

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

FRANCISCA JOSEANE PAULINO BRILHANTE

TellMe: UMA PONTE DE COMUNICAÇÃO SIMPLIFICADA ENTRE O ALUNO E A  
INSTITUIÇÃO DE ENSINO

CAJAZEIRAS  
2025

Francisca Joseane Paulino Brilhante

TellMe: UMA PONTE DE COMUNICAÇÃO SIMPLIFICADA ENTRE O ALUNO E A  
INSTITUIÇÃO DE ENSINO

Trabalho de Conclusão de Curso  
submetido ao Instituto Federal de  
Educação, Ciência e Tecnologia da  
Paraíba, Campus Cajazeiras, como  
requisito parcial para a obtenção do  
grau de Tecnólogo em Análise e  
Desenvolvimento de Sistemas.

Orientador: Prof. Dr. Fabio Gomes  
de Andrade

Cajazeiras  
2025

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

B857t Brilhante, Francisca Joseane Paulino.

TellMe : uma ponte de comunicação simplificada entre o aluno e a instituição de ensino / Francisca Joseane Paulino Brilhante. – Cajazeiras, 2025.

47f. : il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2025.

Orientador: Prof. Dr. Fabio Gomes de Andrade.

1. Desenvolvimento de sistemas. 2. Sistema TellMe. 3. Sistema de comunicação. 4. Instituição de ensino. I. Instituto Federal 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

FRANCISCA JOSEANE PAULINO BRILHANTE

**TelIIME: UMA PONTE DE COMUNICAÇÃO SIMPLIFICADA ENTRE O ALUNO E A  
INSTITUIÇÃO DE ENSINO**

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. Dr. Fabio Gomes de Andrade

Aprovada em: **21 de Março de 2025.**

Prof. Dr. Fabio Gomes de Andrade - Orientador

Prof. Esp. João Igor Barros Rocha - Avaliador

IFPB - Campus Cajazeiras

Prof. Tec.go. Antônio Ricart Jacinto de Oliveira Medeiros - Avaliador

IFPB - Cajazeiras

Documento assinado eletronicamente por:

- **Fabio Gomes de Andrade**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 24/03/2025 16:18:04.
- **Joao Igor Barros Rocha**, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO, em 24/03/2025 17:39:27.
- **Antonio Ricart Jacinto de Oliveira Medeiros**, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO, em 25/03/2025 08:37:41.

Este documento foi emitido pelo SUAP em 24/03/2025. 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 687198  
Verificador: 5bf2c724f5  
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

A jornada até aqui não foi fácil, e olhar para trás me faz perceber o quanto cada pessoa ao meu lado foi essencial para essa conquista.

Primeiramente, quero agradecer a Deus por sempre estar presente em minha vida, me mostrando o melhor caminho a seguir. Em seguida, aos meus pais, Geraldo e Teresinha, em especial à minha mãe pelo amor incondicional, pelos conselhos e pelo apoio em cada momento de dificuldade. Sua força e dedicação sempre foram minha inspiração.

À minha irmã, Jeane Brilhante, que esteve ao meu lado nos momentos bons e ruins, me incentivando e me lembrando da minha capacidade quando eu mesma duvidava. Sua presença foi essencial para que eu chegasse até aqui.

Ao meu esposo, Klinsman Jorge, por todo carinho, paciência e suporte durante essa caminhada, pois trilhamos juntos o caminho até nossa formação. Obrigada por acreditar em mim, por me motivar nos dias difíceis e por ser meu porto seguro quando o cansaço e a insegurança tentavam me desanimar.

Ao meu orientador, Prof. Dr. Fábio Gomes de Andrade, por toda dedicação, pelos ensinamentos e pelas orientações fundamentais para que eu concluísse esse ciclo. Sua paciência e incentivo fizeram toda a diferença na minha trajetória acadêmica.

Às minhas amigas, Lindinêz e Leila, que foram essenciais no início dos meus estudos, compartilhando não só um apartamento, mas também risadas, desafios e aprendizados. Sou grata por cada momento vivido juntas, que fortaleceu ainda mais a nossa amizade.

E, em especial, à minha filha, Maria Louise. Desde o momento em que chegou à minha vida, trouxe consigo um significado maior para tudo o que faço. Mesmo diante das dificuldades após o seu nascimento, foi ao olhar para você que encontrei forças para seguir em frente. Você foi e sempre será o meu maior incentivo, a razão pela qual busco ser melhor a cada dia. Tudo isso é para nós, para que possamos construir um futuro lindo juntas.

A todos vocês, minha eterna gratidão. Sem o amor, apoio e incentivo de cada um, essa conquista não teria sido possível. Muito obrigada!

## RESUMO

O sistema de Ensino Superior no Brasil tem atendido cada vez mais jovens de diferentes classes sociais, os estudantes buscam um lugar de aprendizado para se inserir no ambiente de trabalho. Com isso as universidades tentam oferecer qualidade de ensino além de um ambiente agradável para os universitários, mas devido a alta demanda, já que diferentes cursos são ofertados por cada instituição de ensino, as universidades abrangem uma larga ocupação dos estudantes e na tentativa de administrar um ambiente composto por muitos discentes e docentes, algumas dificuldades podem surgir e muitas das vezes foge do conhecimento da administração da instituição. A falta de um meio que facilite a interação do aluno com os setores responsáveis do instituto, dificulta ainda mais a administração ter ciência desses problemas, embora seja oferecidos serviços de Ouvidorias para que o aluno deposite seu relato, esse meio não oferece facilidade e muito menos uma certeza de êxito em sua resposta para solucionar tal problema. Com base nisso, este trabalho propõe o desenvolvimento de uma ferramenta *Web* que busque resolver esse problema, facilitando a comunicação entre os discentes e a instituição de ensino de forma ágil e prática.

**Palavras-chave:** Web. Instituição de Ensino. Comunicação.

## **ABSTRACT**

The Higher Education system in Brazil has increasingly served young people from different social classes, students are looking for a place of learning to enter the work environment. With this, universities try to offer quality teaching in addition to a pleasant environment for university students, but due to high demand, since different courses are offered by each educational institution, universities cover a large occupation of students and in an attempt to manage a environment composed of many students and professors, some difficulties may arise and many times it is beyond the knowledge of the administration of the institution. The lack of a means to facilitate the student's interaction with the responsible sectors of the institute, makes it even more difficult for the administration to be aware of these problems, although Ombudsman services are offered for the student to deposit his report, this means does not offer ease, let alone a certainty of success in your answer to solve this problem. Based on this, this work proposes the development of a Web tool that seeks to solve this problem, facilitating communication between students and the educational institution in an agile and practical way.

**Keywords:** Web. Educational institution. Communication.

## LISTA DE FIGURAS

Figura 1 - Gráfico referente a evolução dos indicadores de trajetória dos estudantes ingressantes em Cursos de Graduação em 2014 no Brasil (2014-2023) .....	12
Figura 2 - Arquitetura do Django .....	22
Figura 3 - Diagrama de Casos de uso .....	28
Figura 4 - Modelagem do esquema lógico .....	30
Figura 5 - Arquitetura do sistema .....	32
Figura 6 - Tela de fazer registro .....	39
Figura 7 - Tela de login .....	40
Figura 8 - Tela de recuperação de senha .....	40
Figura 9 - Tela de perfil de usuário .....	41
Figura 10 - Tela de alteração de dados .....	41
Figura 11 - Tela de alteração de senha .....	42
Figura 12 - Tela inicial da aplicação .....	42
Figura 13 - Tela de fazer postagem .....	43
Figura 14 - Tela de deixar reações: curtir, descurtir e comentar .....	43
Figura 15 - Tela de visualização de postagens do usuário .....	44

## LISTA DE QUADROS

Quadro 1 - Cronograma de atividades do TCC 1 .....	17
Quadro 2 - Cronograma de atividades do TCC 2 .....	17
Quadro 3 - Comparação entre as ferramentas .....	19
Quadro 4 - Casos de teste .....	29

## LISTA DE ALGORITMOS

Algoritmo 1 - Implementação de um Filtro de Publicações .....	25
Algoritmo 2 - Trecho do código da view de cadastro do usuário .....	35
Algoritmo 3 - Trecho do código da view de login .....	36
Algoritmo 4 - View de Criar Postagem .....	37
Algoritmo 5 - Função de excluir postagem .....	38
Algoritmo 6 - View editar postagem .....	38

## LISTA DE ABREVIATURAS E SIGLAS

CPA	Comissão Própria de Avaliação
CSS	Cascading Style Sheet
DRY	Don't Repeat Yourself
EJA	Educação de Jovens e Adultos
HTML	HiperText Markup Language
HTTPS	HyperText Transfer Protocol Secure
IES	Instituição de Ensino Superior
IFPB	Instituto Federal da Paraíba
INEP	Instituto Nacional de Estudos e Pesquisas Educacionais
LLMs	Large Language Models
MTV	Model Template View
MVC	Model View Controller
NUCA	Núcleo de Combate ao Assédio
ORM	Object Relational Mapper
PRAE	Instituído pela Pró-Reitoria de Assuntos Estudantis
RCA	Rede de Combate ao Assédio
SGBD	Sistema de Gestão de Bases de Dados
TCC	Trabalho de Conclusão de Curso
URL	Uniform Resource Locato

## SUMÁRIO

<b>1. INTRODUÇÃO</b>	<b>11</b>
1.1 Motivação	13
1.2 Objetivos	14
1.2.1 Objetivo Geral	14
1.2.2 Objetivos Específicos	14
1.3 Metodologia	15
1.4 Cronograma	16
1.5 Organização do Documento	17
<b>2. TRABALHOS RELACIONADOS</b>	<b>18</b>
<b>3. FUNDAMENTAÇÃO TEÓRICA</b>	<b>20</b>
3.1 Python	20
3.2 Django	21
3.3 Ollama	24
<b>4. A SOLUÇÃO PROPOSTA</b>	<b>26</b>
4.1 Usuários da aplicação	26
4.2 Requisitos Funcionais	26
4.3 Diagrama de Casos de uso	28
4.4 Casos de testes	28
4.5 O banco de dados	29
4.5.1 Modelo de Banco de Dados	29
4.5.2 Modelagem de Dados	30
4.6 Arquitetura da Ferramenta	32
4.7 Implementação da Ferramenta	34
4.7.1 Módulo de Autenticação e controle de Usuário	35
4.7.2 Módulo de Publicação	36
4.7.3 Módulo de Visualização	39
<b>5. CONCLUSÃO</b>	<b>45</b>
<b>REFERÊNCIAS</b>	<b>46</b>

## 1. INTRODUÇÃO

As Instituições de Ensino Superior (IES) desempenham um papel crucial na formação profissional, desenvolvendo competências essenciais para o mercado de trabalho (ALMEIDA; SILVA; FRANÇA; REIS, 2023). Para que isso ocorra, durante toda a trajetória acadêmica, são executados trabalhos de pesquisa e atividades avaliativas com o intuito de medir os conhecimentos dos estudantes adquiridos durante cada período letivo. Ao final do curso, os estudantes obtêm o diploma de conclusão do curso e assim estão aptos a exercerem a profissão escolhida. Como as IES exercem um papel tão importante na formação de profissionais, sejam eles da saúde, da construção civil, da educação, da tecnologia, entre outros, é essencial avaliar como se encontra o andamento da capacitação dos seus estudantes, principalmente identificando as dificuldades enfrentadas pelos alunos durante a busca por sua formação.

A falta de ambientes propícios para o ensino, a escassez de transportes disponibilizados aos alunos para a sua locomoção até o centro universitário, a indisponibilidade de refeições oferecidas pela instituição e a desqualificação de profissionais docentes são alguns dos possíveis problemas enfrentados pelos alunos. De acordo com o estudo de Dias et al. (2019), muitos estudantes percebem dificuldades significativas ao realizar a transição para o ensino superior, enfrentando desafios como a adaptação ao ambiente acadêmico e questões emocionais que afetam o desempenho. Já a pesquisa de Ganam e Pinezi (2021) revelou que os estudantes atendidos por programas de assistência estudantil enfrentam desafios materiais e simbólicos, como a falta de recursos financeiros e a dificuldade em se sentir parte da universidade.

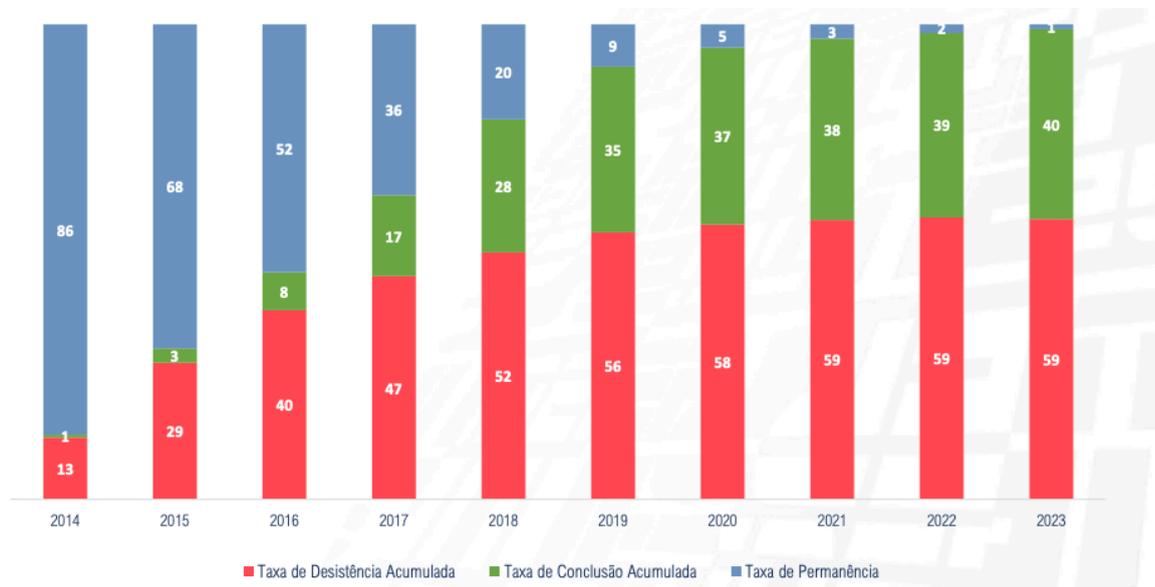
O gráfico da Figura 1 mostra os indicadores de trajetória dos estudantes ingressantes em Cursos de Graduação em 2014 no Brasil, entre os anos de 2014 e 2023, de acordo com dados coletados pelo INEP<sup>1</sup> (Instituto Nacional de Estudos e Pesquisas Educacionais). Nesse período, a taxa de desistência acumulada aumentou de 13% para 59%, com o crescimento mais acentuado até 2018. A taxa de conclusão acumulada subiu de 1% para 40%, com maior avanço entre 2016 e 2019. Já a taxa de permanência caiu de 86% para 1%, indicando que, ao longo do

---

<sup>1</sup> portal.inep.gov.br

tempo, a maioria dos estudantes concluiu o curso ou desistiu. Os dados destacam a importância de estratégias para redução da evasão e incentivo à conclusão.

