

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

**MECANOSYSTEM: SISTEMA PARA GERENCIAMENTO DE  
SOLICITAÇÕES DE IMPRESSÕES DE DOCUMENTOS DO SETOR DE  
MECANOGRAFIA DO IFPB-CAMPUS CAJAZEIRAS**

**KAÍQUE AUGUSTO DE SOUSA**

**Cajazeiras  
2025**

**KAIQUE AUGUSTO DE SOUSA**

**MECANOSYSTEM: SISTEMA PARA GERENCIAMENTO DE SOLICITAÇÕES DE  
IMPRESSÕES DE DOCUMENTOS DO SETOR DE MECANOGRAFIA DO  
IFPB-CAMPUS CAJAZEIRAS**

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

Orientador

Prof. Me. Fabio Abrantes Diniz.

**Cajazeiras  
2025**

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

S725m Sousa, Kaíque Augusto de.

Mecanostystem: sistema para gerenciamento de solicitações de impressões de documentos do setor da mecanografia do IFPB-Campus Cajazeiras / Kaíque Augusto de Sousa. – Cajazeiras, 2025.  
79f. : il.

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

Orientador: Prof. Me. Fabio Abrantes Diniz.

1. Desenvolvimento de software. 2. Programa de computador. 3. Gestão de documentos. 4. Automatização. I. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.

IFPB/CZ

CDU: 004.4:002(043.2)



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

KAÍQUE AUGUSTO DE SOUSA

**MECANOSYSTEM: SISTEMA PARA GERENCIAMENTO DE SOLICITAÇÕES DE IMPRESSÕES DE DOCUMENTOS DO SETOR DE MECANOGRRAFIA DO IFPB-CAMPUS CAJAZEIRAS**

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

Orientador

Prof. Me. Fábio Abrantes Diniz

Aprovada em: **10 de setembro de 2025.**

Prof. Me. Fábio Abrantes Diniz - Orientador

Prof. Me. Francisco Paulo de Freitas Neto - Avaliador

IFPB - Campus Cajazeiras

Prof. Dr. Fabio Gomes de Andrade - Avaliador

IFPB - Campus Cajazeiras

Documento assinado eletronicamente por:

- **Francisco Paulo de Freitas Neto**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 10/09/2025 16:45:56.
- **Fabio Abrantes Diniz**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 10/09/2025 16:52:05.
- **Fabio Gomes de Andrade**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 10/09/2025 16:52:13.

Este documento foi emitido pelo SUAP em 10/09/2025. 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 764428  
Verificador: 04f03120f3  
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 Deus, pela minha vida, por ter me dado a capacidade de concluir este curso e forças para suportar todas as dificuldades.

A minha mãe Aparecida e meu Pai Raimundo, que sempre me ensinaram os mais importantes valores da vida e sempre a correr atrás das coisas mais importantes, por sempre me motivarem e apoiarem nos meus estudos, e por terem me ajudado a chegar onde eu cheguei.

As minhas irmãs Keliane e Maria Vitória, por também sempre me apoiarem e me incentivado nos estudos, em especial Keliane que acordava cedo todos os dias para me levar ao ponto de ônibus faça chuva ou faça sol.

Ao Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), campus Cajazeiras, professores e funcionários, por terem proporcionado um bom aprendizado e conhecimento durante o curso de Análise e Desenvolvimento de Sistemas (ADS).

Ao meu orientador, Prof. Fabio Abrantes Diniz, que sempre esteve aberto a ouvir todas as ideias e sugestões, ter ajudado na escolha do que melhor se encaixava neste Trabalho de Conclusão de Curso (TCC). E por sempre me incentivar e motivar a concluir esse Trabalho de Conclusão de Curso (TCC). Foi muito bom dividir e trocar essas experiências, que considero essenciais para a conclusão deste trabalho.

Aos meus colegas e amigos Antônia, Rogério, Lucas, Ericlys e todos os outros feitos durante essa jornada, foi muito bom passar esse tempo com vocês resenhando, compartilhando conhecimentos e sempre contando com ajuda nos momentos difíceis.

E por ultimo meus familiares e amigos, que sempre me apoiaram e incentivaram a não desistir, saibam que sou muito grato a todos.

## RESUMO

O setor de Mecanografia, responsável pelas impressões de documentos no IFPB - Campus Cajazeiras, é essencial para as rotinas administrativas e de ensino. Contudo, alguns processos operam de forma manual, resultando em falhas de comunicação, falta de rastreabilidade das solicitações e erros recorrentes. Tais gargalos geram retrabalho, desperdício de recursos e podem comprometer a entrega de materiais importantes. Diante deste cenário, o presente trabalho desenvolveu uma aplicação, intitulada MecanoSystem, com versões *Web/Mobile* a fim de automatizar o gerenciamento de impressões de documentos. Permitindo que os servidores possam realizar suas solicitações, anexar arquivos e monitorar o andamento de suas impressões, enquanto o operador gerencia a demanda de forma organizada e eficiente. Espera-se que a implantação do sistema no IFPB – campus Cajazeiras promova a diminuição de custos e tempo, aumente a agilidade dos processos e melhore a comunicação entre os envolvidos.

**Palavras-chave:** mecanografia, automatização, impressão, *web*, *mobile*, reprografia.

## **ABSTRACT**

The Mecanography Sector, responsible for printing documents at IFPB – Campus Cajazeiras, is essential for administrative and teaching routines. However, some processes are still carried out manually, resulting in communication failures, lack of request traceability, and recurring errors. These bottlenecks generate rework, resource waste, and may compromise the delivery of important materials. In this context, the present work developed an application, entitled MecanoSystem, with Web and Mobile versions, aimed at automating print management. The system enables staff members to submit requests, attach files, and monitor the progress of their print jobs, while operators manage the demand in an organized and efficient manner. It is expected that the implementation of the system at IFPB – Campus Cajazeiras will reduce costs and time, increase process agility, and improve communication among the parties involved.

**Keywords:** reprography, automation, printing, web, mobile, document management.

## LISTA DE FIGURAS

Figura 1 – Fluxograma do processo de impressão . . . . .	21
Figura 2 – Autorização para cópias . . . . .	22
Figura 3 – <i>Clean Architecture</i> . . . . .	27
Figura 4 – Painel administrativo do <i>PaperCut</i> . . . . .	30
Figura 5 – Configuração do recurso <i>Find-Me Printing</i> . . . . .	31
Figura 6 – Tela do recurso <i>Follow-You Printing</i> no <i>Equitrac</i> . . . . .	32
Figura 7 – Autenticação de usuário no terminal do <i>Equitrac</i> . . . . .	33
Figura 8 – Casos de uso do Administrador . . . . .	36
Figura 9 – Casos de uso do Mecanógrafo . . . . .	37
Figura 10 – Casos de uso dos Professores e Servidores . . . . .	38
Figura 11 – Arquitetura do sistema . . . . .	39
Figura 12 – Arquitetura backend MecanoSystem . . . . .	40
Figura 13 – Tela de Solicitar Impressão - <i>Web</i> . . . . .	44
Figura 14 – Tela de Solicitações Abertas - <i>Web</i> . . . . .	45
Figura 15 – Tela de Solicitar Impressão - <i>Mobile</i> . . . . .	46
Figura 16 – Tela de Solicitações Abertas - <i>Mobile</i> . . . . .	47
Figura 17 – <i>Prototype mobile</i> . . . . .	58
Figura 18 – Tela de Login - <i>Web</i> . . . . .	59
Figura 19 – Tela de Cadastrar Setores - <i>Web</i> . . . . .	60
Figura 20 – Tela de Gerenciamento de Setores - <i>Web</i> . . . . .	60
Figura 21 – Tela de Cadastrar Usuários - <i>Web</i> . . . . .	61
Figura 22 – Tela de Gerenciamento de Usuários - <i>Web</i> . . . . .	62
Figura 23 – Tela de Editar Solicitações - <i>Web</i> . . . . .	63
Figura 24 – Tela de Visualizar Solicitações - <i>Web</i> . . . . .	64
Figura 25 – Tela de Publicação de Avisos - <i>Web</i> . . . . .	65
Figura 26 – Tela de Lista de Avisos - <i>Web</i> . . . . .	66
Figura 27 – Tela para Ler um Aviso - <i>Web</i> . . . . .	66
Figura 28 – Tela de Mensagens - <i>Web</i> . . . . .	67
Figura 29 – Tela de Relatórios - <i>Web</i> . . . . .	68
Figura 30 – Tela de Login - <i>Mobile</i> . . . . .	69
Figura 31 – Tela de Editar Solicitação - <i>Mobile</i> . . . . .	70
Figura 32 – Tela de Visualizar Solicitação - <i>Mobile</i> . . . . .	71
Figura 33 – Tela de Lista de Avisos - <i>Mobile</i> . . . . .	72
Figura 34 – Tela de Ler Aviso - <i>Mobile</i> . . . . .	73
Figura 35 – Tela de Mensagens - <i>Mobile</i> . . . . .	74
Figura 36 – Tela de Relatórios - <i>Mobile</i> . . . . .	75

Figura 37 – Modelo Entidade Relacionamento . . . . .	77
Figura 38 – Modelo lógico . . . . .	78

## LISTA DE QUADROS

Quadro 1 – Requisitos funcionais do sistema . . . . .	35
Quadro 2 – Comparação das características dos trabalhos similares . . . . .	42

## LISTA DE ABREVIATURAS E SIGLAS

ADS	Análise e Desenvolvimento de Sistemas
IFPB	Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
TCC	Trabalho de Conclusão do Curso
REST	<i>Representational State Transfer</i>
MVC	<i>Model - View - Controller</i>
API	<i>Application Programming Interface</i>
SoC	<i>Separation of Concerns</i>
JSON	<i>JavaScript Object Notation</i>
HTTP	<i>Hypertext Transfer Protocol</i>
SQL	<i>Structured Query Language</i>
XML	<i>Extensible Markup Language</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IU	<i>User Interface</i>
DB	<i>Database</i>
IFCE	Instituto Federal de Educação, Ciência e Tecnologia do Ceará
SUAP	Sistema unificado de administração pública

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	JUSTIFICATIVA	15
1.2	OBJETIVO GERAL	16
1.3	OBJETIVOS ESPECÍFICOS	17
1.4	METODOLOGIA	17
<b>1.4.1</b>	<b>ORGANIZAÇÃO DO TRABALHO</b>	<b>18</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>19</b>
2.1	REPROGRAFIA	19
<b>2.1.1</b>	<b>PROCESSO DE IMPRESSÃO E GESTÃO: MECANOGRRAFIA IFPB-CAMPUS CAJAZEIRAS</b>	<b>20</b>
2.2	ARQUITETURA DE DESENVOLVIMENTO DE SOFTWARES	23
<b>2.2.1</b>	<b>ARQUITETURA REST</b>	<b>24</b>
<b>2.2.2</b>	<b>PADRÃO DE ARQUITETURA MVC</b>	<b>25</b>
<b>2.2.3</b>	<b>ARQUITETURA LIMPA</b>	<b>26</b>
2.3	TRABALHOS RELACIONADOS	28
<b>2.3.1</b>	<b><i>PaperCut</i></b>	<b>29</b>
<b>2.3.2</b>	<b><i>Equitrac Print Management</i></b>	<b>31</b>
<b>3</b>	<b>TRABALHO PROPOSTO</b>	<b>34</b>
3.1	LEVANTAMENTO DOS REQUISITOS	34
3.2	ANÁLISE DOS REQUISITOS	35
3.3	PROJETO DA INFRAESTRUTURA DO SISTEMA	39
3.4	ARQUITETURA DO BACKEND DO SISTEMA	40
3.5	Análise Comparativa com Trabalhos Relacionados	42
3.6	IMPLEMENTAÇÃO <i>Web</i>	43
3.7	IMPLEMENTAÇÃO <i>Mobile</i>	45
<b>4</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>48</b>

<b>REFERÊNCIAS</b> . . . . .	<b>49</b>
<b>APÊNDICE A – REQUISITOS DA APLICAÇÃO</b> . . . . .	<b>52</b>
<b>APÊNDICE B – PROTOTIPAGEM DAS TELAS</b> . . . . .	<b>58</b>
<b>APÊNDICE C – IMAGENS DAS TELAS DO SISTEMA <i>WEB</i></b> . . . . .	<b>59</b>
<b>APÊNDICE D – IMAGENS DAS TELAS DO SISTEMA <i>MOBILE</i></b> . . . . .	<b>69</b>
<b>APÊNDICE E – ARQUITETURA DO BANCO DE DADOS DO SISTEMA</b> . . .	<b>76</b>

# 1 INTRODUÇÃO

A modernização dos processos administrativos tem sido um fator crucial para o aumento da eficiência em diversas organizações. Conforme aponta Seas (2012), os avanços tecnológicos desafiam constantemente os métodos tradicionais de trabalho, especialmente em setores que ainda dependem de fluxos manuais. Este cenário se reflete em instituições de ensino, como é o caso do setor de Mecanografia do Instituto Federal da Paraíba (IFPB) - Campus Cajazeiras, cujo gerenciamento de impressões ainda se baseia em processos não informatizados.

