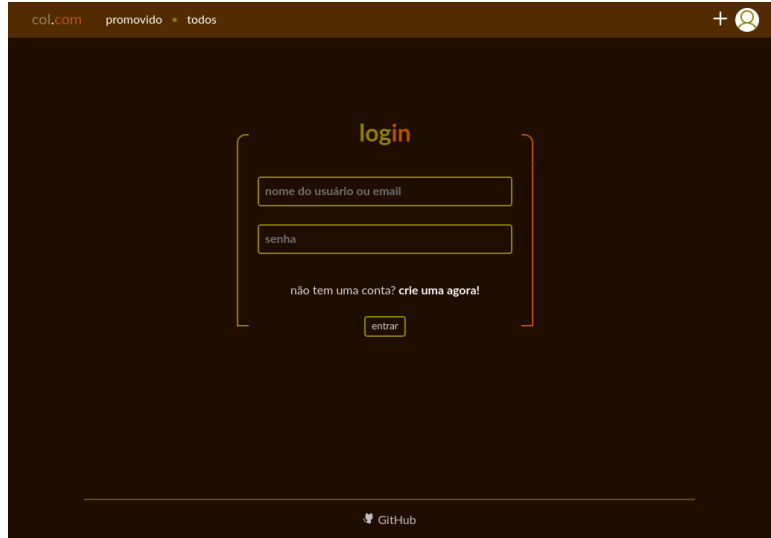
A página de conteúdos salvos contém todos os tópicos, *posts* e críticas salvos pelo usuário, dispostos de maneira similar à árvore de tópicos.

3.5.2.6 Página de edição de texto

Na página de edição de texto, o usuário pode escrever um *post*, informando um título e fornecendo um corpo. O corpo do *post* pode conter texto, com estilizações do *Markdown*, tabelas e gráficos.



(a) Versão mobile



(b) Versão desktop

Figura 9: Tela de *login* e cadastro.

Figura 10: Formulário de cadastro.

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Este capítulo é destinado para a exposição dos resultados, os quais foram obtidos através do plano de testes, detalhado a seguir.

4.1 Plano de testes

O objetivo do plano de testes é permitir a identificação de eventuais problemas de implementação na aplicação, de modo a assegurar o funcionamento esperado da plataforma. Para tal, será realizada uma navegação por todas as principais funcionalidades da aplicação, partindo do cadastro de usuário até a publicação de conteúdos.

Para efetuar tais testes, tanto a aplicação *front-end* como a *back-end* foram hospedadas localmente em uma máquina dedicada com *Ubuntu Server 22.04 LTS*. A partir desta, o sistema foi acessado de diferentes dispositivos, como *desktops*, *notebooks* e *smartphones*, e usou-se diferentes navegadores, como *Google Chrome*, *Firefox* e *Safari*, a fim de garantir a disposição correta da aplicação em diferentes combinações de tecnologias.

Infelizmente, os testes foram conduzidos exclusivamente pelo autor do presente trabalho, dado que não foi possível realizar testes com usuários reais antes da data de apresentação do mesmo.

4.2 Casos de teste

A seguir, Nas Tabelas 1-13, serão descritos os casos de teste considerados de maior importância para o fluxo principal de funcionamento da plataforma.

Título	[T01] Cadastrar usuário
Descrição	Testa-se o cadastro de um usuário não registrado. Refere-se ao [RF01].
Passos do teste	O usuário preenche os campos de nome de usuário, email e senha e clica em “cadastrar”.
Resultado esperado	O cadastro do usuário deve ser criado no banco de dados, e o usuário dever ser redirecionado para a tela de login.
Resultado obtido	Sucesso.

Tabela 1: Caso de teste [T01].

Título	[T02] Logar usuário
Descrição	Testa-se a realização do <i>login</i> de um usuário. Refere-se ao [RF02].
Passos do teste	O usuário preenche os campos de nome de usuário/email e senha e clica em “entrar”.
Resultado esperado	O usuário deve ser redirecionado para a tela de “promovido” com seus dados carregados.
Resultado obtido	Sucesso.