*Figura 1 - Gráfico referente a evolução dos indicadores de trajetória dos estudantes ingressantes em Cursos de Graduação em 2014 no Brasil (2014-2023)*



*Fonte: INEP<sup>2</sup> (2023)*

Diante do contexto apresentado, a comunicação entre os universitários e as IES torna-se fundamental para descobrir quais são as circunstâncias que ocasionam a desistência dos alunos. Essa interação deve ocorrer de maneira constante e ágil, com a finalidade de prevenir a evasão do público estudantil e buscar uma melhoria na qualidade de ensino. Muitas universidades adotam em seus portais institucionais mecanismos de comunicação como, por exemplo, *Fale Conosco*, *Ouvidoria* e *Contatos Telefônicos*. Contudo, é necessário observar como está o atendimento por esses meios e se eles realmente atendem aos chamados dos alunos, sejam eles relacionados à educação ou à administração.

Assim, disponibilizar um suporte estudantil que tenha agilidade e eficiência para resolver os impedimentos encontrados durante a graduação não é fácil e por isso as ferramentas de comunicação que já existem apresentam pequenas falhas em seus atendimentos, uma vez que a resposta esperada pelos alunos, ao se

<sup>2</sup>

[https://download.inep.gov.br/educacao\\_superior/censo\\_superior/documentos/2023/apresentacao\\_censo\\_da\\_educacao\\_superior\\_2023.pdf](https://download.inep.gov.br/educacao_superior/censo_superior/documentos/2023/apresentacao_censo_da_educacao_superior_2023.pdf)

utilizar desses meios, nem sempre são obtidas ou existe uma demora em seu atendimento. Com isso, ao propor uma ferramenta tecnológica eficaz que auxilie nesse processo de descoberta dos problemas sofridos pelos estudantes, será possível obter como resultado um ambiente mais saudável e acolhedor para os estudantes, o que vai melhorar a qualidade do ensino e proporcionar conforto aos estudantes, de forma que venha ajudar a administração na elaboração de ações e propostas para a redução do grau de evasão estudantil.

### **1.1 Motivação**

Com base nos recursos de comunicação existentes nas IES, destaca-se a Ouvidoria, que exerce um canal de interação muito importante para que alunos, professores e funcionários possam relatar insatisfações, reivindicações, reclamações, dúvidas, apresentar elogios ou até mesmo sugestões para a área administrativa das Instituições, com o objetivo de corrigir e oferecer qualidade no serviço educacional.

Contudo, nem todas as IES utilizam dessa ferramenta ou, em alguns casos, as Ouvidorias não são bem implementadas, apresentando falha em algumas situações como, por exemplo, em relação à demora no tempo de resposta por parte do setor responsável, a falta de acolhimento da reclamação com o foco de melhorar a gestão organizacional das Instituições de Ensino ou até mesmo sendo um meio desconhecido por muitos estudantes.

O Instituto Federal da Paraíba<sup>3</sup> (IFPB) adota em seu regimento interno a *Comissão Própria de Avaliação*<sup>4</sup> (CPA), que é uma comissão responsável por conduzir os processos de avaliação interna da instituição e prestar as informações requeridas pelo INEP. A CPA exerce como principal função a elaboração e a apresentação de uma proposta de autoavaliação institucional, juntamente com os conselhos superiores da instituição de ensino e com a comunidade acadêmica. Atualmente, a cada ano, a CPA elabora um relatório de avaliação. Esse relatório tem como base a aplicação de um questionário disponibilizado aos discentes, docentes e técnicos administrativos, com o foco de monitoramento dos serviços prestados pelo instituto. Apesar de muito importante, a avaliação interna demora

---

<sup>3</sup> <https://ifpb.edu.br/cajazeiras>

<sup>4</sup> <https://www.ifpb.edu.br/cpa>

um certo tempo para ser realizada, o que dificulta a descoberta de problemas ou até mesmo sugestões para o aprimoramento da instituição de ensino.

No último Relatório<sup>5</sup> da CPA feito em 2023, foram propostas ações de melhorias para serem desenvolvidas no IFPB, tendo como base os dados coletados através do questionário aplicado aos discentes, docentes e técnicos administrativos. Alguns exemplos de ações propostas foram expandir os meios de comunicação com os alunos por meio de redes sociais e aplicativos de mensagens instantâneas, aprimorar a comunicação com os estudantes sobre a oferta de estágios por curso e expandir parcerias com empresas, revisar as condições de segurança dos campi, entre outras melhorias propostas.

Dessa forma, com a finalidade de buscar soluções para reduzir as dificuldades encontradas pelo público estudantil e apresentar ideias que venham a melhorar o ensino, este trabalho de conclusão de curso propõe o desenvolvimento o *TellMe*, uma aplicação *Web* com a finalidade de oferecer ao aluno uma forma simples e rápida de comunicação com a administração da Instituição de Ensino. A aplicação proposta tem como intuito proporcionar uma melhor interface de acesso entre os usuários e os setores responsáveis das IES, além de permitir aos alunos, servidores e administradores a possibilidade de fazer reclamações ou feedbacks de forma anônima.

## 1.2 Objetivos

Esta seção descreve os objetivos deste trabalho.

### 1.2.1 Objetivo Geral

O objetivo geral deste TCC consiste no desenvolvimento de uma ferramenta denominada *TellMe*, uma aplicação *web* que tem como intuito proporcionar agilidade e efetividade aos estudantes do campo IFPB no ato de interação com os setores responsáveis da administração institucional, sendo utilizada para relatar problemas ou sugestões para a melhoria do ensino.

### 1.2.2 Objetivos Específicos

O trabalho tem também os seguintes objetivos específicos:

---

<sup>5</sup> <https://www.ifpb.edu.br/cpa/relatorios-de-avaliacao-interna-1>

- disponibilizar uma aplicação para coletar as dificuldades enfrentadas pelos estudantes e colaboradores institucionais;
- Proporcionar aos usuários a opção de realizar publicações anônimas ou públicas, garantindo liberdade de expressão com segurança;
- Dispor de funcionalidades para visualização, edição e exclusão de publicações, permitindo um melhor gerenciamento das informações compartilhadas;
- Criar uma interface intuitiva e responsiva, proporcionando uma experiência de uso fluida e eficiente para todos os usuários.

### 1.3 Metodologia

Com o foco de obter resultados positivos no desenvolvimento deste trabalho de conclusão de curso foi necessário organizar as atividades efetuadas, obtendo assim uma visão clara e objetiva das técnicas necessárias para atingir o êxito. A metodologia utilizada para a elaboração deste projeto foi definida de acordo com as seguintes atividades:

- **Definição do tema (A<sub>1</sub>):** nessa atividade foi definido o tema o qual seria abordado neste trabalho de conclusão, sendo assim a etapa inicial do projeto;
- **Fundamentação teórica (A<sub>2</sub>):** essa atividade está relacionada à pesquisa literária sobre o tema abordado;
- **Levantamento de requisitos (A<sub>3</sub>):** nessa atividade foram analisados e definidos os requisitos da aplicação necessários para que o objetivo geral e os objetivos específicos fossem atingidos, além da elaboração das funcionalidades que a ferramenta deve oferecer aos usuários;
- **Projetar a base de dados (A<sub>4</sub>):** nessa atividade foi determinada a estrutura e o esquema dos dados utilizados para a implementação do banco de dados da aplicação;
- **Coleta e análise dos dados (A<sub>5</sub>):** essa atividade refere à coleta e à análise dos dados utilizados para a implementação do trabalho;
- **Definir a arquitetura do sistema (A<sub>6</sub>):** essa atividade está relacionada à elaboração da arquitetura do projeto, assim como os seus módulos e a lógica de comunicação entre os mesmos;

- **Desenvolvimento do protótipo da aplicação (A<sub>7</sub>):** nessa atividade foi desenvolvida a prototipagem do sistema, levando-se em consideração todas as funcionalidades definidas nos requisitos;
- **Entrega e apresentação do TCC1(A<sub>8</sub>):** nessa atividade foi feita a entrega do projeto juntamente com a apresentação do tema a banca avaliadora;
- **Desenvolvimento dos templates (A<sub>9</sub>):** nessa atividade foi feita a criação dos arquivos HTML, CSS e JS da ferramenta;
- **Desenvolvimento e implementação das regras de negócio (A<sub>10</sub>):** essa atividade refere-se à etapa de implementação das views, na qual foram definidas as ações e comportamentos que seriam executados no projeto, garantindo que as regras de negócio fossem corretamente aplicadas e integradas ao fluxo da aplicação;
- **Implementação da ligação com o banco de dados (A<sub>11</sub>):** nessa atividade foi realizada a interação entre a lógica de negócio e o banco de dados;
- **Pesquisa e implementação das validações dos usuários (A<sub>12</sub>):** nessa atividade foi desenvolvida a pesquisa e implementação das validações dos usuários alunos, administradores e servidores;
- **Pesquisa sobre filtros de profanação (A<sub>13</sub>):** essa atividade está relacionada à busca de filtros para barrar publicações que possam denegrir a imagem de alguém e evitar publicações desrespeitosas;
- **Testes de funcionalidades do sistema (A<sub>14</sub>):** nessa atividade foram feitos os testes para verificar o funcionamento da aplicação;
- **Revisão do TCC para entrega final (A<sub>15</sub>):** essa atividade é feita a análise do documento para a entrega final.

#### 1.4 Cronograma

Esta seção descreve a distribuição das atividades relatadas na seção 1.4. O Quadro 1 mostra as atividades desenvolvidas durante a disciplina TCC 1, enquanto que o Quadro 2 mostra as atividades que foram desenvolvidas durante a disciplina TCC 2.

Quadro 1 - Cronograma de atividades do TCC 1

Atividades	ABR	MAIO	JUN	JUL	AGO	SET
A1	X					
A2	X					
A3		X				
A4			X			
A5				X		
A6				X		
A7					X	
A8						X

Fonte: Autor (2022)

Quadro 2 - Cronograma de atividades do TCC 2

Atividades	OUT	NOV	DEZ	JAN	FEV	MAR
A9	X					
A10	X					
A11		X				
A12			X			
A13				X		
A14					X	
A15						X

Fonte: Autor (2024)

### 1.5 Organização do Documento

O restante deste documento está organizado em quatro capítulos. O Capítulo 2 descreve os trabalhos relacionados, destacando aplicações ou canais semelhantes ao *TellMe*. O Capítulo 3 apresenta a fundamentação teórica, mostrando as tecnologias que serão utilizadas para solucionar a problemática abordada neste trabalho. O Capítulo 4 apresenta a solução proposta, descrevendo os principais artefatos que foram produzidos durante o desenvolvimento. Por fim, o Capítulo 5 conclui o documento, apresentando as conclusões e as considerações finais.

## 2. TRABALHOS RELACIONADOS

Atualmente, existem aplicações disponíveis que se assemelham à ferramenta proposta neste trabalho de conclusão de curso. Um desses trabalhos é o *Agenda+*<sup>6</sup>, que é um aplicativo móvel que oferece uma comunicação entre os responsáveis dos educandos e as escolas. As funcionalidades básicas dessa aplicação consistem em receber comunicados escolares, notícias e agendar eventos escolares.

Outro sistema relacionado é o *Connect Escolas*<sup>7</sup>, que é um aplicativo semelhante ao *Agenda+*, desenvolvido para que os responsáveis dos alunos fiquem informados sobre o aprendizado dos discentes e o andamento escolar. A aplicação oferece informações sobre o calendário das atividades e provas, eventos, além de artigos e vídeos com o intuito de orientar os pais ou responsáveis dos alunos.

Diante das funcionalidades das aplicações descritas, a principal diferença relacionada ao sistema proposto por este TCC e as aplicações já existentes está relacionada à priorização das comunicações de forma direta e rápida entre alunos e os setores responsáveis da administração de ensino. Conforme mencionado, o foco desta ferramenta é oferecer aos próprios estudantes a possibilidade de relatarem problemas e avaliarem o ambiente escolar e o ensino oferecido pelas IES, propondo assim mudanças para aprimorar as Instituições de Ensino.