A dependência de um sistema manual gera uma série de gargalos operacionais. A comunicação entre os servidores solicitantes e o operador da Mecanografia é dificultada, não há um método eficaz para rastrear o andamento de cada solicitação e a incidência de erros de impressão é elevada. Como consequência, acumula-se retrabalho, há desperdício de recursos e, fundamentalmente, compromete-se a disponibilidade de materiais para práticas de ensino e para os processos administrativos, gerando atrasos que impactam a rotina do campus.

Diante desta problemática, a automação surge como uma solução estratégica. Automatizar processos, segundo Moraes (2018), permite otimizar e racionalizar atividades, resultando na redução de tempo e custos, além de substituir tarefas repetitivas por aplicações de software. Adotando essa abordagem, este trabalho propõe o desenvolvimento de uma ferramenta tecnológica para sanar as deficiências identificadas.

Portanto, o presente trabalho desenvolveu uma aplicação, intitulada MecanoSystem, uma solução tecnológica composta por duas interfaces distintas e complementares para automatizar o gerenciamento de impressões da Mecanografia do IFPB - Campus Cajazeiras. A primeira é uma versão *mobile*, destinada aos servidores, que permitirá solicitar impressões e acompanhar o status de seus pedidos em tempo real. A segunda é uma versão *web* para o operador da Mecanografia, mas que também pode ser utilizada pelos servidores. Esta interface *web* centraliza o gerenciamento dos serviços, permitindo atender às solicitações, enviar notificações de status e extrair relatórios detalhados sobre a produção e o uso de recursos.

## 1.1 JUSTIFICATIVA

O setor de mecanografia do IFPB – Campus Cajazeiras ainda realiza o processo de solicitação e gerenciamento de impressões de forma manual, o que gera uma série

de problemas operacionais e administrativos. Em decorrência disso as consequências diretas estão o desperdício de papel e insumos, atrasos em entregas das impressões, acúmulo de filas no setor, e retrabalho devido a erros ou solicitações duplicadas. São os principais afetados por esse cenário os professores e técnicos administrativos, que enfrentam dificuldades para acompanhar o andamento de seus pedidos, e o mecanógrafo, que precisa gerenciar inúmeras solicitações sem apoio tecnológico. Para a instituição, esses problemas resultam em custos desnecessários, redução da eficiência administrativa e prejudicam a percepção de modernidade e profissionalismo do campus, impactando diretamente a qualidade dos serviços prestados.

Para resolver essas questões, é proposto nesse documento o desenvolvimento de uma aplicação chamada MacanoSystem, solução essa composta por uma versão Web e outra Mobile. A versão Web será utilizada por todos os usuários: mecanógrafo, discentes e servidores, permitindo o gerenciamento das solicitações e acompanhamento do status das impressões. Já a versão Mobile seria utilizada apenas por discentes e servidores, oferecendo uma maior comodidade e mobilidade no acompanhamento de suas solicitações. Além da aplicação apresentar um canal de mensagens interno e notificações, garantindo uma comunicação clara e imediata, reduzindo o retrabalho e evitando erros. A existência de duas versões Web e Mobile otimiza o fluxo de trabalho, fornecendo a ferramenta mais adequada para cada tipo de usuário, sem sobrecarregar o setor de mecanografia.

A aplicação por gerar relatórios, permite que a instituição monitore o consumo dos recursos, promovendo sua sustentabilidade e a racionalização. Também contribui para a redução de desperdício de matérias, pois possibilita o acompanhamento detalhado das solicitações, evitando impressões incorretas ou duplicadas.

Logo a relevância da aplicação vai além da resolução dos problemas operacionais. Ao modernizar o setor de mecanografia do IFPB – Campus Cajazeiras, a aplicação MecanoSystem reduz o uso de recursos públicos e melhora o ambiente de trabalho para servidores e discentes, promovendo eficiência no processo de impressão. Dessa forma, a aplicação contribui para a valorização institucional, reforçando a imagem de inovação e eficiência administrativa do campus e garantindo serviços de impressão mas confiáveis, rápidos e sustentáveis.

## **1.2 OBJETIVO GERAL**

Desenvolver um sistema de gestão multiplataforma para o setor de Mecanografia do IFPB - Campus Cajazeiras, que otimiza o fluxo de trabalho ao oferecer uma interface

*Web* para o operador gerenciar a fila de serviços e uma aplicação *Mobile* para facilitar a solicitação e o acompanhamento dos pedidos de impressão pelos servidores do campus.

### 1.3 OBJETIVOS ESPECÍFICOS

A fim de alcançar o objetivo geral proposto, os seguintes objetivos específicos foram definidos:

**Mapear** o fluxo de trabalho atual do setor de Mecanografia do IFPB - Campus Cajazeiras para identificar os principais gargalos e requisitos do sistema;

**Realizar** uma revisão bibliográfica sobre as tecnologias e arquiteturas de software adequadas para o desenvolvimento de sistemas *Web* e *Mobile* escaláveis e de fácil manutenção;

**Modelar** a arquitetura do sistema MecanoSystem, incluindo o banco de dados, e projetar as interfaces de usuário (UI/UX) para as plataformas *Web* (operador e servidores) e *Mobile* (servidores);

**Implementar** as funcionalidades da aplicação, desenvolvendo o back-end, a interface web administrativa e o aplicativo *mobile* para os servidores;

### 1.4 METODOLOGIA

Para a criação deste Trabalho de Conclusão de Curso II (TCCII), foi estabelecido um processo segmentado em diferentes etapas

- **Escolha do tema e planejamento inicial:** Etapa que define o planejamento e escolha inicial do tema.
- **Estudos com base na literatura nacionais e/ou internacionais os fundamentos para compreender a construção deste ensinamento:** Etapa de estudo com base na literatura para formular ideias.
- **Desenvolvimento do projeto:** Etapa de desenvolvimento do projeto dividida nas seguintes etapas:
  - **Organizado ideias:** Etapa onde foi organizado as ideias para o desenvolvimento da aplicação.

- **Levantamento de requisitos:** Etapa onde foram levantados os requisitos para o desenvolvimento da aplicação. Nesta etapa foi feito o estudo do processo manual utilizado na mecanografia para as solicitações de impressão, junto a reuniões com mecanógrafo.
- **Comparação com sistemas já existentes:** Etapa no qual foi estudada as plataformas existentes e o que melhorar nas mesmas.
- **Padrões para desenvolvimento de software:** Etapa onde foi estudada e definida as tecnologias para desenvolver a aplicação.
- **Implementação:** Etapa onde foi desenvolvido o sistema.
- **Correções e adicionando informações ao documento:** Etapa no qual foi corrigido alguns pontos e adicionado outros no documento final.

#### 1.4.1 ORGANIZAÇÃO DO TRABALHO

As demais seções deste documento está dividida em três capítulos. O capítulo 2 é responsável por abordar a fundamentação teórica, descrevendo as principais tecnologias e técnicas utilizadas. O Capítulo 3 discute a fase de análise e projeto do sistema proposto neste TCC. Finalmente, o Capítulo 4 conclui o documento, apontando as possibilidades futuras e as limitações do desenvolvimento da solução desenvolvida.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda os conceitos e tecnologias utilizados durante o desenvolvimento do trabalho. Inicialmente, o capítulo descreve sobre a reprografia de seus serviços e como é feita a gestão da mecanografia do IFPB-Cajazeiras no que diz respeito ao gerenciamento do processo de impressão. Em seguida, descreve arquiteturas de desenvolvimento software mais abordados. Por fim, detalha os trabalhos relacionados.

### 2.1 REPROGRAFIA

Segundo Foresti & Neto (2014), os serviços de reprografia apresentam como característica marcante a participação de empresas que atuam na área de tecnologia da informação, tendo em vista a necessidade de uma solução de tecnologia de informação para o gerenciamento e controle de impressão. O que torna, importante a escolha de serviços de maneira que proporcionem redução dos gastos e uma gestão sustentável dos recursos de impressão. No que diz respeito aos serviços de reprografia, os serviços da reprografia caracterizam-se pela prestação de serviços de impressão, reprodução, e digitalização, incluindo o fornecimento de materiais consumíveis, ficando a empresa contratada responsável pelo gerenciamento da solução instalada.

Quando não há um controle estruturado, o desperdício de recursos como papel, tinta entre outros, tende a aumentar gerando custos elevados para a instituição, segundo (SILVA; PEREIRA, 2018). Por esse motivo a adoção de seus serviços estão diretamente relacionados à eficiência na utilização de equipamentos e insumos. Sendo assim, soluções que podem informatizar a reprografia desempenham um papel importante ao possibilitar o monitoramento em tempo real, controle de solicitações e gerenciamento de relatórios.

A reprografia para instituições como o IFPB, possui um papel muito importante, uma vez que apoia as atividades pedagógicas e administrativas. Portanto o gerenciamento das impressões garante que os discentes e servidores tenham acesso aos materiais necessários em tempo hábil, sem comprometer a organização ou eficiência dos recursos disponíveis. Pois conforme Souza (2020) aponta, a informatização desse setor fortalece o acompanhamento das solicitações, reduz falhas de comunicação e assegura maior transparência na utilização dos serviços.

Logo, enquanto o modelo manual depende de registros físicos em papel e do

acompanhamento direto dos discentes e servidores, a informatização desse setor permite automatizar etapas, economizar recursos e reduzindo tempo de resposta, entre o mecanógrafo e os que solicitam as impressões. Com isso fica evidente a relevância dos serviços de reprografia modernos. E dessa forma sistemas informatizados ampliam a capacidade de controle e facilitam a integração com outras áreas da instituição (MOURA, 2021).

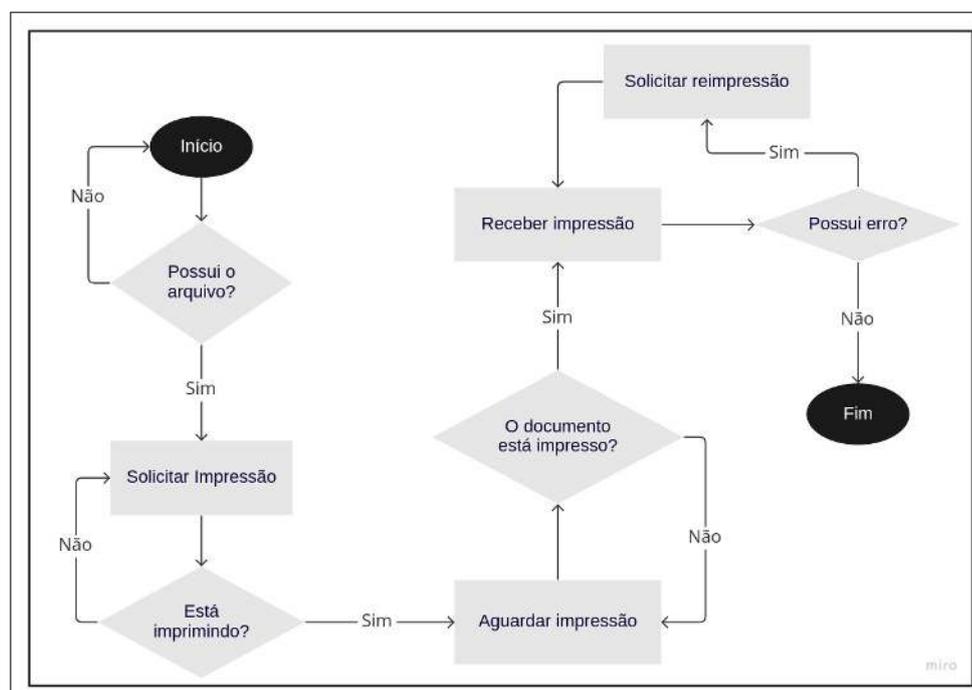
### **2.1.1 PROCESSO DE IMPRESSÃO E GESTÃO: MECANOGRRAFIA IFPB-CAMPUS CAJAZEIRAS**

O setor de Mecanografia do IFPB – Campus Cajazeiras é muito importante pois desempenha um papel fundamenta no apoio às atividades acadêmicas e administrativas, responsável pela impressão de matérias didáticos e administrativos solicitados por discentes e servidores. Sua função vai além da reprodução de cópias, pois centraliza e organiza os processos de impressão garantindo a padronização dos serviços prestados e o controle do materiais utilizados.

E para entender como ocorre o processo das impressões na mecanografia do IFPB-Campus Cajazeiras foi desenvolvido um fluxograma com as ações necessárias para a impressão.

A Figura 1 ilustra o fluxograma desenvolvido a partir de visitas ao setor da mecanografia e reuniões com o mecanógrafo, o fluxograma é importante para o presente trabalho pois ajuda a entender todo o processo necessário para a realização das impressões e quais passos devem ser seguidos.

Figura 1 – Fluxograma do processo de impressão



Fonte: Elaborado pelo autor

O fluxograma ilustrado na Figura 1 mostra todo o processo de impressão de documentos, que inicia com o servidor possuindo o arquivo do documento que deseja imprimir, e em posse desse arquivo o mesmo irá a mecanografia onde fará uma solicitação para a impressão, devendo preencher uma autorização para cópias como ilustrado na Figura 2.