Tabela 2: Caso de teste [T02].

Título	[T03] Criar um tópico
Descrição	Testa-se a criação de um tópico. Refere-se ao [RF05]
Passos do teste	O usuário clica no botão com um “+” na barra de navegação, abrindo um modal, o usuário então preenche os campos de título, marca se deseja que outros usuários possam submeter múltiplos posts e descreve as possíveis respostas, caso já as tenha previamente.
Resultado esperado	O tópico é criado no banco de dados e o usuário é redirecionado para a página do tópico.
Resultado obtido	Sucesso.

Tabela 3: Caso de teste [T03].

Título	[T04] Avaliar um tópico como relevante
Descrição	Testa-se os botões de avaliação “relevante” e “não relevante” em um tópico. Refere-se ao [RF03].
Passos do teste	O usuário clica nas setas de avaliação de um tópico, primeiro para definir como “não relevante” e depois para “relevante”.
Resultado esperado	As setas aparecem preenchidas quando o usuário as clica, gerando um registro de interação correspondente no banco de dados.
Resultado obtido	Sucesso.

Tabela 4: Caso de teste [T04].

Título	[T05] Salvar um tópico
Descrição	Testa-se o botão de salvar de um tópico. Refere-se ao [RF04].
Passos do teste	O usuário clica no botão de salvar de um tópico.
Resultado esperado	O ícone do botão se torna preenchido e uma interação de salvamento é criada no banco de dados.
Resultado obtido	Sucesso.

Tabela 5: Caso de teste [T05].

Título	[T06] Promover um tópico
Descrição	Testa-se o botão de promoção de um tópico. Refere-se ao [RF06].
Passos do teste	O usuário clica no botão de promover de um tópico.
Resultado esperado	O ícone do botão de promover se torna preenchido e uma interação de promoção vigente até o fim do dia é criada no <i>back-end</i> .
Resultado obtido	Sucesso.

Tabela 6: Caso de teste [T06].

Título	[T07] Criar um <i>post</i> em resposta a um tópico
Descrição	Testa-se o fluxo de criação de um <i>post</i> em resposta a um tópico. Refere-se aos [RF09] e [RF10].
Passos do teste	O usuário clica no botão de responder de um tópico, e, após ser redirecionado à página de escrita, preenche o campo de título e resposta, caso haja opções de resposta, e digita um texto contendo todas as funcionalidades do editor de texto, e então clica no botão “publicar”.
Resultado esperado	O <i>post</i> é criado no <i>back-end</i> e o usuário é redirecionado para a página do textitpost.
Resultado obtido	Sucesso.

Tabela 7: Caso de teste [T07].

Título	[T08] Editar um <i>post</i>
Descrição	Testa-se a edição de um <i>post</i> . Refere-se ao [RF12].
Passos do teste	O usuário clica no botão de edição de um tópico, então faz alterações no texto do <i>post</i> através do editor de texto, e após de finalizado clica novamente no símbolo de edição, abrindo um modal, por fim o usuário preenche o campo de resumo das alterações e clica no botão “enviar”.
Resultado esperado	O <i>slider</i> de versões registra a nova versão e a edição é enviada para o <i>back-end</i> , que cria um novo <i>commit</i> no banco de dados.
Resultado obtido	Sucesso.

Tabela 8: Caso de teste [T08].

Título	[T09] Criar uma sugestão de edição para um <i>post</i>
Descrição	Testa-se a criação de uma sugestão de edição para um <i>post</i> . Refere-se ao [RF13].
Passos do teste	Loga-se em uma conta diferente da usada para a criação do <i>post</i> e segue-se o exato mesmo passo a passo do [T08].
Resultado esperado	São criados uma nova interação de sugestão e um <i>commit</i> em uma nova <i>branch</i> no banco de dados.
Resultado obtido	Sucesso.

Tabela 9: Caso de teste [T09].