O IFPB disponibiliza de um *Núcleo de Combate ao Assédio* (NUCA), integrado à *Rede de Combate ao Assédio* (RCA) e instituído pela *Pró-Reitoria de Assuntos Estudantis* (PRAE), atua na promoção de um ambiente acadêmico seguro e respeitoso. Sua atuação foca no desenvolvimento de ações preventivas e educativas, além do enfrentamento a qualquer forma de preconceito, discriminação, constrangimento e assédios moral e sexual, garantindo a proteção e o bem-estar da comunidade discente do IFPB.

Além desse órgão existe a CPA, como já mencionada no documento, responsável por apresentar relatórios de avaliações direcionadas ao Instituto e ensino, contribuindo para a melhoria contínua dos serviços educacionais. No entanto, a realização desse processo de forma espaçada pode retardar a identificação de desafios e a implementação de melhorias, evidenciando a

---

<sup>6</sup> <https://apps.apple.com/br/app/agenda/id1557763546>

<sup>7</sup> <http://www.connectescolas.com.br/>

necessidade de estratégias mais ágeis para captar percepções e otimizar a gestão acadêmica e institucional.

O Quadro 3 resume os principais pontos da comparação dos trabalhos relacionados com a aplicação proposta no trabalho, a ferramenta *TellMe* se destaca por proporcionar uma comunicação direta, rápida e aberta entre os alunos e a administração da instituição, enquanto as outras plataformas possuem objetivos mais específicos e limitados.

*Quadro 3 - Comparação entre as ferramentas*

<b>Critério</b>	<b>TellMe</b>	<b>Agenda+</b>	<b>Connect Escolas</b>	<b>NUCA</b>	<b>CPA</b>
<b>Público-alvo</b>	Alunos, professores e colaboradores do IFPB	Pais e responsáveis de alunos	Pais e responsáveis de alunos	Toda a comunidade acadêmica	Toda a comunidade acadêmica
<b>Objetivo principal</b>	Permitir que usuários relatem dificuldades acadêmicas, administrativas e institucionais	Comunicação de comunicados, eventos e notícias escolares	Acompanhar desempenho escolar dos alunos e manter responsáveis informados	Prevenção e combate ao assédio, discriminação e constrangimentos	Avaliação periódica da instituição para melhorias no ensino
<b>Forma de comunicação</b>	Publicações diretas dos usuários na plataforma	Mensagens da escola para os responsáveis	Informações sobre desempenho e calendário escolar	Relatórios e ações preventivas	Pesquisas e relatórios institucionais periódicos
<b>Frequência de atualização</b>	Contínua e em tempo real	Conforme necessidade da escola	Conforme necessidade da escola	Ações planejadas e conforme necessidade	Avaliação periódica e espaçada
<b>Capacidade de resposta</b>	Comunicação rápida e direta com setores responsáveis	Comunicação unilateral (da escola para os responsáveis)	Comunicação unilateral (da escola para os responsáveis)	Encaminhamentos internos conforme necessidade	Implementação de melhorias a médio/longo prazo
<b>Foco na melhoria institucional</b>	Sim, baseado nas dificuldades relatadas pelos usuários	Não	Não	Sim, mas focado em segurança e respeito no ambiente acadêmico	Sim, mas com um processo mais espaçado

*Fonte: Autor (2025)*

### 3. FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta as principais tecnologias utilizadas no desenvolvimento deste trabalho. Primeiramente, será explorada a linguagem *Python*, com uma introdução ao seu conceito e características. Em seguida, será abordado o framework *Django*, detalhando sua arquitetura e principais funcionalidades. E, por fim, é apresentado o conceito da plataforma *Ollama*.

#### 3.1 Python

A linguagem *Python* é uma das mais populares e versáteis no cenário da programação. Criada por Guido van Rossum em 1990, como sucessora da linguagem ABC, *Python* foi desenvolvida com o objetivo de ser simples, legível e produtiva. Sua sintaxe clara e intuitiva permite que desenvolvedores escrevam menos linhas de código para executar a mesma funcionalidade que, em outras linguagens, exige mais complexidade. Essa característica a torna uma excelente escolha tanto para iniciantes quanto para programadores experientes.

Uma das principais vantagens da linguagem *Python* é a sua natureza interpretada. Diferente de linguagens compiladas, que geram arquivos executáveis, *Python* utiliza um interpretador que traduz o código-fonte em tempo de execução. Isso proporciona maior flexibilidade, pois permite testar trechos de código rapidamente, sem a necessidade de uma etapa de compilação. Essa abordagem facilita o desenvolvimento ágil e a depuração de erros.

Além disso, *Python* é uma linguagem de código aberto, o que significa que seu código-fonte está disponível gratuitamente para qualquer usuário. Desenvolvedores podem modificá-lo conforme necessário, sem precisar pagar licenças ou taxas, promovendo uma comunidade ativa e colaborativa. O suporte comunitário é um dos pontos fortes da linguagem, com milhares de desenvolvedores ao redor do mundo contribuindo para sua evolução e manutenção.

Outro diferencial do *Python* é seu vasto ecossistema de bibliotecas e frameworks, que tornam o desenvolvimento de aplicações mais eficiente. Bibliotecas como *NumPy*, *Pandas* e *Matplotlib* são amplamente utilizadas para análise de dados e ciência de dados. O *TensorFlow* e o *PyTorch* são referências em aprendizado de máquina e inteligência artificial. Já o *Django* e o *Flask* são frameworks poderosos para desenvolvimento web. Essas ferramentas tornam *Python* adequado para

diversas áreas, incluindo desenvolvimento web, automação, análise de dados, inteligência artificial, cibersegurança e até mesmo computação científica.

*Python* é utilizado por grandes empresas e plataformas conhecidas, como Instagram, Dropbox, Google, Netflix e Spotify. Sua flexibilidade e facilidade de integração com outras tecnologias fazem com que seja uma escolha ideal para aplicações robustas e escaláveis.

Graças à sua simplicidade, poder e versatilidade, *Python* continua a crescer e se consolidar como uma das linguagens de programação mais influentes da atualidade. Seu uso abrange desde projetos acadêmicos até sistemas corporativos de grande porte, tornando-se uma ferramenta essencial no mundo da tecnologia.

### 3.2 Django

O *Django*<sup>8</sup> é um framework web escrito na linguagem *Python*, sendo guiado pela premissa *DRY* (Don't Repeat Yourself) (Django introduction - Learn web development, 2022), que na língua portuguesa significa Não Se Repita. Tal premissa tem como objetivo permitir um ótimo reaproveitamento de códigos, sendo necessário a realização apenas de pequenas modificações para adequar um código previamente desenvolvido a um novo projeto. Para trabalhar com esse framework é necessário ter conhecimento em *Python*, já que toda a sua base foi desenvolvida nessa linguagem. Além disso, como o *Django* é um framework web, é essencial que se tenha uma familiaridade com o desenvolvimento de aplicações para a Internet, sendo muito importante conhecer tecnologias como HTML, CSS e JavaScript, três blocos fundamentais para o desenvolvimento de sistemas web.

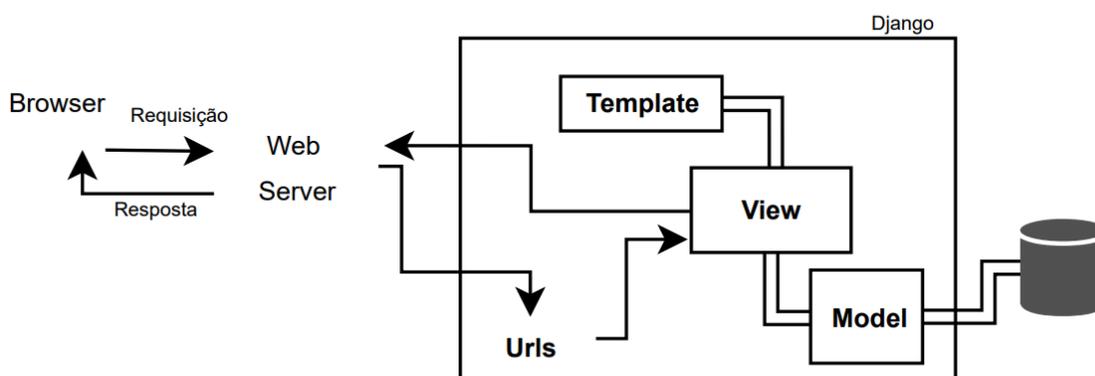
O *Django* foi criado por um grupo de programadores do *Lawrence Journal-World* com a finalidade de tornar o desenvolvimento de aplicações web mais rápido. Esse objetivo é alcançado por meio da disponibilização de algumas tarefas já prontas como autenticação de usuário, site de administração, formulários, entre outros. O que difere esse framework dos demais é o fato de ele usar a estrutura MTV (*Model-Template-View*) (THOUTAM, 2021), enquanto que outros frameworks web utilizam o padrão MVC (*Model-View-Controller/ Modelo-Visão-Controlador*) (LEFF; RAYFIELD, 2001). A estrutura MTV, que é mostrada na Figura 2, tem as seguintes camadas:

---

<sup>8</sup> <https://www.djangoproject.com/>

- **Model:** essa camada é responsável pela interação com o banco de dados. Nela são realizadas as tarefas de inclusão, atualização e exclusão de dados;
- **Template:** é a camada de apresentação, onde ficam os arquivos HTML, CSS e JavaScript que serão renderizados nos navegadores para a visualização do usuário final;
- **View:** é a camada responsável por determinar o que irá acontecer no projeto em desenvolvimento. É nela onde fica a lógica de negócio da aplicação. A *view* recebe uma requisição do cliente e envia uma resposta para a mesma. Assim, a função dessa camada é manipular os dados vindos do banco de dados através da camada *model* e entregar os dados recuperados a camada de apresentação, o *template*.

Figura 2 - Arquitetura do Django



Fonte: Autor (2022)

Observa-se na Figura 2 o fluxo da arquitetura MTV do framework *Django*. O fluxo do processamento de uma requisição inicia-se com o usuário inserindo alguma informação no navegador. Depois, o navegador solicita uma requisição ao servidor web, que logo em seguida verifica a *URL* inserida na camada de *URLs*. Então, entra em cena a camada *View* que é encarregada de formatar os dados recebidos pela camada de *URLs* e entregá-los para a camada *model*, que é responsável por acessar o banco de dados. Em seguida, a informação solicitada é devolvida para a

*view*, que a transfere para o *template* para que a mesma seja apresentada ao usuário.

O framework *Django* possui um *Object Relational Mapper* (ORM) que significa mapeador relacional de objetos. Ele é compatível com os sistemas de gerência de bancos de dados *SQLite*, *MySQL*, *PostgreSQL* e *Oracle*. O ORM pode ser utilizado como um meio de interação entre o banco de dados e a aplicação, tendo como função gerenciar ações no banco de dados.

Outra vantagem do *Django* é que ele tem um módulo de autenticação que oferece uma série de recursos que agilizam o desenvolvimento de aplicações que precisam lidar com a autenticação de usuários. Os principais recursos do módulo de autenticação do *Django* são:

- **Gerenciamento de Usuários:** o *Django* possui um modelo embutido chamado *User*, que já inclui campos essenciais como nome de usuário, e-mail, senha, nome e sobrenome. Além disso, ele permite a criação, atualização e exclusão de usuários de forma fácil;
- **Autenticação e Login:** o framework fornece funções prontas para autenticação, como *authenticate* e *login*, que permitem verificar credenciais e manter sessões de usuários logados;
- **Controle de Permissões:** o *Django* utiliza um sistema de grupos e permissões, permitindo restringir acessos com base em regras predefinidas. Por exemplo, é possível usar `@login_required` e `@permission_required` para proteger views;
- **Sistema de Grupos:** usuários podem ser organizados em grupos, facilitando a atribuição de permissões coletivas, tornando o gerenciamento de acessos mais eficiente;
- **Recuperação de Senha:** o módulo já traz mecanismos para redefinição de senha via e-mail, usando templates personalizáveis para o envio de mensagens aos usuários;
- **Autenticação Personalizada:** caso o modelo padrão de usuário não atenda às necessidades do projeto, é possível estendê-lo ou criar um modelo personalizado utilizando *AbstractUser* ou *AbstractBaseUser*.

### 3.3 Ollama

O *Ollama* é uma plataforma que possibilita a execução local de modelos de inteligência artificial em dispositivos locais, eliminando a necessidade de conexão com a nuvem. Isso oferece maior privacidade, controle sobre os dados e desempenho aprimorado.

Ao integrar grandes modelos de linguagem (Large Language Models - LLM) em aplicações web desenvolvidas com *Django*, o *Ollama* pode ser utilizado para a moderação de conteúdo, identificando e bloqueando automaticamente publicações inadequadas. Isso inclui a detecção de discursos de ódio, spam, ofensas e outros tipos de textos impróprios. O *Ollama* oferece suporte a diversos modelos de IA, incluindo:

- **Llama 2:** modelo de código aberto desenvolvido pela Meta AI, utilizado para geração e compreensão de texto;
- **Code Llama:** versão especializada do Llama 2, projetada para programação e geração de código;
- **Mistral:** modelo compacto e eficiente, com ótimo desempenho em tarefas de processamento de linguagem natural (Natural Language Processing - NLP);
- **Falcon:** outra alternativa de modelo aberto para processamento de texto;
- **WizardCoder:** modelo treinado especificamente para tarefas de codificação e assistência a programadores.

Para a utilização de qualquer um desses modelos no *Ollama*, basta fazer o download e configurá-lo.

O modelo usado na implementação deste trabalho foi o *Mistral*, que é um modelo de linguagem natural de código aberto, criado pela empresa Mistral AI. O modelo *Mistral* pode ser aplicado em diversas áreas, tornando-se uma ferramenta versátil para diferentes necessidades. Ele é amplamente utilizado na moderação de conteúdo, ajudando a identificar e filtrar textos inadequados em plataformas de comunicação. Além disso, é uma excelente opção para o desenvolvimento de *chatbots* e assistentes virtuais, proporcionando interações rápidas e inteligentes com os usuários.