Figura 2 – Autorização para cópias

 INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA COORDENAÇÃO DE BIBLIOTECA E RECURSOS MULTIMEIOS SETOR - MECANOGRÁFIA	
<b>AUTORIZAÇÃO PARA CÓPIAS</b>	
PREENCHIDO PELO SOLICITANTE	USO DA MECANOGRÁFIA
PROCEDÊNCIA: _____ SOLICITANTE: _____ TIPO DE SERVIÇO <input type="checkbox"/> Provas <input type="checkbox"/> Apostilas <input type="checkbox"/> Outros Obs: _____ Nº DE ORIGINALS: _____ Nº DE CÓPIAS P/ ORIGINAL: _____ TOTAL DE CÓPIAS: _____	MATERIAL A SER REPRODUZIDO FRENTE E VERSO <input type="checkbox"/> SIM <input type="checkbox"/> NÃO GRAMPEAR <input type="checkbox"/> SIM <input type="checkbox"/> NÃO RECEBIDO POR: _____ EM: ____/____/____
AUTORIZADO PELO(A) SR.(A): _____ EM: ____/____/____	<b>ATENÇÃO:</b> <i>Não protele! Solicite a reprodução do seu material com no mínimo 48 horas de antecedência.</i>
CUIDADO! REPRODUÇÃO DE LIVROS NA ÍNTEGRA OU PARCIALMENTE SEM AUTORIZAÇÃO DO AUTOR OU DA EDITORA, É CRIME COM PENA PREVISTA EM LEI – (9.610/98). PORTANTO NÃO INSISTA.	

Fonte: Elaborado pelo autor

A autorização de cópias ilustrada na Figura 2 possui duas colunas. A primeira Coluna (da esquerda) será preenchida pelo servidor solicitante com informações como procedência, a pessoa solicitante, o tipo de serviço com uma observação caso tenha, o número de originais, o número de cópias das originais, o total de cópias, a assinatura do solicitante e a data em que a autorização está sendo feita. Na segunda coluna (da direita) as seguintes informações a serem preenchidas pelo mecanógrafo: sim ou não para impressão frente e verso, sim ou não se deseja que o documento seja grampeado, o nome da pessoa que irá receber o documento quando estiver impresso e a data em que foi recebido. É importante ressaltar que a solicitação de impressão deve ser feita com até 48 horas de antecedência.

Com a autorização preenchida o próximo passo é a impressão que caso não esteja sendo realizada devido a problemas como falta de energia, internet, material para impressão ou qualquer outro problema que inviabilize a impressão, a solicitação será

guardada e a impressão será feita quando o problema for resolvido. Podendo ocorrer nesse tempo de espera o cancelamento da impressão por parte do servidor solicitante. Caso esteja realizando as impressões, a solicitação será atendida no mesmo instante e a documentação impressa será entregue, podendo haver um intervalo maior de tempo a depender do total de cópias a serem impressas e o total de solicitações recebidas pela mecanografia no momento em que a solicitação for feita. Após receber a impressão e identificar algum erro de impressão, seja no conteúdo ou na quantidade de cópias a reimpressão poderá ser feita, caso a impressão não possua nenhum o processo de impressão é concluído.

As solicitações concluídas serão utilizadas na geração de relatórios que serão usados para gerenciar o consumo dos materiais das impressões pela instituição como também registrar o total de impressões realizadas, que podem ser relatórios diários, mensais, bimestrais e anuais podendo ser organizados por coordenações ou por tipo de solicitante. Todo o processo de impressão até a geração de relatórios é feito de forma manual.

As análises do processo manual de impressões, foi fundamental para este trabalho, pois permitiram identificar os principais problemas enfrentados, como a necessidade de autorizações preenchidas em papel, a total falta de acompanhamento de qual parte do processo se encontra a impressão e atrasos por problemas técnicos. A partir da compreensão do fluxo, foi possível definir as requisitos para a aplicação proposta, informatizando etapas do processo de impressão que antes eram realizadas de forma manual, oferecendo notificações automáticas aos solicitantes, geração de relatórios e vários outros requisitos que são discutidos no decorrer deste documento. Dessa forma, o entendimento do processo atual utilizado pela instituição foi indispensável para que a aplicação desenvolvida de forma efetiva às demandas do setor de mecanografia do IFPB – Campus Cajazeiras.

## **2.2 ARQUITETURA DE DESENVOLVIMENTO DE SOFTWARES**

Citado pela primeira vez em 1970 no relatório técnico *Software Engineering Techniques* por Barbosa (2009), e posteriormente por vários outros autores diferentes que buscaram dar definições ao termo. Com o objetivo de promover uma aceitação na definição do termo entre autores, professores e estudiosos da área, a *Institute of Electrical and Electronics Engineers* (IEEE), no ano de 2000 por meio do Padrão 1471, propuseram o conceito de arquitetura de software como, a organização dos componentes de um sistema, dos relacionamentos entre os componentes e com

ambiente e os princípios utilizados para o design e evolução do software (IEEE, 2000).

A arquitetura de software segundo Lattanze (2009), é considerada um dos atributos de qualidade do sistema, já para Garland (2003), a arquitetura de sistemas é um conjunto de entidades, suas propriedades e seus relacionamentos, que definem a estrutura do sistema e a arquitetura de software como um conjunto de componentes de software, subsistemas, relacionamentos, interações, e as propriedades de cada um desses elementos. E o conjunto de princípios que juntos constituem as propriedades fundamentais e as restrições de um sistema de software ou de um conjunto de sistemas. Portanto, Tsuruta (2010), argumenta que, um software desenvolvido com uma arquitetura que não se preocupa com a manutenção e adaptação da aplicação apresentará muitos problemas nesses quesitos.

Pretendendo melhorar o desempenho das aplicações desenvolvidas e diminuir o risco de acontecerem problemas como os citados por Tsuruta (2010), vários padrões de arquitetura surgiram, no entanto os mais abordados para sistemas *web* e *mobile* são: *Representational State Transfer* (REST), o padrão de arquitetura *Model - View - Controller* (MVC), e o padrão de *Clean Architecture* (Arquitetura Limpa).

### 2.2.1 ARQUITETURA REST

Caracterizada como estilo arquitetural e proposta por Fielding (2021), nos anos 2000 em sua tese de doutorado o *Representational State Transfer* (REST), é ideal quando se trata de integração de sistemas, segundo Fielding (2021), *Representational State Transfer* (REST) é um híbrido de diversos conceitos relacionados à comunicação de sistemas distribuídos que são unidos para providenciar uma interface de programação única.

Alguns conceitos utilizados são estilo de comunicação Cliente-Servidor via troca de mensagens por *request-response* e a utilização do protocolo HTTP (*HyperText Transfer Protocol*) para a troca, formatação e interligação dos sistemas integrados, e por usar apenas o protocolo HTTP para troca de mensagens, o REST dá liberdade de escolher a linguagem de programação no qual a *Application Programming Interface* (API) definida como um conjunto de regras e protocolos que permite a comunicação entre diferentes softwares IBM (2021), vai ser implementada (MARCOS, 2013). Para uma API seja considerada REST é necessário ter como principal as seguintes características (TOBALDINI, 2021):

- **Protocolo cliente/servidor** - Todas as mensagens devem ser gerenciadas pelo protocolo HTTPS, levando sempre em consideração que as solicitações devem

conter todas as informações necessárias para a compreensão do pedido;

- **Operações bem definidas**- Todas as operações devem seguir um padrão, para que isso aconteça, o HTTP concebe um conjunto de operações POST, GET, PUT e DELETE, que devem ser combinadas com a persistência das informações do banco de dados;
- **Apresentações** - Normalmente a exposição das informações em uma arquitetura REST se faz por meio do *JavaScript Object Notation (JSON)* ou *Extensible Markup Language (XML)*. O JSON permite a representação de informações estruturadas e de forma legível por humanos e interpretável por máquinas, utilizando arrays, objetos, números, strings, valores nulos e booleanos, é um formato leve de troca de dados baseado em texto, sendo amplamente empregado em aplicativos (Amazon Web Services, 2025b). Já o XML é uma linguagem de marcação que representa dados estruturados usando tags, organizando informações em formato de árvore. Embora mais antigo que JSON, o XML ainda é usado em sistemas legados e em aplicações que exigem definição explícita de dados (Amazon Web Services, 2025a). Dessa forma, o recurso poderá ser utilizado com facilidade.

Entender a arquitetura REST foi essencial para o desenvolvendo desde trabalho, pois por possuir as características e vantagens descritas acima, foi possível a criação de uma API que garante a comunicação eficiente entre as duas versões da aplicação, Web e Mobile, possibilitando o acesso as mesmas funcionalidades de forma padronizada, segura e independente da linguagem ou plataforma utilizada.

A adoção do REST assegura portanto, a integridade entre os módulos das aplicações *web* e *mobile*, permitindo a eficiência no gerenciamento das solicitações pelo setor de mecanografia, o envio de notificações, a geração de relatórios, troca de mensagens e todas as outras funcionalidades, que não são mais importantes ou menos importantes que estas. Diante desses fatores, o seu emprego incorporado neste trabalho é indispensável.

### 2.2.2 PADRÃO DE ARQUITETURA MVC

O padrão *Model - View - Controller (MVC)* surgiu no começo da década de 80, mas se tornou muito popular na criação de aplicações *Web*, Luciano & Alves (2011), segundo oliveira (2013), é um padrão formado por três tipos de objetos, onde o primeiro consiste no *Model* (Modelo) que é o responsável por armazenar os dados e as regras de negócio do sistema, a segunda camada que é chamada de *View* (Visão) e tem a função de apresentar de forma gráfica os dados retornados pelo modelo.

Por último temos o *Controller* (Controlador), que é a camada responsável por fazer a mediação entre a visão e o modelo e controlar a entrada de dados. Sua dinâmica é simples, todas as requisições da aplicação são direcionadas para a camada Controller, que acessa a camada Model para processar a tal requisição, e por fim exibe o resultado da camada View. O padrão *Model - View - Controller* (MVC) separa as camadas de apresentação, de lógica de negócio e de gerenciamento do fluxo da aplicação, aumentando as capacidades de reutilização e de manutenção do projeto Luciano & Alves (2011), tornando um padrão de arquitetura ideal para o presente trabalho.

Assim como a arquitetura REST, entender o padrão de arquitetura MVC, foi de extrema importância para o desenvolvimento desta aplicação, pois permitiu a separar de forma clara a apresentação das informações, o controle das interações do usuário e a lógica de negócio. Além de possibilitar a reutilização de códigos em diferentes partes da aplicação, tornando-a mais flexível e de fácil manutenção. Para a aplicação desenvolvida neste trabalho a utilização dessa abordagem foi muito importante para manter a consistência entre a versão web, e mobile. Dessa forma, o uso do padrão MVC garantiu maior escalabilidade e clareza na implementação das funcionalidades, contribuindo para a eficiência do gerenciamento das solicitações de impressão no setor de mecanografia.

### 2.2.3 ARQUITETURA LIMPA

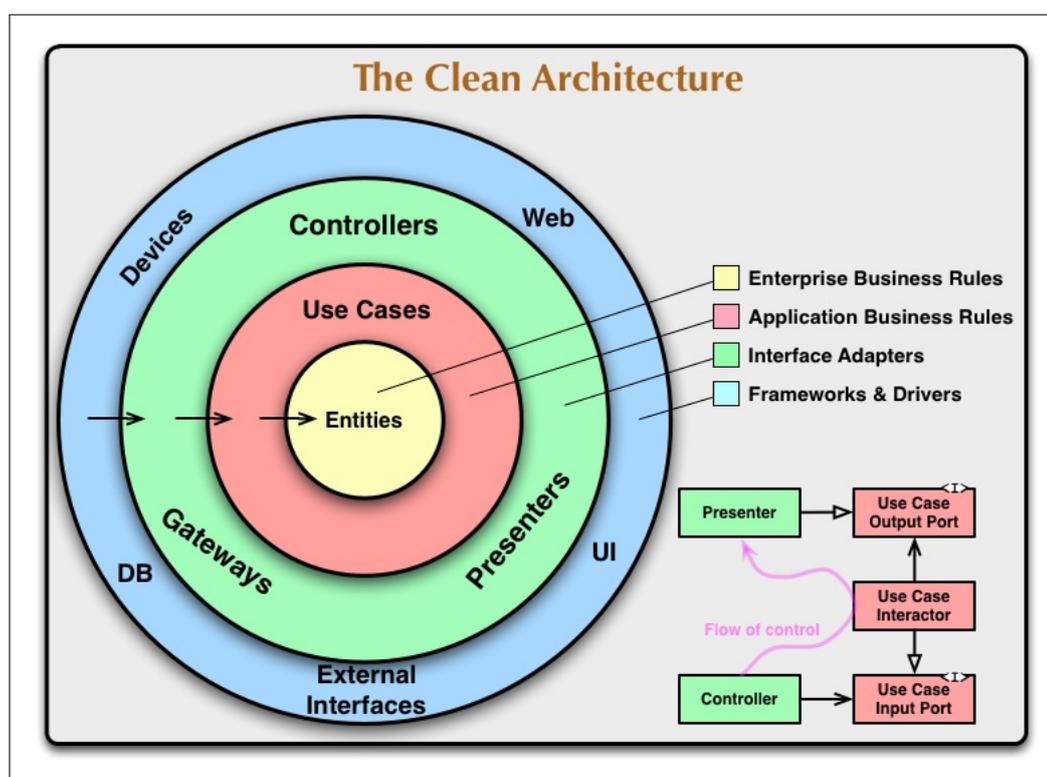
Arquitetura Limpa ou do inglês, *Clean Architecture*, foi uma arquitetura proposta por Martin (2017), e assim como as outras diferentes arquiteturas de software, tem como objetivo o conceito de *Separation of Concerns* (SoC), que traduzido do inglês fica Separação das Responsabilidades, e essa meta é atingida por meio da separação do software em camadas, pois, segundo Martin (2017), cada aplicação tem pelo menos uma camada de regras de negócio e outra para interface de usuário e sistema.