Título	[T10] Criar um “ramo” de uma versão específica de um <i>post</i>
Descrição	Testa-se a criação de um “ramo” de uma versão específica de um <i>post</i> . Refere-se ao [RF14].
Passos do teste	Loga-se em uma conta diferente da usada para a criação do <i>post</i> e o usuário clica no botão de clonar do <i>post</i> , abrindo um modal, o usuário então preenche o campo “título da cópia” e clica em “clonar”.
Resultado esperado	É criado uma nova <i>branch</i> pertencente ao usuário no banco de dados, por fim o usuário é redirecionado para a página do <i>post</i> criado.
Resultado obtido	Sucesso.

Tabela 10: Caso de teste [T10].

Título	[T11] Acatar sugestão de edição de um <i>post</i>
Descrição	Testa-se o acatamento de uma sugestão de edição em um <i>post</i> . Refere-se aos [RF16] e [RF17].
Passos do teste	O usuário autor do <i>post</i> clica no botão de incorporar sugestões do <i>post</i> , abrindo um modal, então o usuário clica no título de uma das sugestões sendo redirecionado ao <i>post</i> com a edição sugerida, então clica no botão de aceitar sugestão.
Resultado esperado	É realizado o <i>merge</i> da <i>branch</i> do autor do <i>post</i> e do autor da sugestão no banco de dados, então o usuário é redirecionado para a página do <i>post</i> refletindo as alterações.
Resultado obtido	Sucesso.

Tabela 11: Caso de teste [T11].

Título	[T12] Realizar uma crítica em um <i>post</i>
Descrição	Testa-se a realização de uma crítica em um <i>post</i> . Refere-se ao [RF19].
Passos do teste	O usuário seleciona um trecho do texto do <i>post</i> , e então clica no botão “criticar”, abrindo um <i>frame</i> de crítica, o usuário então preenche o título e corpo da crítica e clica no botão de confirmar.
Resultado esperado	O trecho criticado é sublinhado e é criado um registro da crítica no banco de dados.
Resultado obtido	Sucesso.

Tabela 12: Caso de teste [T12].

4.3 Comparativo com outras plataformas

Nesta seção, faremos um comparativo de funcionalidades entre a nossa solução e outras propostas semelhantes, já estabelecidas

4.3.1 *Pol.is*

A premissa principal do *Pol.is* é coletar um grande número de amostras de opiniões de seus usuários e aplicar uma redução de dimensionalidade, através de uma Análise de Componentes Principais (do inglês, *Principal Components Analysis* - PCA), e então aplicar um algoritmo de clusterização, em seu caso um *K-means*, para agrupar os participantes em grupos de opiniões semelhantes, e, então, encontrar pontos comuns entre os diferentes grupos e os pontos que melhor representam cada grupo, visando estimular o diálogo e compreensão mútua entre tais grupos e eventualmente fomentar o consenso.

Nossa plataforma se diferencia ao possibilitar que os participantes criem conjuntamente respostas cada vez mais complexas e completas, ao fazer uso das propriedades evolutivas do SCV. Através de um engajamento contínuo e competição entre as diversas respostas espera-se que os usuários cheguem em um conjunto de respostas que representem melhor cada grupo. Enquanto o *Pol.is* objetiva ao fim obter várias frases curtas que representem a opinião de seus usuários, nós pretendemos que os usuários criem as respostas mais completas, que representem suas opiniões. Outro ponto de distinção é que nossa plataforma tem um funcionamento satisfatório até em contextos com poucos usuários, bem como com perguntas binárias ou com um viés prévio, cenários os quais o *Pol.is* pode não perfomar bem [1].

4.3.2 *Decidim*

O *Decidim*¹⁴ é uma plataforma modular, que conta com múltiplos espaços para a participação de seus usuários. Cada um desses espaços direciona o esforço dos usuários para objetivos específicos e podem utilizar de diferentes “componentes” que a ferramenta fornece — como enquetes, votações, reuniões, debates, dentre outros —, o que permite que organizações consigam modelar um sistema democrático adaptado a suas necessidades. Inclusive, a plataforma *Brasil Participativo*, citada no capítulo de introdução, trata-se de uma instância adaptada do *Decidim*.