O Algoritmo 1 exibe a função *verificar\_linguagem(texto)*, que utiliza o modelo *Mistral* via *Ollama* para detectar se um texto contém linguagem imprópria. A função

constrói um prompt que pergunta se o texto é ofensivo e solicita que o modelo responda apenas "sim" ou "não". Em seguida, essa pergunta é enviada ao modelo de IA, e a resposta recebida é processada. Se a resposta for "sim", a função retorna *True*, indicando que o texto contém palavras inadequadas; caso contrário, retorna *False*. Além disso, o código inclui um tratamento de erros, garantindo que, caso ocorra uma falha na comunicação com o modelo, a função retorne *False*, evitando que o programa interrompa sua execução. Esse método é útil para moderação automática de conteúdo, podendo ser aprimorado com modelos mais especializados em detecção de linguagem ofensiva.

### *Algoritmo 1 - Implementação de um Filtro de Publicações*

```
def verificar_linguagem(texto):
    try:
        prompt = f"""
        Este texto contém palavras ofensivas ou linguagem
        imprópria?
        Responda apenas com 'sim' ou 'não'.
        Texto: {texto}
        """
        response = ollama.chat(model="mistral", messages=[{"role":
"user", "content": prompt}])
        if response and "message" in response and "content" in
response["message"]:
            resposta =
response["message"]["content"].strip().lower()
            return resposta == "sim"
        else:
            return False
    except Exception as e:
        return False
```

*Fonte: Autor (2025)*

## **4. A SOLUÇÃO PROPOSTA**

Este capítulo apresenta a implementação da ferramenta proposta por este TCC. De início, a seção 4.1 mostra os usuários da aplicação. Em seguida, a seção 4.2 descreve os requisitos funcionais. A seção 4.3 aborda sobre os diagramas de casos de uso. A seção 4.4 mostra os casos de testes. A seção 4.5 apresenta o processo de modelagem do banco de dados. A seção 4.6 aborda o projeto arquitetural da ferramenta, descrevendo os módulos que serão desenvolvidos. E, por fim, a seção 4.7 apresenta a implementação da aplicação.

### **4.1 Usuários da aplicação**

A aplicação *TellMe* é um aplicativo de comunicação voltado para o uso no IFPB e tem como os seguintes usuários:

- Alunos Regulares: Matriculados em cursos técnicos, superiores ou pós-graduação;
- Professores:
  1. Professores Efetivos: Docentes concursados da instituição.
  2. Professores Substitutos: Temporários, mas envolvidos no ambiente acadêmico.
- Colaboradores: Técnicos Administrativos: Responsáveis por setores administrativos e operacionais.
- Gestores:
  1. Coordenadores de Curso: Responsáveis pela gestão acadêmica dos cursos.
  2. Direção de Campus: Diretor-geral e equipe administrativa do campus.
  3. Reitoria: Órgão central da instituição, que pode ter acesso a dados gerais da aplicação.

### **4.2 Requisitos Funcionais**

Os requisitos funcionais foram elaborados pela autora com o intuito de alcançar os objetivos gerais e específicos apresentados na seção (1.2). Dessa forma, os requisitos a seguir buscam garantir o desenvolvimento de uma ferramenta que possa disponibilizar aos usuários a possibilidade de relatar

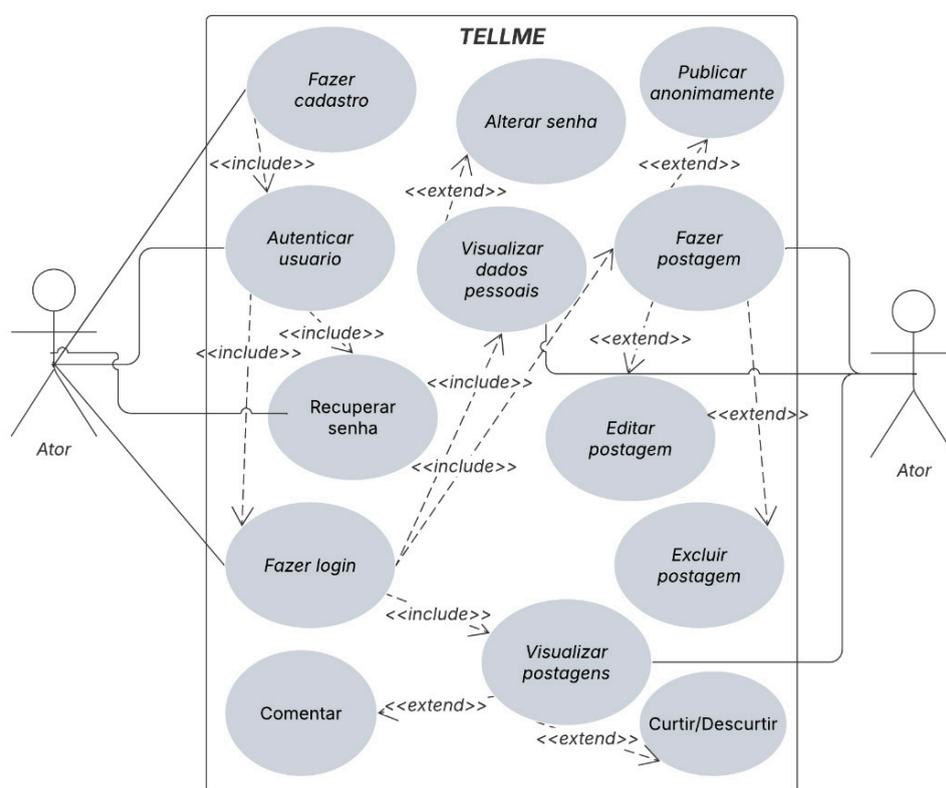
dificuldades e fazer sugestões para a melhoria nas IES, proporcionando uma melhor experiência acadêmica para os atuais e futuros alunos. Os requisitos funcionais propostos são os seguintes:

- **Gerenciamento de usuários:** a ferramenta deve permitir ao usuário efetuar o cadastro no sistema, informando o email e senha. Além disso, a aplicação deve permitir que o usuário acesse a ferramenta (por meio de um processo de autenticação) e realize a alteração da senha cadastrada. Esse requisito tem como finalidade permitir que o usuário gerencie as informações da sua conta na ferramenta;
- **Recuperação de senha:** a ferramenta deve disponibilizar ao usuário a opção de recuperação da senha cadastrada no sistema. Nesse requisito, o usuário poderá recuperar sua senha caso tenha esquecido e assim não ficará sem utilizar a aplicação;
- **Gerenciar feedback/reclamação:** a ferramenta deve oferecer ao usuário a possibilidade de inserir feedbacks ou reclamações, bem como a opção de postar uma publicação de forma anônima ou pública. Além disso, deve permitir ao usuário a visualização das próprias publicações feitas no sistema e as funcionalidades de edição e exclusão das postagens realizadas, sejam elas feedbacks ou reclamações. Esse requisito oferece um controle das publicações postadas pelo usuário e disponibiliza a correção de algum erro cometido pelo usuário no texto postado, seja por meio de edição ou exclusão;
- **Visualizar feedback/reclamação de outros usuários:** a ferramenta deve permitir ao usuário a visualização dos feedbacks e reclamações feitos por outros usuários do sistema;
- **Reforçar e comentar publicações:** a ferramenta deve permitir ao usuário apoiar feedbacks ou reclamações postadas por outros usuários do sistema, através de botões *likes*, *deslikes* e/ou comentários. Esse requisito tem como finalidade reforçar um determinado assunto para que ele seja visualizado no topo das postagens e, assim, caso seja um problema postado pelo usuário, poderá ser solucionado o mais rápido possível.

### 4.3 Diagrama de Casos de uso

A Figura 3 ilustra o diagrama de casos de uso, proporcionando uma melhor compreensão da execução de cada funcionalidade durante a fase de desenvolvimento. E, para uma análise mais detalhada do funcionamento da aplicação, foram selecionadas os casos de uso para representar os requisitos do sistema.

Figura 3 - Diagrama de Casos de uso



Fonte: Autor (2025)

### 4.4 Casos de testes

Os Casos de teste são cenários planejados para verificar se as funcionalidades de um sistema atendem aos requisitos definidos. Eles simulam diferentes situações de uso, validando desde ações básicas, como login e cadastro, até interações mais complexas, como criação e gerenciamento de publicações. Estes testes garantem a qualidade da aplicação, identificando possíveis falhas e

assegurando uma experiência estável e segura para os usuários. No quadro 4 apresenta os casos de testes relacionados às principais funcionalidades do sistema.

*Quadro 4 - Casos de teste*

<b>ID</b>	<b>Caso de Teste</b>	<b>Descrição</b>	<b>Passos</b>	<b>Entrada Esperada</b>	<b>Resultado Esperado</b>
<b>CT01</b>	Cadastro de Usuário	Verificar se o usuário pode se cadastrar no sistema.	<ol style="list-style-type: none"> <li>1. Acessar a tela de cadastro.</li> <li>2. Informar e-mail e senha válidos.</li> <li>3. Confirmar o cadastro.</li> </ol>	E-mail e senha válidos.	Conta criada com sucesso e usuário pode fazer login.
<b>CT02</b>	Login do Usuário	Verificar se o usuário pode acessar a conta.	<ol style="list-style-type: none"> <li>1. Acessar a tela de login.</li> <li>2. Informar e-mail e senha cadastrados.</li> <li>3. Confirmar o login.</li> </ol>	E-mail e senha corretos.	Usuário acessa o sistema com sucesso.
<b>CT03</b>	Recuperação de Senha	Verificar se o usuário pode recuperar a senha esquecida.	<ol style="list-style-type: none"> <li>1. Acessar a tela de login.</li> <li>2. Selecionar "Esqueci minha senha".</li> <li>3. Informar e-mail cadastrado.</li> <li>4. Aguardar o link de redefinição por e-mail.</li> <li>5. Definir nova senha.</li> </ol>	E-mail cadastrado no sistema.	Usuário recebe e-mail e redefine a senha com sucesso.
<b>CT04</b>	Criar Publicação	Verificar se o usuário pode criar um feedback ou reclamação.	<ol style="list-style-type: none"> <li>1. Acessar a área de publicações.</li> <li>2. Escrever uma publicação.</li> <li>3. Escolher entre anônima ou pública.</li> <li>4. Confirmar a publicação.</li> </ol>	Texto válido.	Publicação salva e exibida no feed.

*Fonte: Autor (2025)*

## **4.5 O banco de dados**

Esta seção descreve o processo de modelagem e criação do banco de dados da ferramenta proposta por este TCC.

### **4.5.1 Modelo de Banco de Dados**

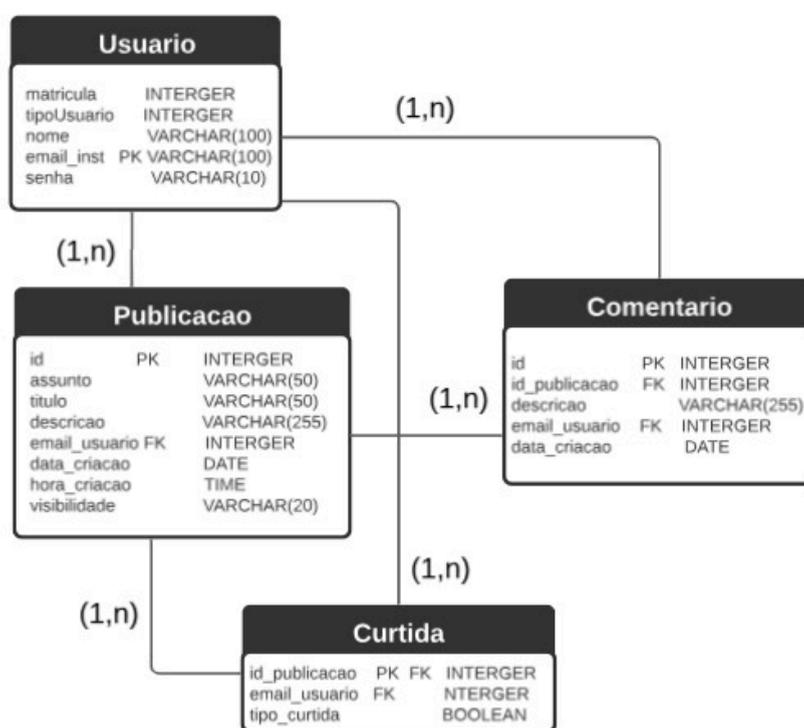
É muito importante a utilização de um banco de dados para uma aplicação, visto que os dados precisam estar armazenados em alguma base de dados para a utilização futura do sistema. Dessa forma, após definir tal importância em usar um banco de dados, veio a necessidade de se definir qual banco de dados utilizar com

o intuito de facilitar a manipulação desses dados. Para essa aplicação foi decidido utilizar o *SQLite*<sup>9</sup>. O *SQLite* consiste em um banco de dados baseado em arquivo, no qual todos os dados ficam armazenados em um arquivo com a extensão “.sqlite”. Dessa forma, esse tipo de banco de dados relacional não necessita de um SGBD<sup>10</sup> (Sistema de Gestão de Bases de Dados) para manipular os dados, deixando a biblioteca encarregada de gerenciar o acesso aos dados. Além disso, o *SQLite* oferece uma alta compatibilidade, já que apresenta um ótimo funcionamento em qualquer tipo de sistema operacional.