Segundo Martin (2017), a utilização desse padrão irá impactar em cinco características do sistema, tais como:

- **a independência de frameworks**, onde a arquitetura não depende da existência de nenhum framework, ou seja, só serão utilizados como ferramenta o que evitará restrições impostas pelos mesmos na construção do software;
- **a testabilidade**, que faz com que as regras de negócio da aplicação possam ser testadas sem o uso de nenhuma interface, base de dados ou qualquer outro tipo de elemento externo;

- **a independência de interface de usuário**, que significa que a aplicação funcionará independente da representação gráfica usada, pois as regras de negócio não serão alteradas;
- **a independência de banco de dados**, pois as regras de negócio não estarão relacionadas à ele, podendo trocar de uma base de dados para outra sem nenhum tipo de conflito;
- **e a independência de qualquer agente externo**, á última especificidade que resume todas as outras, pois sua aplicação não terá conhecimento nenhum sobre as interfaces com o mundo exterior (MARTIN, 2017).

Figura 3 – Clean Architecture



Fonte: Robert C. Martin (2017)

A Figura 3 ilustra a estrutura proposta pela arquitetura que é composta por quatro camadas: regras de negócio da empresa (*Entities*), regras de negócio da aplicação (*Use Cases*), adaptadores de interface (*Gateways*, *Presenters* e *Controller*) e a camada de *frameworks* e *drivers* (*Devices*, *Web*, *UI*, *DB*, *External Interfaces*). A camada mais interna é a de entidades, ela é responsável por encapsular as regras de negócio, sejam

da empresa quando a construção do software for gerenciado pela mesma, ou seja da aplicação em si quando não há uma empresa atuando no desenvolvimento (MARTIN, 2017).

Segundo Martin (2017), os casos de uso tem a função de manipular os dados das entidades e contém as regras de negócio específicas do aplicativo. Outra camada tratada na proposta de Martin (2017), é a de adaptadores de interface, essa camada é responsável por converter os dados da forma que as entidades e casos de uso utilizam, para a forma que um serviço externo como por exemplo uma interface na web ou banco de dados espera. É essa camada que engloba totalmente a arquitetura MVC, pois contém os controladores e visões, uma vez que o autor trata os modelos como as estruturas de dados utilizadas para trocar informações entre controladores e casos de uso. A camada mais externa é composta por bases de dados e interfaces web, e para Martin (2017), é importante manter essas ferramentas fora da estrutura da arquitetura limpa, pois, apresentam menos riscos de causar danos à aplicação.

Logo, tão importante quanto as arquiteturas explicadas nas subseções 2.2.1 e 2.2.2, entender a arquitetura limpa foi fundamental, pois permitiu organizar a aplicação em camadas independentes, o que resultou em uma flexibilidade para que tanto a versão web acessada por todos os usuários, quanto a versão mobile, voltada para discentes e servidores, pudessem evoluir. Além disso, proporcionou uma independência em relação a interfaces, frameworks e banco de dados, garantindo uma maior facilidade de manutenção, assim como a possibilidade de futuras integrações com novas ferramentas, tornando a solução mais robusta e duradoura para o setor de mecanografia.

## **2.3 TRABALHOS RELACIONADOS**

Realizou-se uma pesquisa com o objetivo de identificar sistemas que possam auxiliar o setor de mecanografia do IFPB-Campus Cajazeiras no gerenciamento de solicitações de impressões. Durante essa pesquisa, foram encontradas alguns sistemas. Em seguida, procedeu-se à seleção com base nos seguintes critérios: Os sistemas precisam ser multiplataforma, permitir o controle de cadastro de usuários, a solicitação de impressão, permitir publicações de avisos pelo mecanógrafo, permitir o acompanhamento de impressões, gerar relatórios, gerenciar as solicitações feitas. Com base nessas diretrizes, foram selecionados e analisados dois sistemas relacionados ao tema de gerenciamento de impressões as quais estão detalhadas nas seções seguintes.

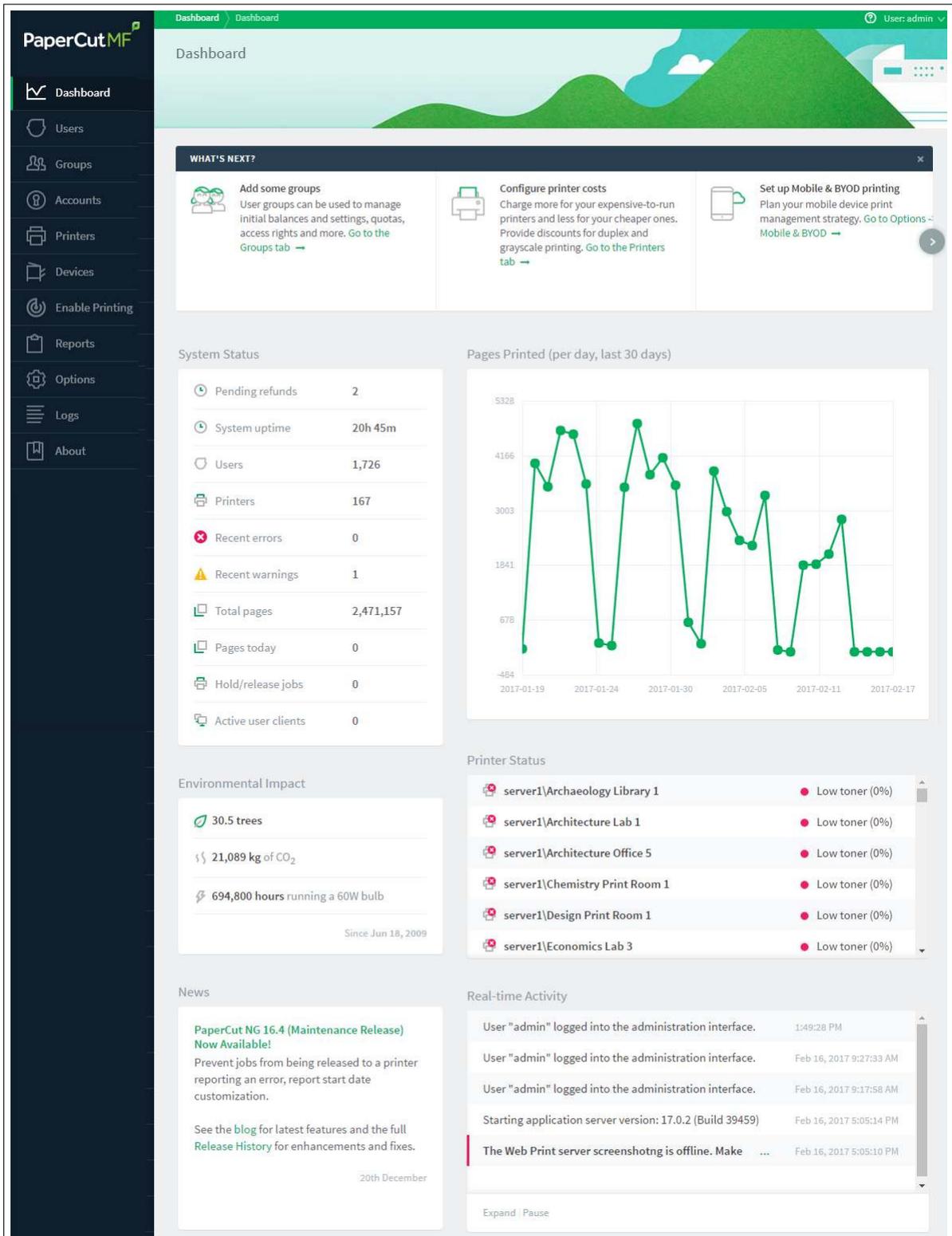
### 2.3.1 *PaperCut*

Desenvolvido pela empresa *PaperCut Software International*, o *PaperCut*<sup>1</sup> é uma solução de gerenciamento de impressões utilizada por empresas privadas, organizações governamentais e instituições de ensino, para controlar, monitorar e reduzir os custos de impressão. Dentre suas principais funcionalidades o software permite emitir relatórios, rastrear o uso por usuários e departamentos, definir cotas de impressão e automatizar processos manuais relacionados ao fluxo de documentos, diminuindo o desperdício de papel e suprimentos (SOFTWARE, 2024d).

Sua interface web intuitiva é uma das suas principais características, pois possibilita o acesso de seus usuários ao sistema diretamente pelo navegador, sem a necessidade de instalação no computador do cliente. O PaperCut também possui versões que oferecem aplicativos móveis, permitindo aos usuários a possibilidade de fazer solicitações e acompanhamento das mesmas por smartphone (SOFTWARE, 2024a). Por fim, o PaperCut é um sistema comercializado sob licença paga, com planos que podem ser adaptados de acordo com o porte da instituição.

<sup>1</sup> PaperCut: <https://www.papercut.com/pt-br/>

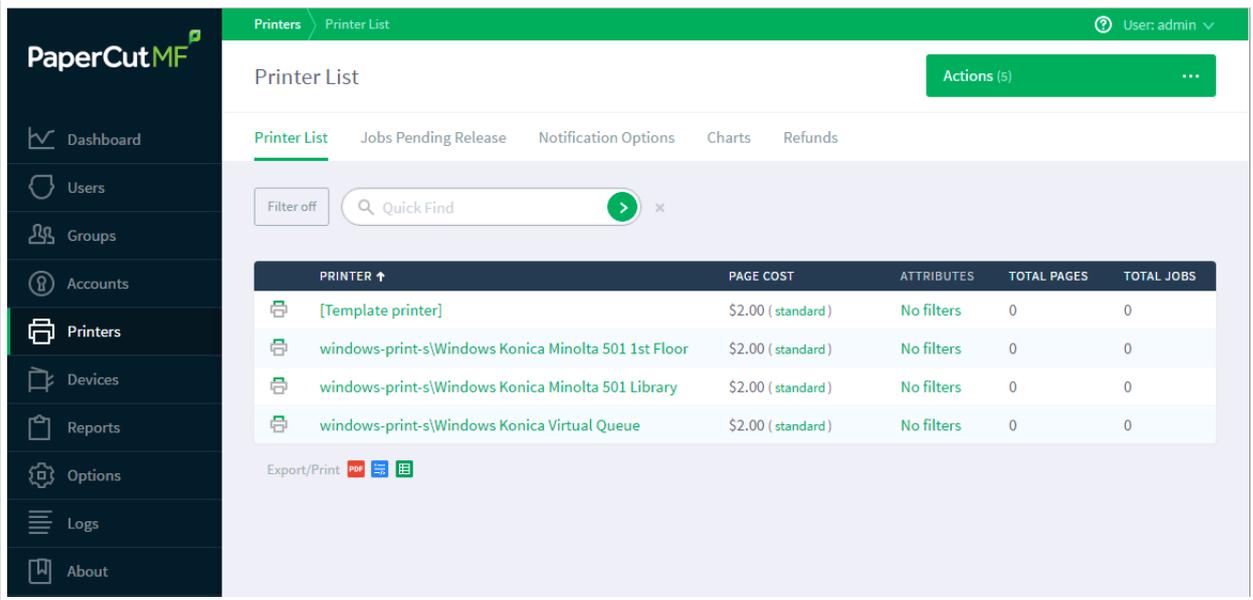
Figura 4 – Painel administrativo do PaperCut



Fonte: (SOFTWARE, 2024b)

A Figura 4 mostra o painel administrativo do PaperCut, é nesse painel onde o administrador tem acesso às informações dos usuários que estão usando o sistema, o volume de impressões e a utilização de impressoras, tornando-se esse recurso essencial para o acompanhamento em tempo real do controle de custos e o consumo de recursos.