A solução que propomos se distingue ao possibilitar a realização de boa parte das funcionalidades propostas pelo *Decidim* sob uma única estrutura — similar à de um fórum de discussão —, o que pode facilitar o aprendizado do uso da plataforma por novos usuários, diminuindo, assim, a barreira de entrada para a adoção por uma comunidade.

4.3.3 *Consul*

O *Consul*¹⁵ é uma plataforma modular semelhante ao *Decidim*, mas ao invés de fornecer diferentes espaços, como ocorre no *Decidim*, a plataforma oferta os chamados “processos participativos”, os quais são módulos que oferecem as funcionalidades de debates, propostas, votação, legislação colaborativa e orçamento participativo.

Assim como no caso do *Decidim*, nossa plataforma se difere ao ter uma maior simplicidade de design, sem ficar para trás em possibilidades de aplicação.

4.3.4 *Wikilegis*

O *Wikilegis*¹⁶ é uma ferramenta digital que permite a realização de trabalho colaborativo na construção da lei, em que os cidadãos podem atuar diretamente sobre projetos de lei e propostas legislativas com indicações de apoio, comentários e sugestões de nova redação a artigos ou parágrafos.

Nossa proposta se difere ao não se limitar ao escopo legislativo, podendo ser usada para uma multitude de aplicações. Além disso, também oferece uma interface mais enxuta para as operações de edição do texto e possibilita uma maior customização dos textos por parte dos usuários, ao permitir que criem cada um sua “ramificação” própria do conteúdo original.

¹⁴Website oficial disponível em: <https://decidim.org/>. Acesso em 15 de abril de 2024

¹⁵Website oficial disponível em: <https://consuldemocracy.org/>. Acesso em 18 de abril de 2024

¹⁶Há diversas instâncias do Wikilegis no ar, mas sua primeira instância foi utilizada pela Câmara dos Deputados do Brasil e está disponível em: <https://edemocracia.cl.df.leg.br/wikilegis>. Acesso em 15 de abril de 2024

4.4 Análise dos resultados

Os resultados dos testes, que cobriram o fluxo de uso da aplicação através das funcionalidades chave, apontam que a aplicação está em condições adequadas para uso, dado que foram todos bem sucedidos. Todavia, é necessário notar que os testes descritos aqui não consideram uma enorme gama de possibilidades de casos que usuários maliciosos poderiam utilizar para ganhar vantagens na plataforma, como criação de contas falsas para manipulação das votações e opiniões e outras diversas finalidades, como já ocorre em outras redes sociais [22].

Por fim, quando comparado à outras soluções, o **colcom** demonstrou ser uma plataforma com diferenciais significativos e que pode oferecer uma contribuição agregadora para o processo participativo de diversas comunidades.

5 CONCLUSÕES E TRABALHOS FUTUROS

A rede social **colcom** se mostrou uma solução viável para o fomento da participação direta de indivíduos de uma comunidade em seus processos de decisão, fornecendo tecnologias já bem estabelecidas em favor da deliberação. Ademais, trata-se de uma contribuição nova para o cenário de aplicações que promovem a democracia digital ao fazer uso de conceitos de SCV para gerenciar as propostas dos usuários.

Entretanto, deve ser salientado que a aplicação, em seu presente estado, ainda é inviável para ser usada em comunidades de grande escala, pois, como visto no capítulo 4, ainda não possui os devidos tratamentos adequados para a verificação dos usuários e das postagens realizadas, sendo assim altamente susceptível à possíveis tentativas de abuso, como: criação de contas falsas, fraude nas votações com uso de tais contas, *spamming*, *trolling*, dentre outras.