#### 4.5.2 Modelagem de Dados

O esquema lógico utilizado para a implementação do banco de dados da aplicação é mostrado na Figura 4. Conforme pode ser observado nessa figura, o esquema é dividido em quatro tabelas: *Usuario*, *Publicacao*, *Comentario* e *Curtidas*.

Figura 4 - Modelagem do esquema lógico



Fonte: Autor (2025)

<sup>9</sup> <https://www.sqlite.org/index.html>

<sup>10</sup>

A tabela *Usuario* armazena os dados sobre os usuários que vão utilizar a aplicação. Para cada usuário, são armazenadas as seguintes informações: o *nome* (relacionado ao nome do usuário no sistema), o *email\_inst* (chave primária que representa o email acadêmico do usuário), a *senha* (referente à senha cadastrada no sistema pelo usuário), a *matricula* (atributo no qual o usuário insere a sua matrícula acadêmica) e o *tipoUsuario* (atributo que define o tipo de usuário que está acessando o sistema).

Na tabela *Publicacao* são armazenados os dados relacionados às publicações postadas no sistema. Para cada publicação é guardado: o *id* (chave primária para representar uma publicação), o *assunto* (está relacionado ao assunto da publicação, podendo ser sobre a instituição, auxílios estudantis, acessibilidade, disciplina, entre outros), o *titulo* (está relacionado ao título da publicação), a *descricao* (refere-se ao texto inserido pelo usuário), o *email\_usuario* (chave estrangeira relacionada ao usuário que fez a publicação), a *data\_criacao* (atributo que irá guardar a data em que foi postada a publicação), a *hora\_criacao* (atributo que irá guardar a hora em que foi postada a publicação) e a *visibilidade* (relaciona a opção e permite ao usuário optar por realizar a publicação de forma anônima ou não).

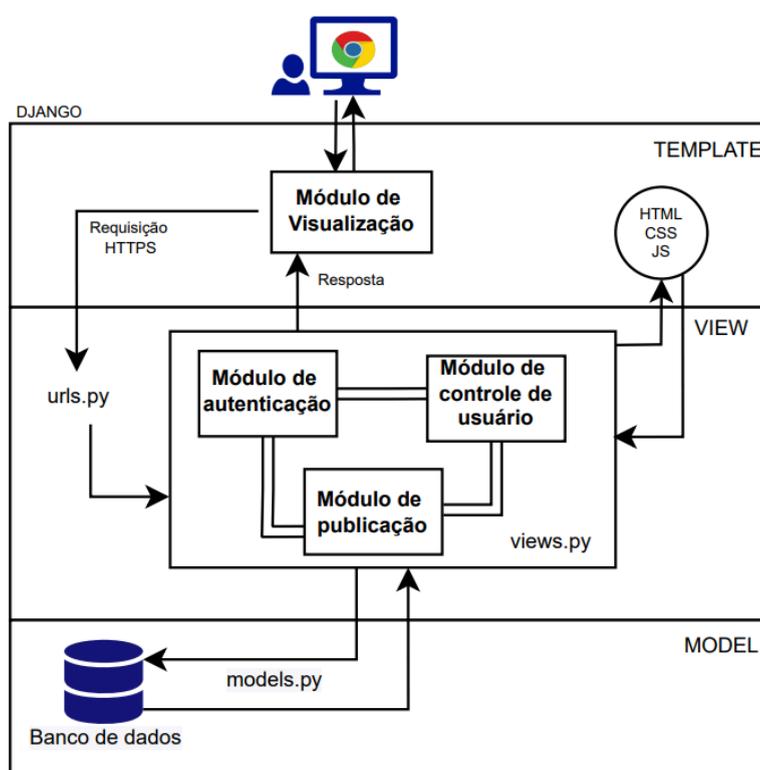
A tabela *Comentario* contém dados referentes aos comentários deixados nas publicações pelos usuários da aplicação. Cada comentário contém: *id* (chave primária, está atribuído o valor para identificar cada comentário), *id\_publicacao* (chave estrangeira, este atributo representa a identificação da publicação que receberá os comentários), *descricao* (relacionada ao texto inserido pelos usuários com base em cada publicação), *email\_usuario* (chave estrangeira, esse atributo está relacionado ao usuário que postou o comentário em uma determinada publicação) e *data\_criacao* (atributo que irá guardar a data em que foi postado o comentário).

Na tabela *Curtidas* encontram-se as informações sobre os *likes* deixados pelos usuários da ferramenta. Para cada curtida é armazenado: o *id\_publicacao* (chave primária e estrangeira, esse atributo está relacionado à publicação que recebeu o like), o *email\_usuario* (chave estrangeira, refere-se à identificação do usuário que deixou o like) e *tipo\_curtida* (nesse atributo é guardado qual reação o usuário deixou na publicação, verdadeiro para Curtir ou Falso para Descurtir).

#### 4.6 Arquitetura da Ferramenta

Esta seção descreve a arquitetura definida para a ferramenta desenvolvida, apresentando as suas camadas e as relações que existem entre elas. Essa arquitetura é mostrada na Figura 5. A arquitetura de software é muito importante para que seja exibida a maneira como as partes do sistema estão organizadas, obtendo como resultado a facilidade de manutenção da aplicação, já que essa organização proporciona uma melhor compreensão dos seus componentes, além de melhorar a portabilidade e a flexibilidade do sistema.

Figura 5 - Arquitetura do sistema



Fonte: Autor (2025)

Ao se observar a Figura 5, é possível notar que a arquitetura é dividida em três camadas: *template*, *view* e *model*, seguindo o padrão MTV apresentado na seção 3.2. A camada *Template* é responsável pela interação com o usuário, que ocorre por meio do módulo de visualização. Nela ficam armazenados os arquivos HTML, CSS e JavaScript usados para a renderização das páginas web que serão

apresentadas aos usuários. Para acessar a aplicação, o usuário, por meio de um navegador, solicita uma requisição HTTPS (*Hypertext Transfer Protocol Secure*) ao servidor *web*. Essa requisição é feita através de uma rota, e neste momento verifica-se a existência da *URL* inserida no script de *urls.py*.

A camada *View* é responsável pelo processamento das requisições e pela implementação das regras de negócio. Ela é composta por três módulos:

- **módulo de autenticação:** está relacionado à confirmação dos dados cadastrados pelo usuário para realizar a sua autenticação e assim ter acesso às funcionalidades da ferramenta;
- **módulo de publicação:** neste módulo se encontram todas as ações que podem ser executadas referentes às publicações registradas no sistema;
- **módulo de controle de usuário:** neste módulo é realizado o gerenciamento dos usuários inseridos na aplicação.

A camada *Model* é responsável pelo acesso ao banco de dados, armazenando e recuperando as informações solicitadas nas requisições. Essas informações são enviadas para a camada *View* para que elas sejam formatadas e renderizadas pela camada *Templates* para visualização dos usuários.

No *Django*, cada *model* representa uma tabela no banco de dados, que no script é representado por classes estendendo uma classe principal chamada *models*. Cada *model* contém atributos que representam as colunas de uma tabela do banco de dados. Esses atributos são instâncias de classes específicas do *Django*, como *CharField* para armazenar texto curto, *IntegerField* para números inteiros, *BooleanField* para valores booleanos, *DateField* para datas, entre outros.

Na construção da aplicação, foi utilizado um conjunto de *models* que estruturam o armazenamento e a manipulação dos dados de maneira eficiente. Cada modelo desempenha um papel essencial na dinâmica da plataforma. Eles foram projetados seguindo as melhores práticas do *Django*, garantindo integridade e escalabilidade. Durante o desenvolvimento da aplicação foram implementados os seguintes modelos:

- **Usuário:** este modelo é a base do sistema de autenticação e gerenciamento de usuários. Ele substitui o modelo padrão do *Django*, estendendo as classes *AbstractBaseUser* e *PermissionsMixin*. A autenticação é feita com base no *e-mail*, garantindo maior flexibilidade.

Além dos atributos essenciais como matrícula, username e email, o modelo inclui um sistema de verificação de e-mail, através do atributo *email\_verified* e um *verification\_token*. A classe *CustomUserManager* gerencia a criação de usuários e superusuários, assegurando que as regras de negócio sejam respeitadas ao longo do cadastro;

- **Postagem:** este modelo representa as publicações feitas pelos usuários. Cada postagem pode estar associada a uma categoria específica, como acessibilidade, auxílio estudantil ou disciplina. Além do título e da descrição, há um atributo anônimo, que permite ao usuário publicar sem revelar sua identidade. Também foram adicionados contadores de curtidas positivas e negativas, permitindo um sistema de avaliação da relevância do conteúdo. O relacionamento com o usuário é feito através de uma chave estrangeira, garantindo que cada postagem seja corretamente associada ao seu autor;
- **Curtida:** este modelo foi implementado para permitir interações diretas dos usuários com as postagens. Ele possibilita que cada usuário possa curtir ou descurtir uma postagem, registrando se a curtida foi positiva ou negativa. Para evitar curtidas duplicadas, foi aplicada a restrição *unique\_together*, garantindo que um mesmo usuário não possa curtir ou descurtir uma postagem mais de uma vez. Esse modelo melhora a estrutura de armazenamento, permitindo consultas e análises mais detalhadas;
- **Comentário:** este modelo também permite que os usuários interajam diretamente com as postagens, deixando mensagens e opiniões sobre os conteúdos compartilhados. Cada comentário é vinculado a uma postagem e a um usuário, garantindo a rastreabilidade das interações. O atributo *texto* armazena o conteúdo do comentário, enquanto o atributo *data\_criacao* registra o momento exato da publicação. A implementação desse modelo promove uma maior interação entre os usuários da plataforma.

#### 4.7 Implementação da Ferramenta

Para o desenvolvimento da aplicação *TellMe*, foi escolhida a linguagem de programação *Python*, juntamente com o framework *Django*. Além disso, a IDE

*PyCharm* foi utilizada como ambiente de desenvolvimento, oferecendo ferramentas avançadas como autocompletar código, depuração eficiente e integração com controle de versão, o que torna o processo de programação mais produtivo e intuitivo. As próximas seções descrevem a implementação dos componentes da arquitetura apresentada na Figura 5. Nelas, são explorados os módulos representados na imagem da arquitetura da ferramenta, destacando as tecnologias empregadas e explicando o papel e o funcionamento de cada um deles.

#### 4.7.1 Módulo de Autenticação e controle de Usuário

O Algoritmo 2 mostra um trecho do código da *view* de cadastro do usuário. Ele pertence à classe *CreateView* do *Django* e processa um formulário de criação de usuário, garantindo que o e-mail informado pertença ao domínio *@academico.ifpb.edu.br*. Caso essa condição não aconteça, ele exibe um erro e impede o envio. Caso contrário, o usuário é criado sem ser armazenado imediatamente, a senha é criptografada e o atributo *email\_verified* é definido como *False*. Após criar o usuário no banco, um link de verificação é gerado com um token exclusivo, permitindo que o usuário confirme seu e-mail antes de ativar a conta.

##### *Algoritmo 2 - Trecho do código da view de cadastro do usuário*

```
def form_valid(self, form):
    email = form.cleaned_data['email']
    if not email.endswith('@academico.ifpb.edu.br'):
        form.add_error('email', 'O e-mail deve pertencer ao domínio
@academico.ifpb.edu.br.')
        return self.form_invalid(form)

    usuario = form.save(commit=False)
    senha = form.cleaned_data['password']
    usuario.set_password(senha)

    usuario.email_verified = False
    usuario.save()

    verification_link = self.request.build_absolute_uri(
        reverse('verificar_email',
args=[usuario.verification_token])
    )
```

*Fonte: Autor (2025)*

O Algoritmo 3 apresenta um trecho do código da *view* de *login*. Nele, é feita a autenticação de um usuário com base no e-mail e senha fornecidos. Se as credenciais estiverem corretas, é verificado se o e-mail já foi confirmado. Caso contrário, o usuário é desconectado, recebe uma mensagem de aviso e é redirecionado para a página de login. Se o e-mail estiver verificado, o login é realizado com sucesso, e o usuário é redirecionado para a página inicial. Caso as credenciais estejam incorretas, uma mensagem de erro é exibida, e o usuário retorna à tela de login.

### *Algoritmo 3 - Trecho do código da view de login*

```
def form_valid(self, form):
    email = form.cleaned_data['email']
    senha = form.cleaned_data['senha']

    usuario = authenticate(self.request, email=email,
password=senha)

    if usuario is not None:
        if not usuario.email_verified:
            logout(self.request)
            messages.warning(
                self.request,
                'Por favor, verifique seu e-mail antes de fazer
login.'
            )
            messages.info(
                self.request,
                email
            )
            return redirect('login')

        login(self.request, usuario)
        return redirect('index')
    else:
        messages.error(self.request, 'E-mail ou senha incorretos.')
        return redirect('login')
```

*Fonte: Autor (2025)*

#### 4.7.2 Módulo de Publicação

O Módulo de Publicação é responsável por gerenciar todas as ações relacionadas às publicações dentro do sistema. Esse módulo permite que os usuários criem, editem, excluam e visualizem publicações, garantindo um fluxo

eficiente de compartilhamento de informações. Ele também possibilita interações, como comentários e reações, promovendo um maior engajamento entre os usuários.

No Algoritmo 4 observa-se como está implementada a *view* de criação de postagens. A classe *CriarPostagemView* é uma *view* baseada em classe que permite a criação de postagens, garantindo que apenas usuários autenticados possam acessá-la. Antes de salvar a postagem, o método *form\_valid* verifica se o título ou a descrição contém linguagem inadequada, utilizando a função *verificar\_linguagem()*. A função *verificar\_linguagem* aciona o modelo e solicita que ele verifique se o texto contém palavras ofensivas. O modelo responde com "sim" ou "não", e a função retorna *true* se a resposta for "sim" (indicando linguagem imprópria) ou *false* caso contrário. Caso conteúdo impróprio seja detectado, uma mensagem de erro é exibida ao usuário e a postagem não é salva. Caso contrário, a postagem é atribuída ao usuário logado e processada normalmente.