**Figura 5 – Configuração do recurso *Find-Me Printing***



PRINTER ↑	PAGE COST	ATTRIBUTES	TOTAL PAGES	TOTAL JOBS
[Template printer]	\$2.00 (standard)	No filters	0	0
windows-print-s\Windows Konica Minolta 501 1st Floor	\$2.00 (standard)	No filters	0	0
windows-print-s\Windows Konica Minolta 501 Library	\$2.00 (standard)	No filters	0	0
windows-print-s\Windows Konica Virtual Queue	\$2.00 (standard)	No filters	0	0

Fonte: (SOFTWARE, 2024c)

Observa-se na Figura 5 uma das funcionalidades chamada Find-Me Printing, essa funcionalidade permite que os usuários possam enviar um documento para uma fila centralizada de impressões e depois liberar o documento para qualquer impressora autorizada. Segundo a Software (2024c) essa função reduz o desperdício de papel causado por impressões esquecidas e aumenta a flexibilidade no ambiente corporativo e acadêmico.

### 2.3.2 *Equitrac Print Management*

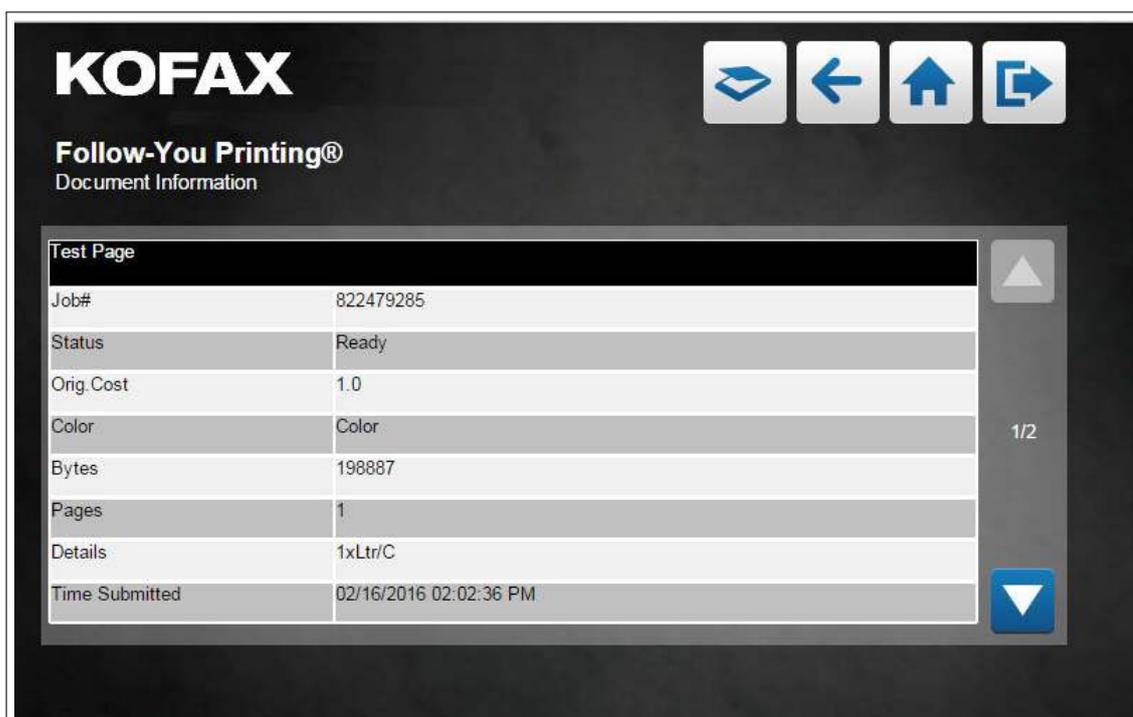
Desenvolvida pela empresa *Tungsten Automation* (antiga Kofax), o *Equitrac Print Management*<sup>2</sup>, é um sistema de gerenciamento de impressões, e tem como foco a segurança, rastreabilidade das impressões e o controle de custos, e assim como

<sup>2</sup> Equitrac: <https://software.com.br/produto/equitrac-print-management/>

o sistema *PaperCut* é utilizado em ambientes corporativos e educacionais. Dentre suas principais funcionalidades, o *software* possibilita a autenticação do usuário antes da liberação para impressão de documentos, gerenciamento de cotas, geração de relatórios e a integração com cartões de identificação. Segundo Automation (2024b), diferente de outras soluções menores, o *Equitrac Print Management* se destaca por oferecer forte integração com grandes parques de impressão.

Assim como o *PaperCut*, o *Equitrac* também possui um sistema web, que pode ser acessado por qualquer navegador e é voltada para administradores, o seu sistema possui módulos que podem ser integrados a estações de trabalho ou a terminais de autenticação que são instalados junto às impressoras e onde a interação principal ocorre com os usuários finais. O *Equitrac* não possui um foco em aplicações móveis próprios, podendo ser utilizado em conjunto com plataformas de gerenciamento de dispositivos móveis e soluções de nuvem. Por fim, trata-se de um *software* comercial como o *PaperCut*, com licenciamento pago e voltado ao mercado corporativo.

Figura 6 – Tela do recurso *Follow-You Printing* no *Equitrac*

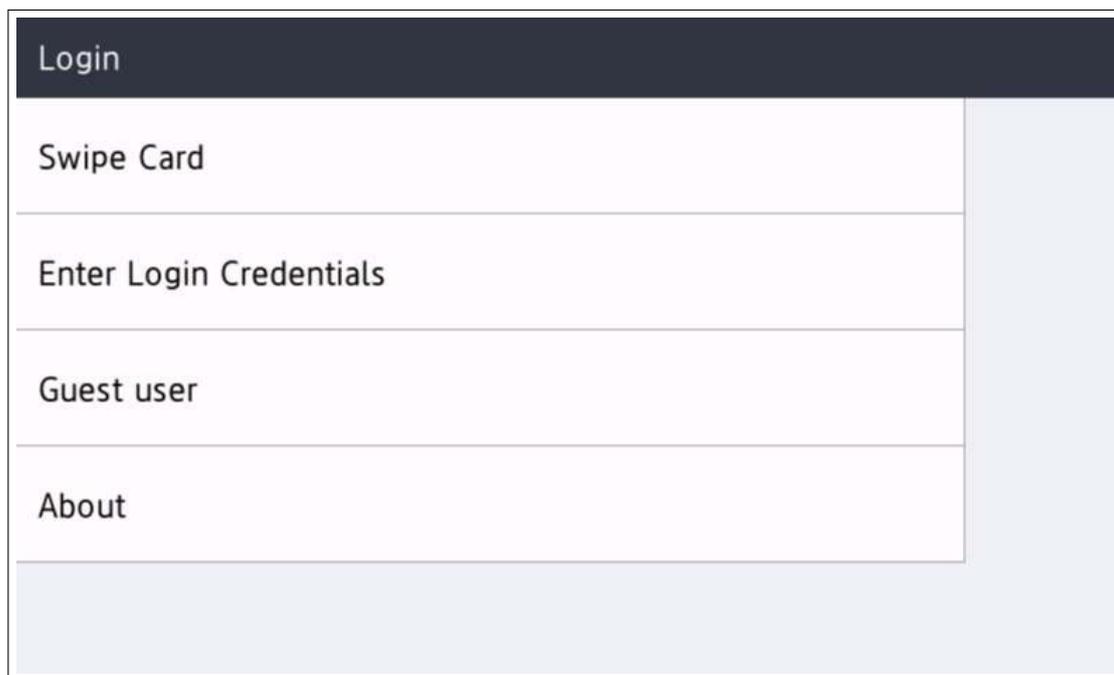


Fonte: (AUTOMATION, 2024a)

Na Figura 6 mostra o recurso *Follow-You Printing*, uma funcionalidade seme-

lhante ao *Find-Me* do *PaperCut*, na qual o usuário envia o documento que deseja imprimir para uma fila central, e onde o documento só é liberado para imprimir na impressora escolhida, após autenticação. Aumentando assim a segurança dos documentos e reduzindo impressões abandonadas.

**Figura 7 – Autenticação de usuário no terminal do *Equitrac***



Fonte: (AUTOMATION, 2024c)

A Figura 7 ilustra o processo de autenticação de usuário em terminal vinculado ao *Equitrac*, exigindo login ou cartão de identificação, esse recurso garante que apenas os usuários que possuem autorização acessem as impressoras, reforçando assim a segurança e a conformidade em instituições e empresas.

### 3 TRABALHO PROPOSTO

O trabalho proposto, intitulado como MecanoSystem, é uma aplicação desenvolvida com o propósito de auxiliar o mecanógrafo no gerenciamento da produção de material para a instituição, e auxiliar também os docentes e servidores que utilizam o serviço, em que a mesma pode ser acessada por qualquer aparelho que possua um navegador *web* instalado ou por uma aplicação *mobile*.

Necessitando de acesso à Internet para seu uso, tanto o sistema *web*, quanto o *mobile* dispõe de funcionalidades como visualização de solicitações feitas, envio de avisos e/ou mensagens, geração de relatórios, para o mecanógrafo e envio, impressão e acompanhamento das solicitações, para os docentes e servidores. Além de várias outras funções pensadas com o objetivo de garantir eficiência no gerenciamento das impressões, e um melhor acompanhamento sobre o estado das impressões solicitadas pelos docentes e servidores, além de ganho de tempo e recursos.

Logo abaixo será descrito o método de coleta de requisitos além de a exposição dos mesmos e, logo, após, será apresentada a arquitetura utilizada para o desenvolvimento da plataforma e as funcionalidades que já estão disponibilizadas.

#### 3.1 LEVANTAMENTO DOS REQUISITOS

Com o intuito de auxiliar o mecanógrafo no gerenciamento das impressões, foi proposto o sistema *web* e *mobile* de gerenciamento de solicitações de impressão MecanoSystem. O mesmo promete disponibilizar uma interface simples e intuitiva para o mecanógrafo e para os docentes e servidores, tornando o gerenciamento das impressões mais eficiente.

Para a coleta dos requisitos do Sistema foi utilizado como base, o modelo de autorização para cópias supracitada na seção 2.1.2, e que é utilizado pela mecanografia, para o controle e gerenciamento das solicitações de impressões. Além disso reuniões foram feitas com o mecanógrafo, um professor, e um servidor e novas funcionalidades que argumentaram ser importantes foram identificadas. Além disso, foram realizadas também pesquisas bibliográficas, e estudos de trabalhos relacionados.

Segundo Sommerville (2013), os requisitos definem as funções que um sistema deve oferecer. Com base nessa definição, os requisitos funcionais identificados para a sistema proposto estão listados no Quadro 1, que detalha os requisitos funcionais, e

que tem como objetivo auxiliar os servidores e o setor de mecanografia gerenciamento das solicitações de impressão do campus Cajazeiras do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB). Portanto, foram feitos os casos de usos explicados na próxima seção e os user stores que se encontram no Apêndice A.

**Quadro 1 – Requisitos funcionais do sistema**

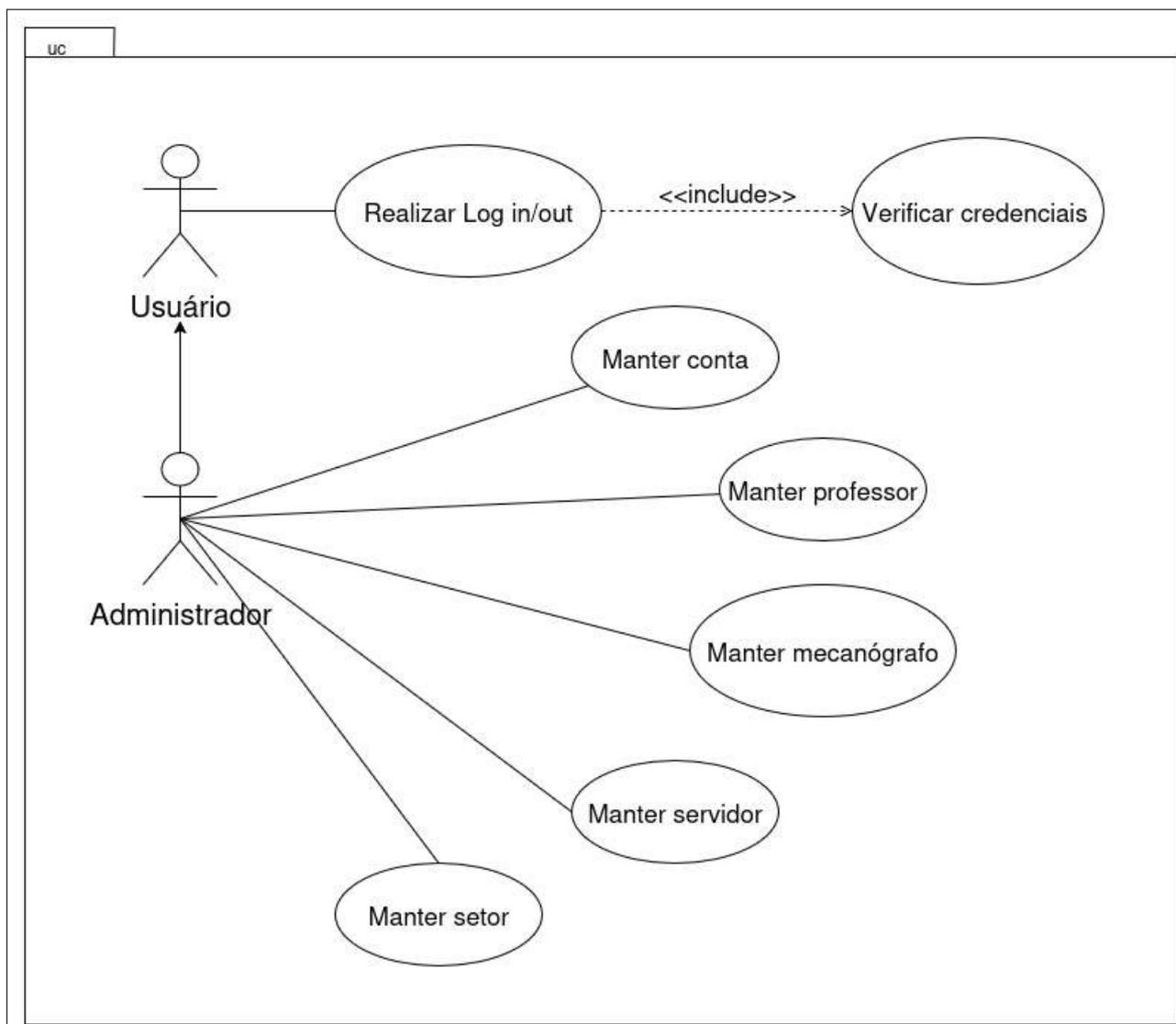
<b>Requisitos Funcionais</b>	<b>Descrição</b>
<b>RF01</b>	A aplicação deve manter as contas dos usuários
<b>RF02</b>	A aplicação deve manter as solicitações de impressão
<b>RF03</b>	A aplicação deve permitir a geração de relatórios pelo mecanógrafo, professor e servidor
<b>RF04</b>	A aplicação deve manter as publicações de avisos feitas pelo mecanógrafo
<b>RF05</b>	A aplicação deve manter as mensagens enviadas entre os usuários
<b>RF06</b>	A aplicação deve manter os setores
<b>RF07</b>	A aplicação deve realizar a autenticação dos usuários