Observa-se também a possibilidade da inserção de mais elementos de gameificação, o que potencialmente pode aumentar o engajamento dos usuários. Contudo, devemos observar que tais dinâmicas acabam por inserir pontos de fricção entre os participantes, cabendo um tratamento cuidadoso para se evitar atritos que venham a degradar a experiência participativa [18], como, por exemplo, usuários mais experientes na plataforma usando as dinâmicas propostas para minar a participação de outros usuários.

Como sugestões de trabalhos futuros, pretende-se:

- Inserir um balanceador de carga e *proxy* reverso, como o *Nginx*;
- Containerizar a aplicação com o *Docker*, para facilitar o *deploy*;
- Alterar o formato das mensagens trocadas entre *front-end* e *back-end* de *JSON* para formatos mais eficientes, como os *Protocol Buffers*;
- Implementar checagens de dados mais robustas, a fim de reduzir o cadastro de potenciais usuários falsos e mal intencionados;
- Implementar testes unitários e de integração automatizados;
- Inserir *tags* para agrupar conteúdos de mesmo tipo;
- Implementar um buscador por texto;
- Implementar a capacidade de realizar um *merge* entre *posts* diferentes;
- Implementar a capacidade de se ter um post com autoria de múltiplos usuários;
- Implementar um visualizador para ver os tópicos mais promovidos em dias passados;
- Inserir um *leaderboard* dos usuários que mais colaboraram.

REFERÊNCIAS

- [1] SMALL, Christopher et al. Polis: Scaling deliberation by mapping high dimensional opinion spaces. **Recerca: revista de pensament i anàlisi**, v. 26, n. 2, 2021. Disponível em: <http://dx.doi.org/10.6035/recerca.5516>.
- [2] GOMES, Wilson. A democracia digital e o problema da participação civil na decisão política. **Revista Fronteiras**, v. 7, n. 3, p. 214-222, 2005.
- [3] AICHNER, Thomas et al. Twenty-five years of social media: a review of social media applications and definitions from 1994 to 2019. **Cyberpsychology, behavior, and social networking**, v. 24, n. 4, p. 215-222, 2021. Disponível em: <http://dx.doi.org/10.1089/cyber.2020.0134>.
- [4] KHONDKER, Habibul Haque. Role of the new media in the Arab Spring. **Globalizations**, v. 8, n. 5, p. 675-679, 2011. Disponível em: <http://dx.doi.org/10.1080/14747731.2011.621287>.
- [5] MACHADO, Jorge; MISKOLCI, Richard. Das jornadas de junho à cruzada moral: o papel das redes sociais na polarização política brasileira. **Sociologia & Antropologia**, v. 9, p. 945-970, 2019. Disponível em: <http://dx.doi.org/10.1590/2238-38752019v9310>.
- [6] BOHDANOVA, Tetyana. Unexpected revolution: the role of social media in Ukraine's Euromaidan uprising. **European View**, v. 13, n. 1, p. 133-142, 2014. Disponível em: <https://doi.org/10.1007/s12290-014-0296-4>
- [7] MUNDT, Marcia; ROSS, Karen; BURNETT, Charla M. Scaling social movements through social media: The case of Black Lives Matter. **Social media+ society**, v. 4, n. 4, p. 2056305118807911, 2018. Disponível em: <http://dx.doi.org/10.1177/2056305118807911>.
- [8] KUBIN, Emily; VON SIKORSKI, Christian. The role of (social) media in political polarization: a systematic review. **Annals of the International Communication Association**, v. 45, n. 3, p. 188-206, 2021. Disponível em: <http://dx.doi.org/10.1080/23808985.2021.1976070>.
- [9] CINELLI, Matteo et al. The echo chamber effect on social media. **Proceedings of the National Academy of Sciences**, v. 118, n. 9, p. e2023301118, 2021. Disponível em: <http://dx.doi.org/10.1073/pnas.2023301118>.
- [10] SAMPAIO, Rafael Cardoso et al. **Democracia digital no Brasil: mapeamento e análises de artigos publicados em periódicos entre 1999-2018**. 2021. Disponível em: <http://dx.doi.org/10.38116/bapi25art2>.