#### Algoritmo 4 - View de Criar Postagem

```
class CriarPostagemView(LoginRequiredMixin, CreateView):
    model = Postagem
    form_class = PostagemForm
    template_name = 'post.html'
    success_url = reverse_lazy('index')
    login_url = reverse_lazy('index')
    def form_valid(self, form):
        """
        Antes de salvar a postagem, verificamos se o título ou a
        descrição contém linguagem inadequada.
        """
        titulo = form.cleaned_data.get('titulo')
        descricao = form.cleaned_data.get('descricao')
        textoTotal = titulo + descricao
        if verificar_linguagem(textoTotal):
            print("improprio")
            messages.error(self.request, "A postagem contém
            linguagem imprópria e não pode ser publicada.")
            return render(self.request, self.template_name,
            {'form': form})
        form.instance.usuario = self.request.user
        return super().form_valid(form)
    def form_invalid(self, form):
        print("erro ao criar")
```

Fonte: Autor (2025)

O Código 5 refere-se à função *excluir\_postagem*. Ele recebe uma requisição e um *post\_id*, recupera a postagem correspondente pertencente ao usuário que fez requisição e a exclui, redirecionando-o para a página de seus posts (*meusPosts*).

#### Algoritmo 5 - Função de excluir postagem

```
def excluir_postagem(request, post_id):
    postagem = get_object_or_404(Postagem, id=post_id,
    usuario=request.user)
    postagem.delete()
    return redirect('meusPosts')
```

*Fonte: Autor (2025)*

Já no Algoritmo 6 é apresentada a classe editar postagem, que permite ao usuário autenticado editar uma postagem existente. Ela utiliza a *UpdateView* do *Django* para exibir um formulário de edição com os campos título e descrição, renderizando a página *meusPosts.html*. Antes de salvar os dados, ele verifica se o texto contém linguagem imprópria usando a função *verificar\_linguagem*. Se encontrar linguagem imprópria, a edição é cancelada e um erro é retornado. Caso contrário, o usuário é associado à postagem e ela é salva com sucesso, retornando uma resposta JSON indicando sucesso.

#### Algoritmo 6 - View editar postagem

```
class EditarPostagemView(LoginRequiredMixin, UpdateView):
    model = Postagem
    fields = ['titulo', 'descricao']
    template_name = 'meusPosts.html'
    success_url = reverse_lazy('meusPosts')
    def form_valid(self, form):
        titulo = form.cleaned_data.get('titulo')
        descricao = form.cleaned_data.get('descricao')
        textoTotal = titulo + " " + descricao
        if verificar_linguagem(textoTotal):
            return JsonResponse({"success": False, "error": "A
postagem contém linguagem imprópria e não pode ser editada."})
        form.instance.usuario = self.request.user
        self.object = form.save()
        return JsonResponse({"success": True})
```

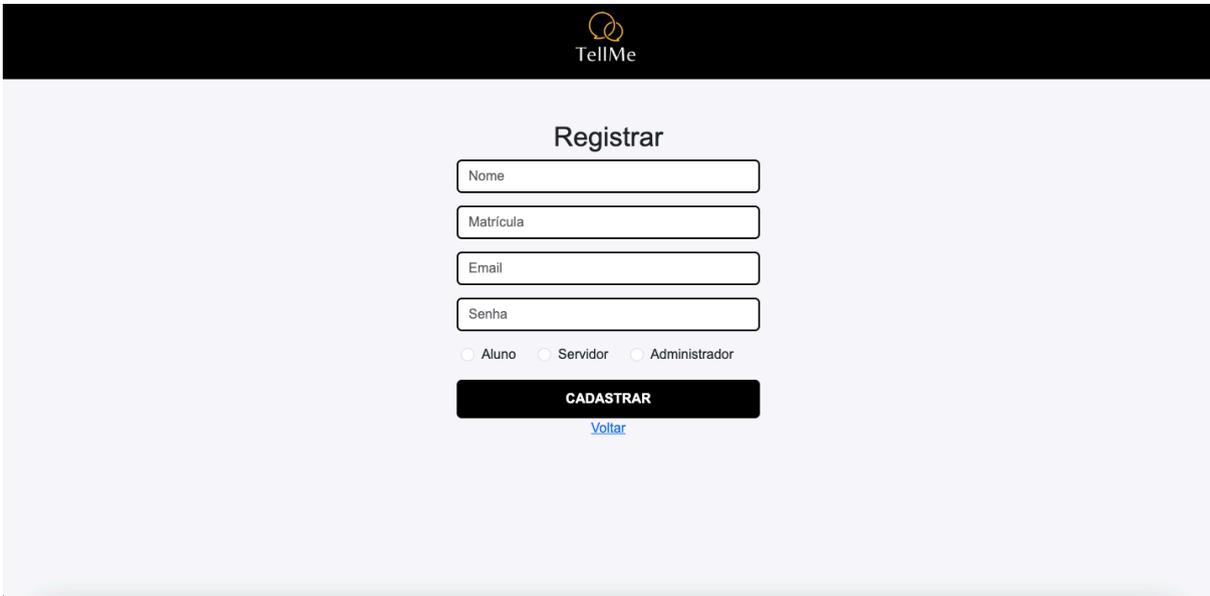
*Fonte: Autor (2025)*

### 4.7.3 Módulo de Visualização

O módulo de visualização é responsável pela geração da interface de usuário da aplicação, ou seja, pelas telas que permitem a interação e a navegação dentro da ferramenta. Esse módulo é fundamental para garantir uma experiência de usabilidade intuitiva e eficiente, permitindo que o usuário execute suas tarefas de forma simples e ágil. Através desse módulo, o usuário pode acessar diferentes funcionalidades da aplicação, visualizar informações, fazer ajustes e interagir com os demais módulos do sistema. Além disso, a interface foi projetada para ser responsiva, oferecendo uma navegação fluida e agradável em diversos dispositivos, como desktops, tablets e smartphones.

A Figura 6 mostra a implementação da tela de registro, que possibilita ao usuário criar uma conta para acessar a aplicação, inserindo nome, matrícula, email, senha e o tipo de usuário. Depois de concluir o cadastro, é enviado um link de verificação para o email informado, para garantir que o usuário informe os dados corretos e assim possa usar as credenciais registradas para fazer login na aplicação.

*Figura 6 - Tela de fazer registro*



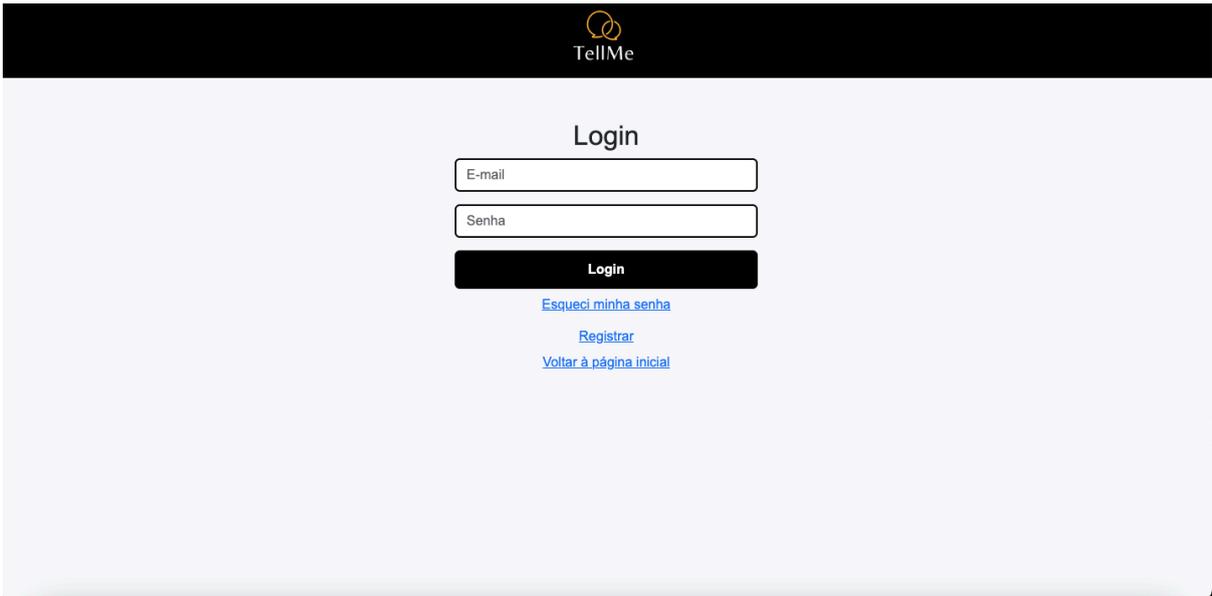
A imagem mostra a tela de registro da aplicação TellMe. No topo, há um cabeçalho preto com o logo TellMe em amarelo. Abaixo, o título "Registrar" está centralizado. O formulário contém quatro campos de entrada: "Nome", "Matrícula", "Email" e "Senha". Abaixo dos campos, há três opções de usuário: "Aluno", "Servidor" e "Administrador", cada uma com um botão de opção desativado. Um botão preto com o texto "CADASTRAR" em branco está centralizado. Abaixo do botão, há um link azul "Voltar".

*Fonte: Autor (2025)*

A Figura 7 apresenta a implementação da tela de *login*. Nela, o usuário registrado pode inserir as suas credenciais (email e senha) para acessar o sistema

e ter acesso às suas funcionalidades. Após o *login*, ele é automaticamente encaminhado para a tela inicial. Nessa tela é oferecido o botão para recuperação de senha. Como mostra a Figura 8, os usuários registrados podem recuperar a senha de acesso informando apenas o email registrado na aplicação e assim poderão adicionar uma nova senha.

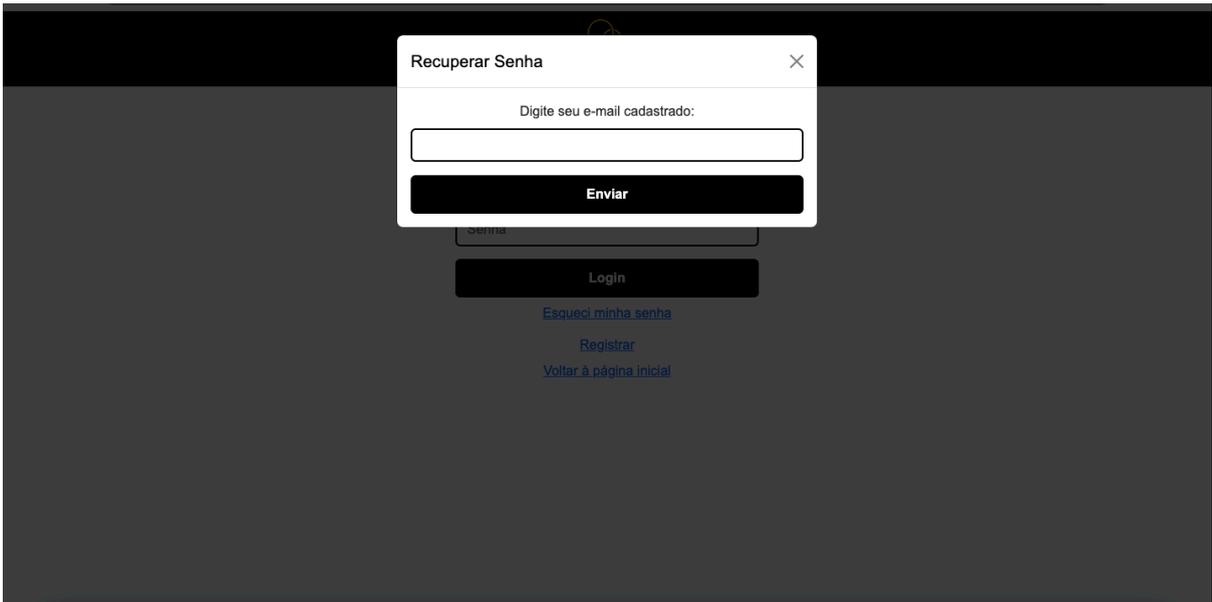
*Figura 7 - Tela de login*



A imagem mostra a tela de login do sistema TellMe. No topo, há um cabeçalho preto com o logo TellMe em amarelo e branco. Abaixo, o título "Login" é centralizado. Há dois campos de entrada: "E-mail" e "Senha". Abaixo dos campos, há um botão "Login" em um fundo preto com o texto em branco. Abaixo do botão, há três links azuis: "Esqueci minha senha", "Registrar" e "Voltar à página inicial".