Fonte: Elaborado pelo autor

## **3.2 ANÁLISE DOS REQUISITOS**

O sistema possui quatro atores: o administrador, mecanógrafo, professor e o servidor, o administrador é responsável por manter o cadastro dos docentes, servidores e do mecanógrafo, além de manter o cadastro dos setores. Como demonstrado na Figura 8.

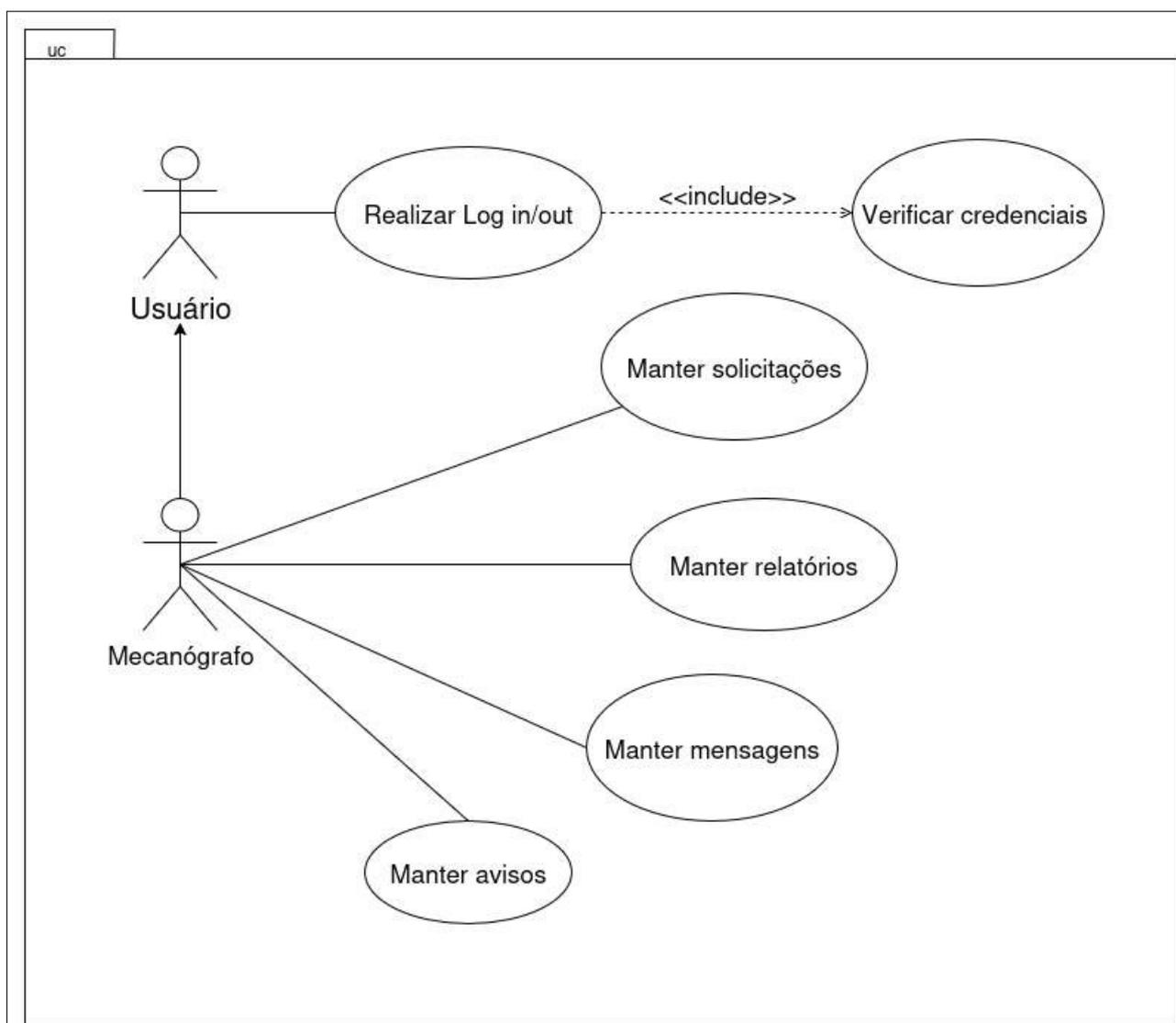
Figura 8 – Casos de uso do Administrador



Fonte: Elaborado pelo autor

Outro ator é o mecanógrafo que desempenha o papel de administrar as solicitações criadas pelos docentes e servidores, publicar avisos com notificações para todos os usuários cadastrados no sistema, o envio de mensagens diretas para professores e servidores por meio de chat, dentre outras funcionalidades demonstradas no diagrama de caso de uso da Figura 9.

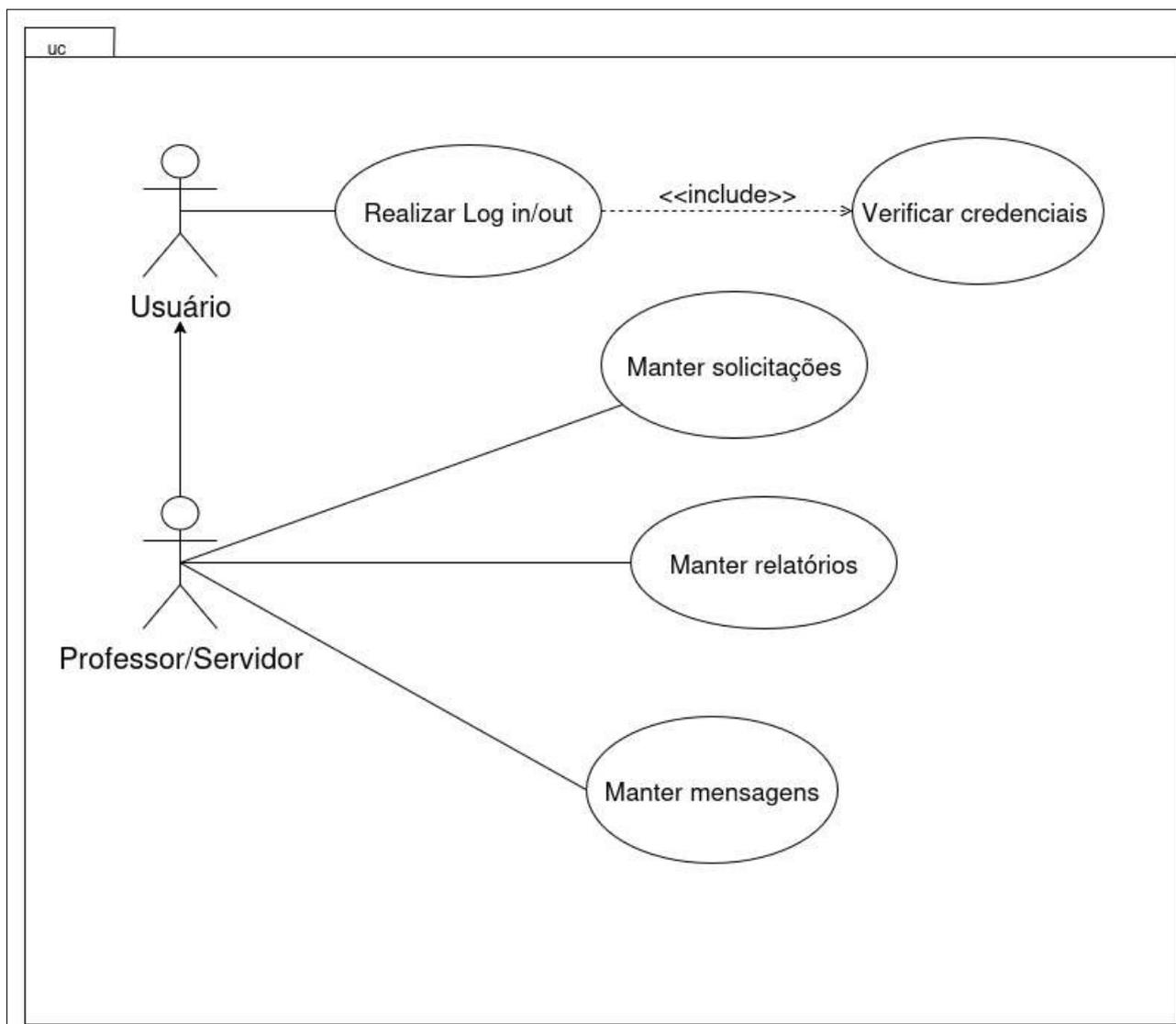
Figura 9 – Casos de uso do Mecnógrafo



Fonte: Elaborado pelo autor

Por fim, os dois últimos atores, o professor e o servidor que utilizam o sistema para criar as solicitações de impressões, acompanhar o estado das solicitações criadas, enviar mensagens diretas para o mecnógrafo através do chat, dentre outras funcionalidades demonstradas no diagrama de caso de uso da Figura 10.

Figura 10 – Casos de uso dos Professores e Servidores



Fonte: Elaborado pelo autor

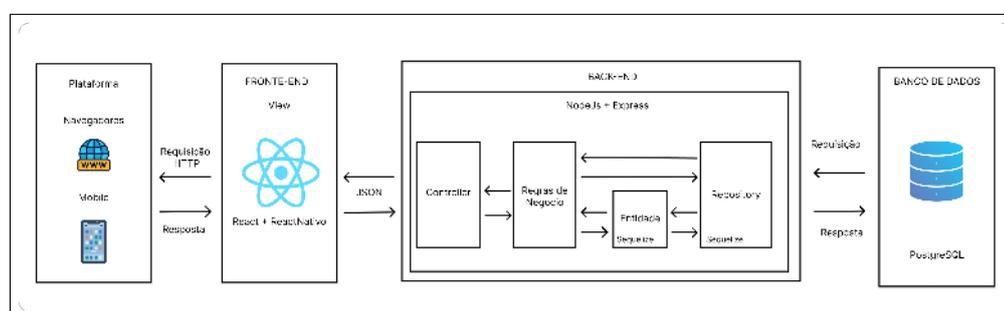
As Figuras 8, 9 e 10 destacam os casos de uso levantados para todos os módulos da aplicação que são apresentados como requisitos funcionais no Quadro 1 acima e estão descritos detalhadamente no Apêndice A.

No Apêndice B encontra-se as páginas da aplicação *Mobile*, e no Apêndice C as telas da aplicação *Web*, que foram feitas a partir da análise de requisitos e demonstra de forma geral as páginas, que podem ser acessadas pelos usuários em seus respectivos módulos e que serão utilizadas como modelo para o desenvolvimentos da aplicação.

### 3.3 PROJETO DA INFRAESTRUTURA DO SISTEMA

O sistema MecanoSystem possui uma arquitetura dividida em três camadas, seguindo o padrão *Model-View-Controller* (MVC), juntamente com a *Clean Architecture* (Arquitetura Limpa). Segundo Martin (2017), o uso dos dois padrões em conjunto possibilita a escalabilidade da aplicação sem muitas preocupações, pois a *Clean Architecture* (Arquitetura Limpa) modulariza as responsabilidades da aplicação de uma maneira que alterações em serviços externos, tais como interfaces e bancos de dados, não impliquem em quebras ou em muitas mudanças no software.