- [11] CONGGE, Umar et al. Digital democracy: A systematic literature review. **Frontiers in Political Science**, v. 5, p. 972802, 2023. Disponível em: <https://doi.org/10.3389/fpos.2023.972802>.
- [12] CHACON, Scott; STRAUB, Ben. **Pro git**. Springer Nature, 2014. Disponível em: <https://doi.org/10.1007/978-1-4842-0076-6>.
- [13] ABDULLAH, Hanin M.; ZEKI, Ahmed M. Frontend and backend web technologies in social networking sites: Facebook as an example. In: **2014 3rd international conference on advanced computer science applications and technologies**. IEEE, 2014. p. 85-89. Disponível em: <https://doi.org/10.1109/ACSAT.2014.22>.
- [14] SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, Shashank. Database system concepts. 2011.
- [15] ZOLKIFLI, Nazatul Nurlisa; NGAH, Amir; DERAMAN, Aziz. Version control system: A review. **Procedia Computer Science**, v. 135, p. 408-415, 2018. Disponível em: <https://doi.org/10.1016/j.procs.2018.08.191>.
- [16] GLINZ, Martin. On non-functional requirements. In: **15th IEEE international requirements engineering conference (RE 2007)**. IEEE, 2007. p. 21-26. Disponível em: <https://doi.org/10.1109/RE.2007.45>.
- [17] CHUNG, Lawrence; DO PRADO LEITE, Julio Cesar Sampaio. On non-functional requirements in software engineering. **Conceptual modeling: Foundations and applications: Essays in honor of john mylopoulos**, p. 363-379, 2009. Disponível em: https://doi.org/10.1007/978-3-642-02463-4_19
- [18] TSENG, Yu-Shan. Rethinking gamified democracy as frictional: a comparative examination of the Decide Madrid and vTaiwan platforms. **Social & Cultural Geography**, v. 24, n. 8, p. 1324-1341, 2023. Disponível em: <http://dx.doi.org/10.1080/14649365.2022.2055779>.
- [19] RAM, Karthik. Git can facilitate greater reproducibility and increased transparency in science. **Source code for biology and medicine**, v. 8, p. 1-8, 2013. Disponível em <https://doi.org/10.1186/1751-0473-8-7>.
- [20] WONG, Nikita A.; BAHMANI, Hamed. A review of the current state of research on artificial blue light safety as it applies to digital devices. **Heliyon**, v. 8, n. 8, 2022. Disponível em: <https://doi.org/10.1016/j.heliyon.2022.e10282>.
- [21] PHIPPS-NELSON, Jo et al. Blue light exposure reduces objective measures of sleepiness during prolonged nighttime performance testing. **Chrono-**

- biology International**, v. 26, n. 5, p. 891-912, 2009. Disponível em: <https://doi.org/10.1080/07420520903044364>
- [22] ORABI, Mariam et al. Detection of bots in social media: a systematic review. **Information Processing & Management**, v. 57, n. 4, p. 102250, 2020. Disponível em: <https://doi.org/10.1016/j.ipm.2020.102250>.
- [23] Sobre o Brasil Participativo. gov.br, 2023. Disponível em: <https://brasilparticipativo.presidencia.gov.br/processes/brasilparticipativo/f/33/>. Acesso em: 25 de março de 2024.
- [24] Usuários de Internet - Indicador Ampliado. cetic, 2023. Disponível em: <https://cetic.br/pt/tics/domicilios/2023/individuos/C2A/>. Acesso em: 25 de março de 2024.
- [25] Global social media statistics. DataReportal, 2024. Disponível em: <https://datareportal.com/social-media-users/>. Acesso em: 24 de março de 2024.
- [26] Digital 2024: Brazil. DataReportal, 2024. Disponível em: <https://datareportal.com/reports/digital-2024-brazil/>. Acesso em: 3 de abril de 2024.