*Fonte: Autor (2025)*

*Figura 8 - Tela de recuperação de senha*

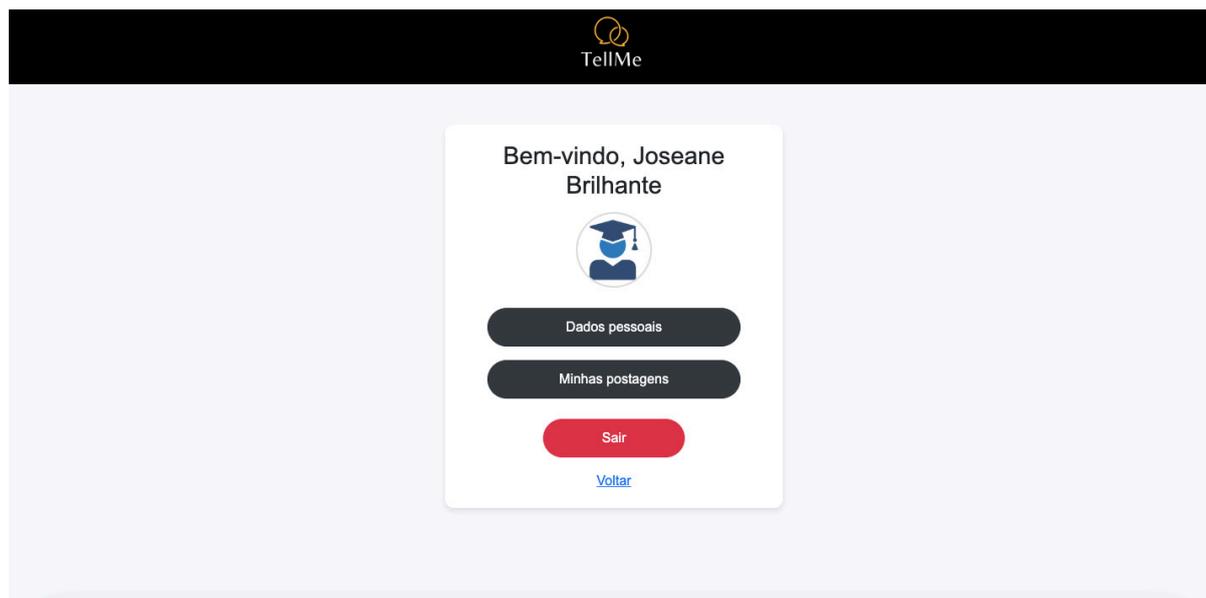


A imagem mostra a tela de recuperação de senha do sistema TellMe. No topo, há um cabeçalho preto com o logo TellMe em amarelo e branco. Abaixo, o título "Recuperar Senha" é centralizado. Há um campo de entrada para o e-mail cadastrado. Abaixo do campo, há um botão "Enviar" em um fundo preto com o texto em branco. Abaixo do botão, há três links azuis: "Esqueci minha senha", "Registrar" e "Voltar à página inicial".

*Fonte: Autor (2025)*

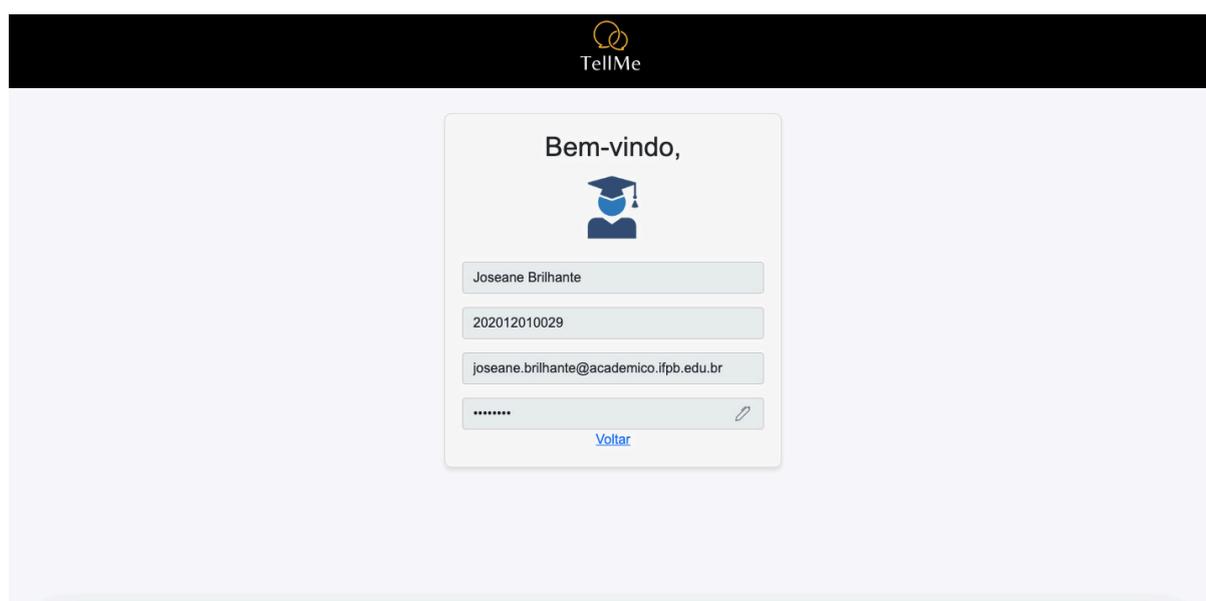
A Figura 9 apresenta a tela de perfil do usuário, onde o usuário da aplicação pode visualizar os seus dados pessoais, bem como as suas postagens. Ao clicar no botão dos dados pessoais, o usuário visualiza os dados inseridos na aplicação, nome, matrícula, email e senha, como pode ser visto na Figura 10, podendo ser alterada a senha, como mostra a Figura 11.

*Figura 9 - Tela de perfil do usuário*



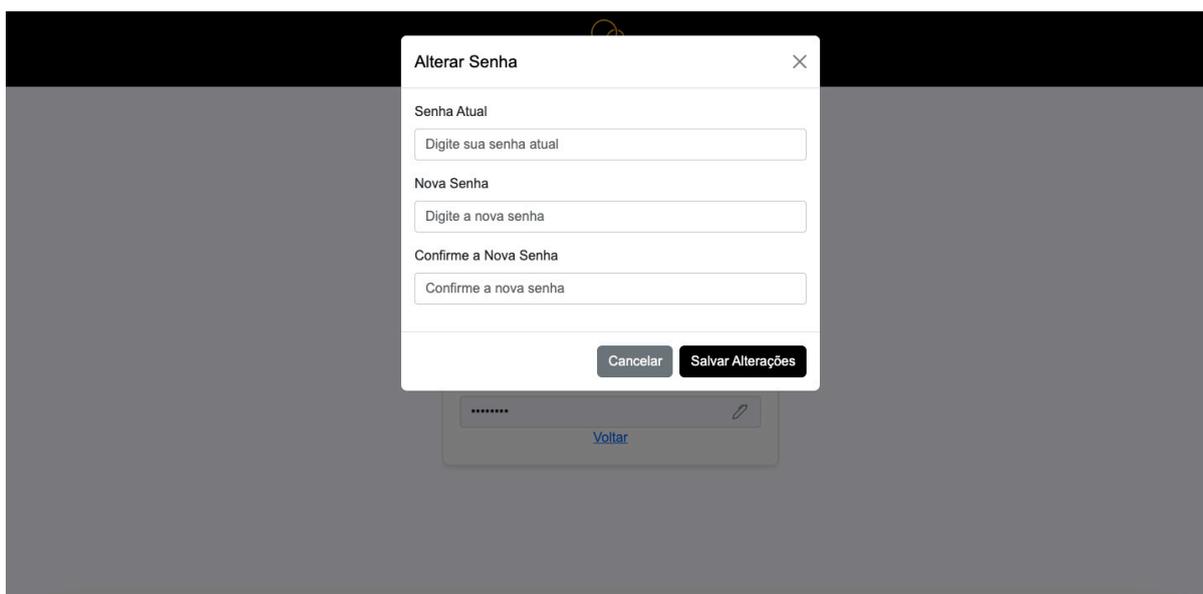
*Fonte: Autor (2025)*

*Figura 10 - Tela de alteração de dados*



*Fonte: Autor (2025)*

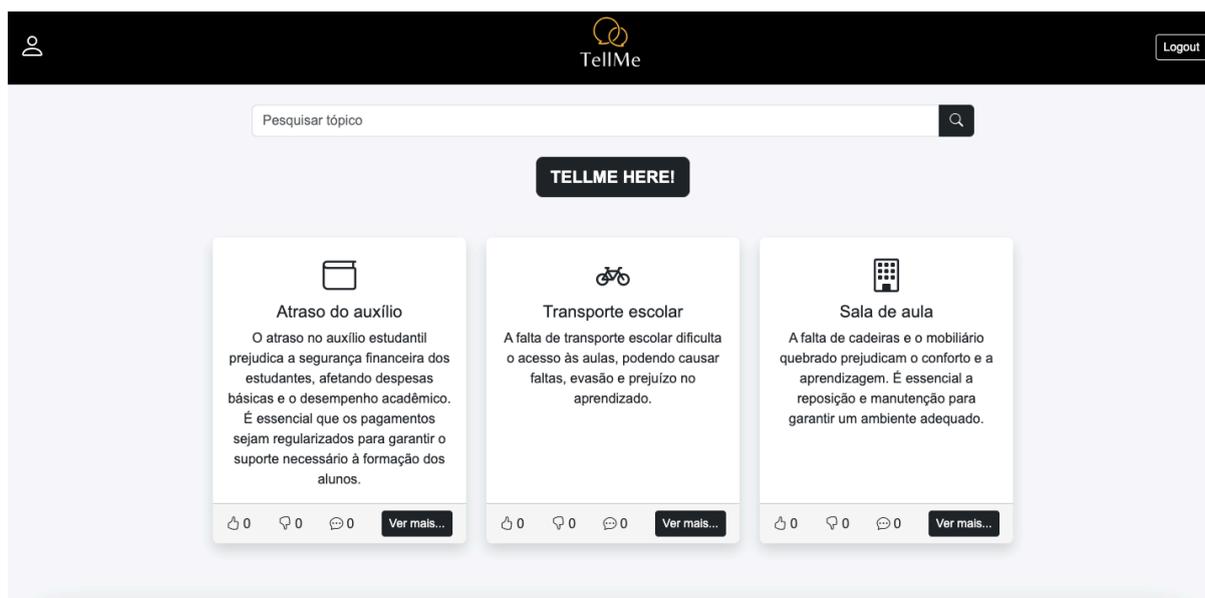
Figura 11 - Tela de alteração de senha



Fonte: Autor (2025)

Na Figura 12 é exibida a tela inicial da aplicação, onde aparecem as publicações feitas por todos os usuários, um botão para fazer uma nova postagem, uma barra de busca, um ícone de perfil que, ao selecionar, o usuário vai visualizar os seus dados e a opção logout para sair da aplicação.

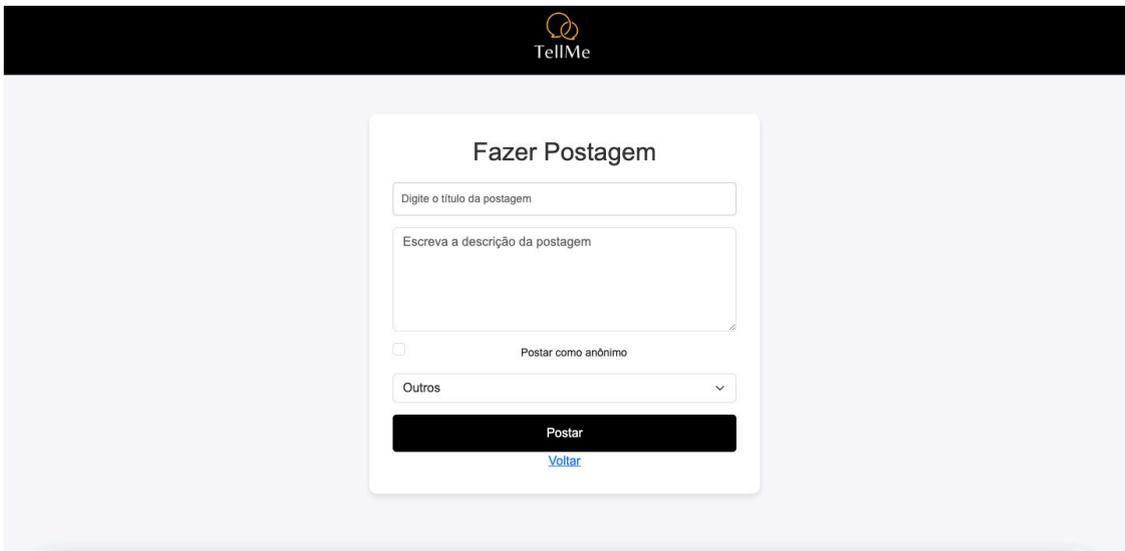
Figura 12 - Tela inicial da aplicação



Fonte: Autor (2025)

A tela de fazer postagem é apresentada na Figura 13. Nela, o usuário pode adicionar um título, uma descrição, o tema ao qual a postagem está relacionada e selecionar se a postagem vai ser feita de forma anônima ou não.

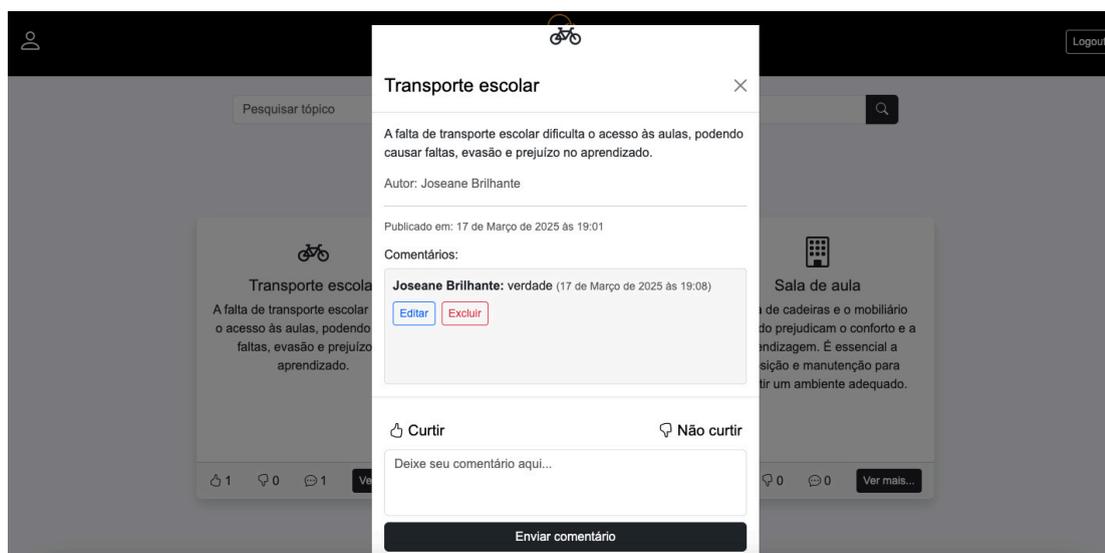
*Figura 13 - Tela de fazer postagem*



*Fonte: Autor (2025)*

A Figura 14 mostra a tela na qual o usuário pode atribuir as reações de curtidas e descurtidas, adicionar um comentário, excluir ou editar um comentário postado.