Figura 11 – Arquitetura do sistema



Fonte: Elaborado pelo autor

Ao analisarmos a Figura 11 pode-se perceber os componentes que fazem parte da estrutura da arquitetura utilizada no desenvolvimento da aplicação, onde a primeira parte se refere às plataformas utilizadas para acessar o sistema, que se divide em duas, Browser e mobile. A interface do usuário se comunica com o servidor da aplicação através de uma das plataformas, que é dividido em *Front-end* e *Back-end*, por requisições HTTP (*HyperText Transfer Protocol*) que acessam a camada de visão, a qual está sendo desenvolvida utilizando o framework JavaScript, ReactJS<sup>1</sup> para versão web e React Native<sup>2</sup> para mobile, e se encontra no *Front-end*. Para a troca de informações na aplicação o *Front-end* envia e recebe dados no formato *JavaScript Object Notation* (JSON<sup>3</sup>) do *Back-end*, que usa o NodeJS<sup>4</sup>, juntamente com a biblioteca Express<sup>5</sup>, fornecendo assim uma *Application Programming Interface* (API).

<sup>1</sup> <https://pt-br.reactjs.org/>

<sup>2</sup> <https://reactnative.dev/>

<sup>3</sup> <https://www.json.org/json-en.html>

<sup>4</sup> <https://nodejs.org/en/>

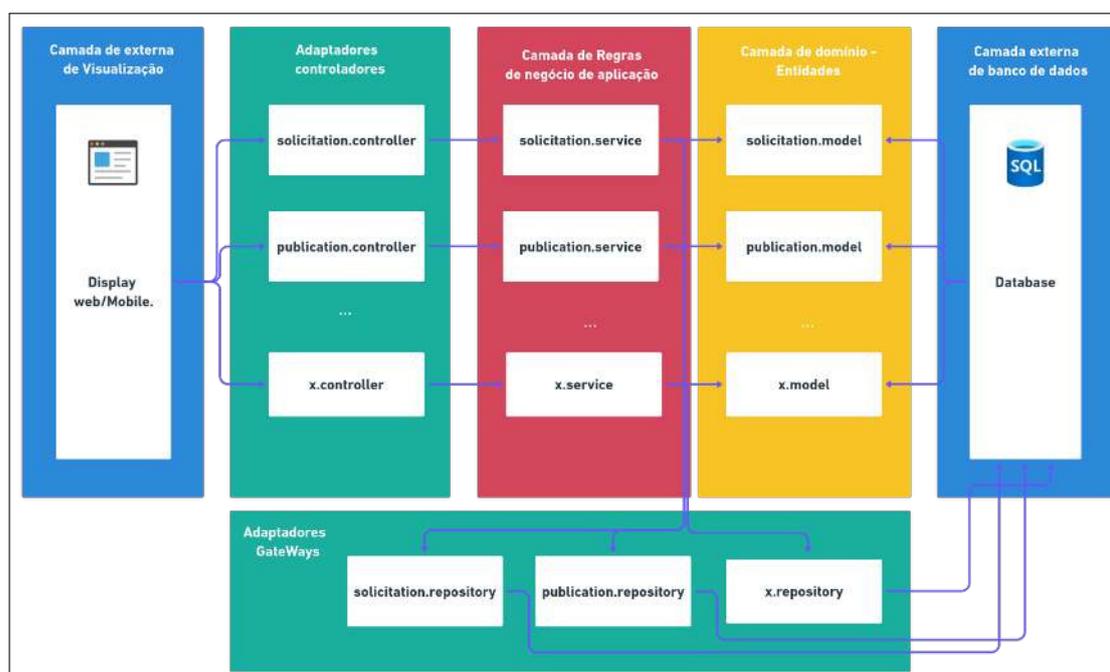
<sup>5</sup> <https://expressjs.com/pt-br/>

Organizada segundo a *Clean Architecture*, a API do sistema é composta por quatro componentes, onde o primeiro são os controladores, que se comunicam com as regras de negócios da aplicação. Os repositórios são responsáveis pela permanência dos dados na base de dados e se comunicam com as entidades para realizar suas funções. A última camada da API é composta pelas entidades, que são uma representação das tabelas do banco de dados e, também, trocam informações com as regras de negócio. Os repositórios e entidades foram implementados utilizando o Sequelize<sup>6</sup>, que auxilia na conversão dos dados presentes na aplicação para o formato utilizado pela base de dados. Para manter os dados da aplicação será utilizado o PostgreSQL<sup>7</sup>, um banco de dados relacional utilizado por muitas aplicações em seu desenvolvimento. No Apêndice E encontra-se o projeto de modelagem do banco de dados.

### 3.4 ARQUITETURA DO BACKEND DO SISTEMA

A ilustração da Figura 12, apresenta a representação visual da organização e atuação do backend da aplicação MecanoSystem com base no padrão de projeto *Clean Architecture*. Essa representação possibilita a separação clara das responsabilidades do sistema, assim possibilitando a identificação do fluxo de dados por trás do MecanoSystem, além de facilitar a manutenção e expansão do sistema.

Figura 12 – Arquitetura backend MecanoSystem



Fonte: Elaborado pelo autor

<sup>6</sup> <https://sequelize.org/>

<sup>7</sup> <https://www.postgresql.org/>

Ao examinar a Figura 12, torna-se evidente o trajeto dos dados dentro da camada de backend da aplicação MecanoSystem, proporcionando uma compreensão clara do fluxo no contexto do projeto baseado na arquitetura *Clean Architecture*. Essa implementação da *Clean Architecture* possibilitou a organização dos serviços da aplicação em camadas, viabilizando a separação de responsabilidades, como a distinção entre lógica de negócios, apresentação e controle de fluxo. O fluxo de dados apresenta-se da seguinte forma:

- 1. O usuário realiza uma solicitação de impressão por meio de uma requisição HTTP através de alguma aplicação web ou mobile que tenha integrado a API (Camada externa de Visualização);
- 2. A API do MecanoSystem recebe a solicitação do controlador ao qual foi encaminhada. Os controladores são encarregados de disponibilizar os endpoints para serviços específicos, como salvar uma nova solicitação de impressão ou publicação, consultar alguma solicitação de impressão ou publicação cadastrada e vários outros serviços, além de aplicar validações de informações utilizando DTO's<sup>8</sup>. Os controladores encaminham os dados recebidos na solicitação para a camada de serviços correspondente ao serviço que o usuário deseja utilizar;
- 3. A camada de serviços (Regras de Negócio) é responsável por aplicar as regras necessárias nas informações provenientes dos controladores. Essa camada pode utilizar entidades (Camada de Domínios) para construir DTO's. Após a aplicação das regras de negócio, os dados são enviados para o repositório correspondente associado ao método HTTP da solicitação;
- 4. O repositório é responsável pelas operações de CRUD conforme informado pela camada de serviços, estabelecendo uma conexão direta com o banco de dados para executar essas operações;
- 5. Subsequentemente, é gerada uma resposta que percorre todas as camadas até alcançar o controlador, que envia uma resposta no formato JSON para a aplicação que o usuário está utilizando;

---

<sup>8</sup> Objeto de Transferência de Dados, é um padrão de projeto de software usado para transferir dados entre subsistemas de um software

### 3.5 ANÁLISE COMPARATIVA COM TRABALHOS RELACIONADOS

A fim de destacar as especificidades de cada aplicação uma análise comparando as características dos três sistemas discutido nas seções anteriores foi realizada, e com o objetivo de organizar melhor as informações no Quadro 2, atribuiu-se as seguintes nomenclaturas às aplicações similares:

- **APP 01** - PaperCut
- **APP 02** - Equitrac Print Management

Quadro 2 – Comparação das características dos trabalhos similares

<b>Características</b>	<b>APP 01</b>	<b>APP 02</b>	<b>MecanoSystem</b>
<b>Controle de cadastro de usuários</b>	SIM	SIM	<b>SIM</b>
<b>Solicitar impressão</b>	SIM	SIM	<b>SIM</b>
<b>Chat</b>	NÃO	NÃO	<b>SIM</b>
<b>Publicação de notificação pelo mecanógrafo</b>	NÃO	NÃO	<b>SIM</b>
<b>Acompanhamento de impressão</b>	SIM	SIM	<b>SIM</b>
<b>Geração de relatórios</b>	SIM	SIM	<b>SIM</b>
<b>Controle de impressão</b>	SIM	SIM	<b>SIM</b>
<b>Aplicação mobile/web</b>	SIM	NÃO	<b>SIM</b>
<b>Aplicação gratuita</b>	NÃO	NÃO	<b>SIM</b>

Fonte: Elaborado pelo autor

Após analisar o Quadro 2 é possível visualizar que o MecanoSystem não apenas une, mas integra as funcionalidades que já são aplicadas pelo sistema comparado, além de utilizar alguns recursos que trazem uma experiência melhor e instigam um interesse maior no uso da aplicação, como, por exemplo, a possibilidade do servidor poder acompanhar o estado de sua impressão, a troca de mensagens entre o mecanógrafo e o servidor, o uso da aplicação em navegadores *web* e dispositivos *mobile*, e a publicação de notificação pelo mecanógrafo.

## 3.6 IMPLEMENTAÇÃO WEB

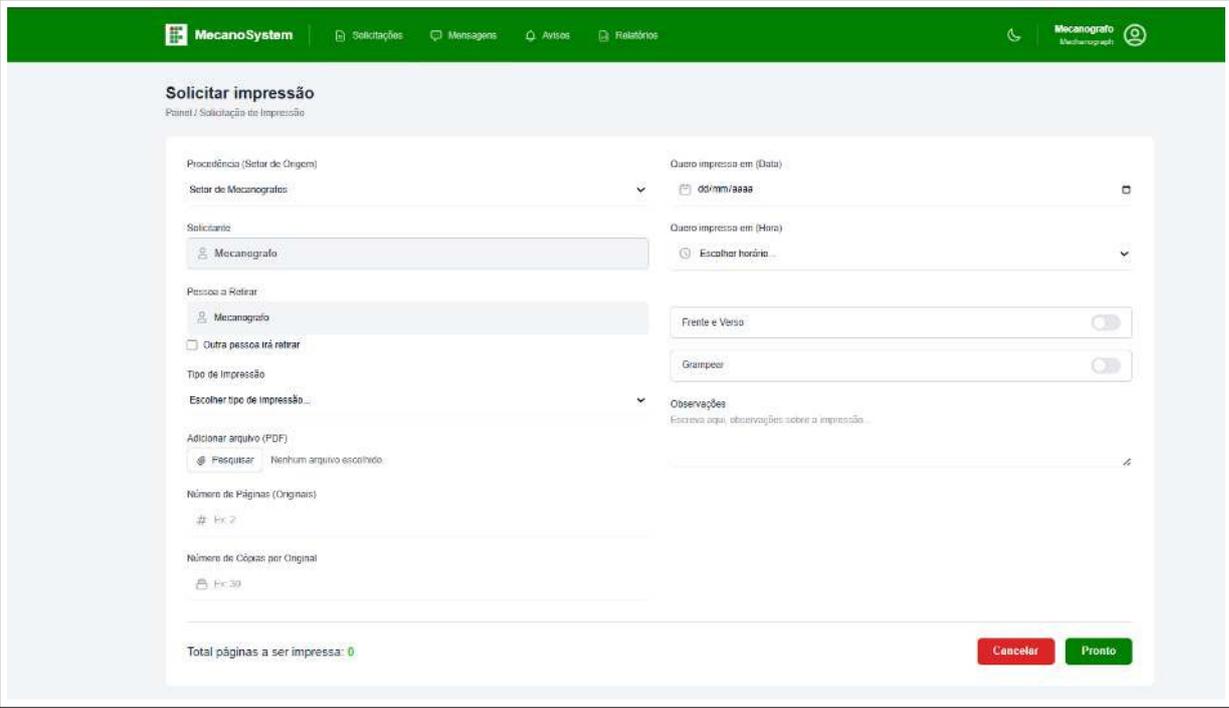
Serão apresentadas, a seguir, algumas telas principais referentes a parte de criação e listagem de uma solicitação feita por um usuário, as demais páginas estarão disponíveis no Apêndice C.

A Figura 13, apresenta a tela da aplicação *Web* na qual o usuário criar uma solicitação de impressão. Essa página contém os seguintes campos para inserção de dados no formulário: procedência, solicitante, pessoa a retirar, tipo de impressão, adicionar arquivos, número de páginas (originais), números de cópias, quero impresso em (data), quero impresso em (hora), frente e verso, grampear, e observações.