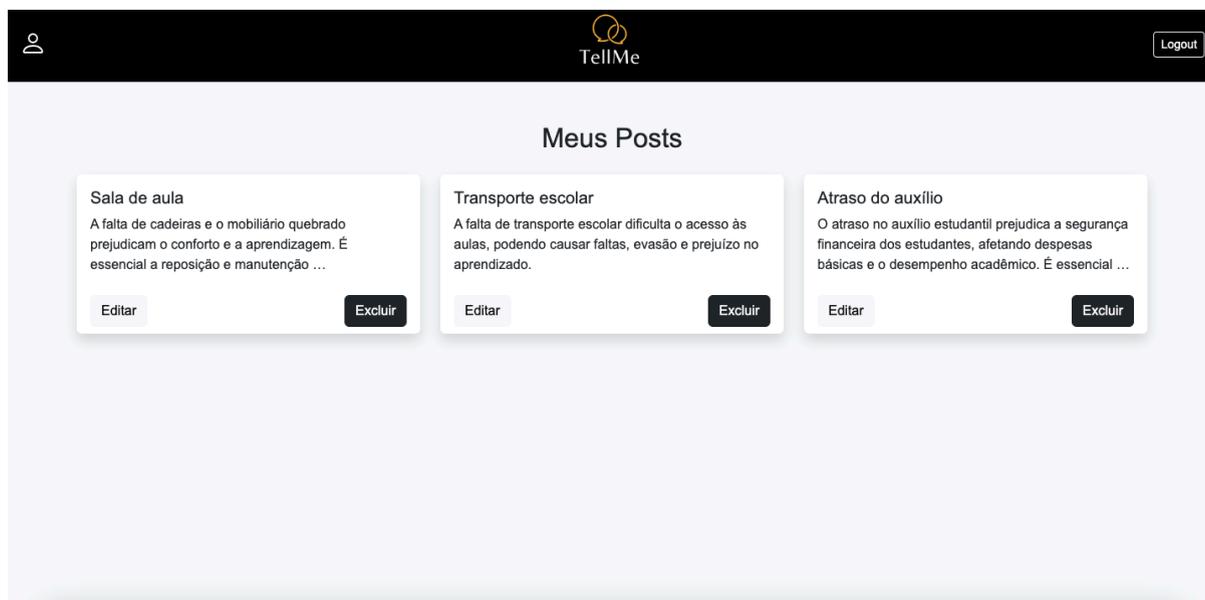
*Figura 14 - Tela de deixar reações: curtir, descurtir e comentar*



*Fonte: Autor (2025)*

Por fim, a Figura 15 apresenta a tela de visualização de postagens do usuário, na qual é permitido que seja feita a edição e exclusão de publicações feitas pelo usuário.

*Figura 15 - Tela de visualização de postagens do usuário*



*Fonte: Autor (2025)*

## 5. CONCLUSÃO

A comunicação nas Instituições de ensino é essencial para o acompanhamento dos serviços prestados pela instituição, uma vez que, em um ambiente com vários docentes, discentes e técnicos administrativos, podem surgir problemas, sejam eles relacionados às partições do instituto, pessoais ou coletivos. E, mesmo com as IES disponibilizando em seus portais recursos de comunicações, observa-se insatisfações no uso desses meios, já que em alguns casos pode haver demora na resposta ou até mesmo nem se obter um retorno para um determinado problema. Em outros casos, o aluno pode não ter conhecimento desses mecanismos de comunicação, ou então, eles podem não ser de fácil acesso.

Com intuito de oferecer uma melhor forma de comunicação entre o aluno e a Instituição, foi proposto neste trabalho uma aplicação denominada *TelMe*. A ferramenta propõe agilidade e rapidez no descobrimento de problemas enfrentados nas IES, já que o aluno, ao passar por alguma dificuldade ou constatar alguma irregularidade na instituição, pode usar a aplicação para informar de forma imediata o ocorrido.

Os objetivos da aplicação *TelMe* foram plenamente alcançados por meio do desenvolvimento de uma plataforma intuitiva e acessível, projetada para que estudantes e colaboradores possam registrar suas dificuldades de forma estruturada e eficiente. Destacando-se pela facilidade de acesso e uso, isso por ser acessível diretamente pelo navegador, dispensa a necessidade de instalação de aplicativos, permitindo que o usuário registre reclamações, relate irregularidades ou compartilhe feedbacks positivos de forma rápida e prática. Além disso, a possibilidade de visualizar, editar e excluir postagens oferece maior controle sobre as informações compartilhadas, promovendo uma gestão mais organizada dos feedbacks.

Para aprimorar a aplicação proposta, os seguintes desenvolvimentos podem ser realizados: transformação em uma rede social acadêmica mais robusta e interativa; implementação de um sistema de chat para facilitar a comunicação direta entre usuários e setores responsáveis; adição de um recurso de upload de fotos, permitindo que os relatos sejam acompanhados de evidências visuais; e criação de um mecanismo de encaminhamento automático das postagens ao setor correspondente ao problema relatado.

## REFERÊNCIAS

ABADI, Martín et al. TensorFlow: A System for Large-Scale Machine Learning. [S.l.], 2016. Disponível em: <https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi>. Acesso em: 12 mar. 2025.

ALMAZROUEI, Ebtessam et al. The Falcon Series of Open Language Models. 2023. Disponível em: <https://arxiv.org/abs/2311.16867>. Acesso em: 15 mar. 2025.

ALMEIDA, Mariangela Lima de; SILVA, Nazareth Vidal da; FRANÇA, Bárbara Rebecca Baumgartem; REIS, Marcela Lemos Leal. Formação continuada na perspectiva inclusiva pela via da pesquisa e extensão: as funções mediadoras da relação teoria e prática na constituição de grupos de estudo-reflexão. 2023. Disponível em: <https://www.scielo.br/j/rbeped/a/vhxMmYmhpQgYb7FSp8wxNkk/>. Acesso em: 23 mar. 2025.

ASLAM, Fankar Armash; MOHAMMED, Hawa Nabeel; LOKHANDE, P. S. Efficient Way Of Web Development Using Python And Flask. International Journal of Advanced Research in Computer Science, v. 6, n. 2, p. 54, 2015. Disponível em: [https://openurl.ebsco.com/EPDB%3Agcd%3A16%3A15329632/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A102304622&cr=c&link\\_origin=scholar.google.com](https://openurl.ebsco.com/EPDB%3Agcd%3A16%3A15329632/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A102304622&cr=c&link_origin=scholar.google.com). Acesso em: 13 mar. 2025.

BHUIYAN, S.; ISACHENKO, T. Getting Started with Generative AI: Learn How to Build Your Own AI Application Step-by-Step. [S.l.]: [s.n.], 2022. Disponível em: <https://www.amazon.com.br/Getting-started-Generative-step-hands-ebook/dp/B0DJV2H8K4>. Acesso em: 14 mar. 2025.

DIAS, Ana Cristina Garcia; CARLOTTO, Rodrigo Carvalho; OLIVEIRA, Clarissa Tochetto de; TEIXEIRA, Marco Antônio Pereira. Dificuldades percebidas na transição para a universidade. Revista Brasileira de Orientação Profissional, Porto Alegre, v. 20, n. 1, p. 19-28, 2019. Disponível em: [https://pepsic.bvsalud.org/scielo.php?pid=S1679-33902019000100003&script=sci\\_arttext](https://pepsic.bvsalud.org/scielo.php?pid=S1679-33902019000100003&script=sci_arttext). Acesso em: 23 mar. 2025.

FORCIER, Jeff; BISSEX, Paul; CHUN, Wesley J. Python Web Development with Django. [S.I.]: Addison-Wesley Professional, 2008. Disponível em: <https://books.google.com/books?id=M2D5nnYImZoC>. Acesso em: 13 mar. 2025.

GANAM, Eliana Almeida Soares; PINEZI, Ana Keila Mosca. Desafios da permanência estudantil universitária: um estudo sobre a trajetória de estudantes atendidos por programas de assistência estudantil. Educação e Realidade, Porto Alegre, v. 46, n. 1, p. 143-160, 2021. Disponível em: <https://www.scielo.br/j/edur/a/LXtF95VpbYyzkJTJtkxLrsw/>. Acesso em: 22 mar. 2025.

HARRIS, Charles R. et al. Array Programming with NumPy. [S.I.], 2020. Disponível em: <https://www.nature.com/articles/s41586-020-2649-2>. Acesso em: 11 mar. 2025.

INSTITUTO FEDERAL DA PARAÍBA. NUCA - Núcleo de Combate ao Assédio. Disponível em: <https://www.ifpb.edu.br/patos/assuntos/nuca-nucleo-de-combate-ao-assedio>. Acesso em: 10 fev. 2025.

INEP - Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. Censo Escolar 2023: Apresentação Coletiva. 2023. Disponível em: [https://download.inep.gov.br/censo\\_escolar/resultados/2023/apresentacao\\_coletiva.pdf](https://download.inep.gov.br/censo_escolar/resultados/2023/apresentacao_coletiva.pdf). Acesso em: 3 fev. 2025.

INSTITUTO FEDERAL DA PARAÍBA. Relatório de Autoavaliação 2023. 2023. p. 102-103. Disponível em: <https://www.ifpb.edu.br/cpa/relatorios-de-avaliacao-interna-1/relatorio-de-autoavaliacao-2023.pdf>. Acesso em: 10 mar. 2025.

INTERNATIONAL ENTERPRISE DISTRIBUTED OBJECT COMPUTING CONFERENCE. [S.I.]: IEEE Comput. Soc, 2001. Disponível em: <http://dx.doi.org/10.1109/edoc.2001.950428>. Acesso em: 29 jun. 2022.

LUO, Ziyang et al. WizardCoder: Empowering Code Large Language Models with Evol-Instruct. 2023. Disponível em: <https://arxiv.org/abs/2306.08568>. Acesso em: 15 mar. 2025.

MOZILLA. Django introduction - Learn web development. [S.l.], 2022. Disponível em: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>.

Acesso em: 15 jun. 2022.

OLLAMA. Ollama: Run large language models locally. [S.l.], [s.d.]. Disponível em: <https://ollama.com>. Acesso em: 4 mar. 2025.

RASCHKA, S.; MIRJALILI, V. Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python. [S.l.]: Packt Publishing, 2022. Disponível em: <https://books.google.com.br/books?id=SVxaEAAAQBAJ>. Acesso em: 12 mar. 2025.

ROZIÈRE, Baptiste et al. Code Llama: Open Foundation Models for Code. 2023. Disponível em: <https://arxiv.org/abs/2308.12950>. Acesso em: 13 mar. 2025.

SIINO, Marco et al. Mistral at SemEval-2024 Task 5: Mistral 7B for argument reasoning in Civil Procedure. 2024. Disponível em: <https://aclanthology.org/2024.semeval-1.24/>. Acesso em: 14 mar. 2025.

THOUTAM, V. A Study On Python Web Application Framework. Journal of Electronics, Computer Networking and Applied Mathematics, v. 1, n. 11, p. 48–55, 2021. Disponível em: <https://doi.org/10.55529/jecnam.11.48.55>. Acesso em: 30 jun. 2022.

TOSI, Sandro. Matplotlib for Python Developers. [S.l.]: Packt Publishing, 2009. Disponível em: <https://books.google.com.br/books?id=8Fs3AgAAQBAJ>. Acesso em: 12 mar. 2025.

TOUVRON, Hugo et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. [S.l.], 2023. Disponível em: <https://arxiv.org/abs/2307.09288>. Acesso em: 14 mar. 2025.

VAN ROSSUM, Guido; DRAKE, Fred L. Jr. The Python Language Reference. Release 2.6.4. [S.l.]: Python Software Foundation, 2010. Disponível em: <https://www.cse.unr.edu/~sushil/class/381/notes/python/docs-pdf/reference.pdf>. Acesso em: 11 mar. 2025.

W3SCHOOLS. Django Introduction. [S.l.], 2025. Disponível em:  
[https://www.w3schools.com/django/django\\_intro.php](https://www.w3schools.com/django/django_intro.php). Acesso em: 6 mar. 2025.

	<b>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA</b>
	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)

### Entrega do TCC

<b>Assunto:</b>	Entrega do TCC
<b>Assinado por:</b>	Francisca Brilhante
<b>Tipo do Documento:</b>	Relatório
<b>Situação:</b>	Finalizado
<b>Nível de Acesso:</b>	Ostensivo (Público)
<b>Tipo do Conferência:</b>	Cópia Simples

Documento assinado eletronicamente por:

- Francisca Joseane Paulino Brilhante, ALUNO (202012010029) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS, em 25/03/2025 17:09:21.

Este documento foi armazenado no SUAP em 25/03/2025. 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: 1434145

Código de Autenticação: b07771c78a