No campo correspondente a Procedência ou setor de origem, o usuário deve escolher qual setor deseja que a solicitação seja registrada. O campo é um seletor de setores e contém os setores ao qual o usuário que fez o login pertence. O campo solicitante será preenchido automaticamente de acordo com o usuário logado. No campo pessoa a retirar o usuário deve escrever o nome da pessoa que irá retirar a impressão junto a mecanografia, ele já vem preenchido automaticamente com o nome do solicitante podendo ser alterado clicando na caixa outra pessoa irá retirar. O próximo campo é o tipo de impressão contém um seletor com as opções (prova, apostila e outros). No campo adicionar arquivo o usuário deve escolher o arquivo que deseja imprimir clicando no botão de Pesquisar. O campo número de páginas (originais) será preenchido automaticamente de acordo com o número de páginas do arquivo escolhido pelo usuário. No campo de número de cópias por original o usuário deve digitar a quantidade de cópia que deseja do arquivo original.

O formulário também conta com os campos de quero impresso em (data) e quero impresso em (hora), onde no primeiro o usuário deve digitar a data que deseja que a impressão esteja pronta ou selecionar no botão de calendário, já no segundo deve selecionar a hora. Os próximos campos são os de frente e verso e grampear, onde o usuário deve marcar como deseja que sua impressão seja tratada. Um campo de observações onde o usuário pode escrever alguma informação ou observação sobre o documento a ser impresso. A tela também possui a informação de quantas páginas serão impressas. Por fim, a tela possui dois botões, um de cancelar caso o usuário não queria mas solicitar a impressão e outro com nome pronto que envia os dados da solicitação.

**Figura 13 – Tela de Solicitar Impressão - Web**

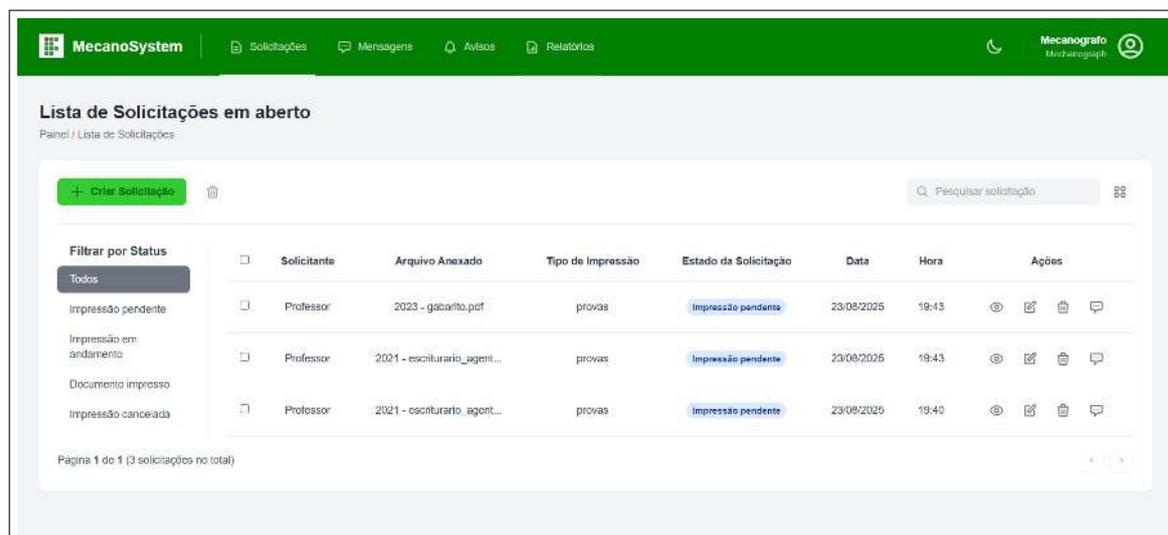


Fonte: Elaborado pelo autor

Na Figura 14, é ilustrada a tela de Solicitações Abertas onde são listadas as solicitações criadas pelos usuários. Na página encontramos um botão criar solicitação onde o usuário, clicando nele, criar a solicitação de impressão e quando feita será listada na tabela abaixo junto as outras caso exista. Logo abaixo do botão de criar solicitações, temos também os filtros por status em que as solicitações se encontram.

Na tabela onde estão listadas as solicitações são mostradas as informações principais dela como: nome do solicitante, nome do arquivo anexado, o tipo de impressão, o estado em que a solicitação se encontra e que é atualizado pelo mecanografo para informar em qual etapa está a solicitação, a data e a hora em que a solicitação foi criada. Por ultimo, as ações que os usuários podem executar nas solicitações listadas, sendo elas: visualizar as informações da solicitação, editar, excluir e enviar mensagem.

Figura 14 – Tela de Solicitações Abertas - Web



Fonte: Elaborado pelo autor

### 3.7 IMPLEMENTAÇÃO MOBILE

Serão apresentadas, a seguir, algumas telas principais referentes a parte de criação e listagem de uma solicitação feita por um usuário na versão *Mobile*, as demais telas estarão disponíveis no Apêndice D. Por seu funcionamento ser igual as telas da versão *Web*, será feito apenas uma breve descrição sobre elas.

Na Figura 15, apresenta a tela de solicitação de impressão na versão *Mobile*. A tela apresenta o mesmo formulário e funcionamento da versão *Web*.

Figura 15 – Tela de Solicitar Impressão - *Mobile*

06:14 23%

**Criar**

### Solicitar Impressão

**Solicitante**  
Professor

**Procedência (Setor de Origem)**  
Setor de Professores

**Pessoa a Retirar**  
Professor

Outra pessoa irá retirar

**Tipo de Impressão**  
Escolha o tipo de impressão...

**N° de Páginas (Originais)**  
Ex: 2

**N° de Cópias por Original**  
Ex: 30

Frente e Verso  Grampear

Solicitações Avisos **Criar** Mensagens Relatórios

Fonte: Elaborado pelo autor

Na Figura 16, apresenta a tela de solicitações abertas, e possui as mesmas funcionalidades e funcionamento da versão *Web*.

Figura 16 – Tela de Solicitações Abertas - *Mobile*

Fonte: Elaborado pelo autor

## 4 CONSIDERAÇÕES FINAIS

Este trabalho partiu da identificação de um processo administrativo manual e ineficiente no setor de Mecnografia do IFPB - Campus Cajazeiras. O fluxo de solicitações de impressão, baseado em interações informais e sem rastreabilidade, gerava gargalos, desperdício de recursos e falhas de comunicação. Diante disso, o objetivo principal deste projeto foi o desenvolvimento do sistema MECANOSYSTEM, uma solução computacional multiplataforma (Web e Mobile) para automatizar e otimizar este serviço. Pode-se afirmar que este objetivo foi plenamente alcançado. Ficando a etapa de validação para o futuro a espera de uma permissão do IFPB Campus Cajazeiras.

Ao longo do desenvolvimento, os objetivos específicos traçados foram cumpridos com sucesso. Foi realizado um mapeamento detalhado do processo existente, que fundamentou o levantamento dos requisitos. A partir da pesquisa de tecnologias, foi projetada e implementada uma arquitetura de sistema que resultou na entrega de duas interfaces funcionais e complementares: Umm aplicativo mobile para os servidores, permitindo a submissão e o acompanhamento de solicitações de impressões remotamente; e uma plataforma web robusta que pode ser utilizada pelos mesmos usuários da versão mobile e principalmente pelo operador da Mecnografia, centralizando o gerenciamento da fila de impressão, o controle de status das solicitações e a geração de relatórios.

As contribuições deste projeto são diversas. Para o IFPB - Campus Cajazeiras, a principal contribuição é a modernização de um serviço essencial, oferecendo uma aplicação que promove a transparência de impressões e possíveis reduções de custos com impressões. Academicamente, este TCC serve como um estudo de caso prático sobre a aplicação de engenharia de software para resolver problemas reais no setor público, agregando conhecimento à área de desenvolvimento de sistemas.

O estudo de novos conceitos também pode ser considerado para um próximo documento, o que pode implicar no levantamento de novas funcionalidades e contribuições para a melhora das aplicações tratada neste trabalho. Um exemplo de melhoria seria a integração da aplicações criado neste trabalho com o Sistema unificado de administração pública (SUAP), além da implementação da funcionalidade não concluída de envio de mensagens. E como trabalho futuro a realização de testes de validação durante a implantação no IFPB-Campus Cajazeiras.

## REFERÊNCIAS

Amazon Web Services. **The difference between JSON and XML**. 2025. <<https://aws.amazon.com/pt/compare/the-difference-between-json-xml/>>. Acesso em: 21 ago. 2025.

\_\_\_\_\_. **NoSQL Document Databases**. 2025. <<https://aws.amazon.com/pt/nosql/document/>>. Acesso em: 21 ago. 2025.

AUTOMATION, T. **How to view job information for Equitrac Follow-You Printing**. 2024. Acesso em: 04 set. 2025. Disponível em: <[https://docshield.tungstenautomation.com/ControlSuite/en\\_US/1.3.0-wz4krwhe7f/help/CS/DRS/Source/KonicaMinoltaCC/Dita/UCK5501\\_HowViewPrintManagerJob.html](https://docshield.tungstenautomation.com/ControlSuite/en_US/1.3.0-wz4krwhe7f/help/CS/DRS/Source/KonicaMinoltaCC/Dita/UCK5501_HowViewPrintManagerJob.html)>.

\_\_\_\_\_. **Launcher screen — Equitrac client**. 2024. Acesso em: 04 set. 2025. Disponível em: <[https://docshield.tungstenautomation.com/ControlSuite/en\\_US/help/clients/DRS/XeroxUC/ControlSuite\\_XeroxUC/c\\_launcherxerox.html](https://docshield.tungstenautomation.com/ControlSuite/en_US/help/clients/DRS/XeroxUC/ControlSuite_XeroxUC/c_launcherxerox.html)>.

\_\_\_\_\_. **Login with Equitrac only — Unified Client for Brother**. 2024. Acesso em: 04 set. 2025. Disponível em: <[https://docshield.tungstenautomation.com/ControlSuite/en\\_US/help/clients/DRS/BrotherUC/ControlSuite\\_BrotherUC/t\\_BRUC\\_Login\\_EQ.html](https://docshield.tungstenautomation.com/ControlSuite/en_US/help/clients/DRS/BrotherUC/ControlSuite_BrotherUC/t_BRUC_Login_EQ.html)>.

BARBOSA, G. M. G. **Um Livro-texto para o Ensino de Projeto de Arquitetura de Software**: Dissertação (mestrado) — universidade federal de campina grande - ufcg, campina grande - pb, 2009. Accessed: 29 de ago. de 2022. Disponível em: <<http://dspace.sti.ufcg.edu.br:8080/xmlui/bitstream/handle/riufcg/12033/GUILHERME%20MAURO%20GERMOGLIO%20BARBOSA%20-%20DISSERTA%20c3%87%c3%83O%20%28PPGCC%29%202009.pdf?sequence=1&isAllowed=y>>., 2009.

FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**.: Subtitulo. Accessed: 29 de ago. de 2022. Disponível em:<[https://www.ics.uci.edu/fielding/pubs/dissertation/fielding\\_dissertation.pdf](https://www.ics.uci.edu/fielding/pubs/dissertation/fielding_dissertation.pdf) >, 2021.

FORESTI, L. F. **Prestação de Serviços de Reprografia**. [S.l.], 2014.

GARLAND, J.; RICHARD, A. **Large-scale software architecture**.: Subtitulo. Wiley, 2003.

LATTANZE, A. J. **Architecting Software Intensive Systems: A Practitioners Guide**.: Subtitulo. Accessed: 29 de ago. de 2022. Disponível em:, 2009.

LUCIANO, J. ALVES, W. J. B. **ARCHITECTURAL PATTERN MVC: MODEL-VIEW-CONTROLLER**. [S.l.], 2011.

MARTIN, R. C. **Clean Architecture: A CRAFTSMAN'S GUIDE TO SOFTWARE STRUCTURE AND DESIGN**. [S.l.], 2017.

MORAES, N. **MELHORIA E AUTOMATIZAÇÃO DE PROCESSOS**: Benefícios para organizaÇÃO. Accessed: 29 de ago. de 2022. Disponível em: <<https://www.candidomendes.edu.br/wp-content/uploads/2019/10/MELHORIA-E-AUTOMATIZAC%CC%A7A%CC%83O-DE-PROCESSOS.pdf>>, 2018.

MOURA, A. C. Automação e controle de processos administrativos em instituições públicas. **Revista de Gestão e Tecnologia**, v. 18, n. 2, p. 77–91, 2021. Acesso em: 29 ago. 2025. Disponível em: <<https://desafioonline.ufms.br/index.php/EIGEDIN/article/download/16587/11999/>>.

OLIVEIRA, L. F. D. **ENTENDENDO A TRIÁDE MODEL-VIEW-CONTROLLER (MVC) UTILIZANDO PADRÕES DE PROJETO DE SOFTWARE ORIENTADO A OBJETOS.** Dissertação (mestrado) — faculdade anhanguera de campinas, 2013. Accessed: 29 de ago. de 2022. Disponível em:<<http://conic-semesp.org.br/anais/files/2013/trabalho-1000014483.pdf>>., 2013.

SAES, D. X. **GESTÃO DA INOVAÇÃO E TECNOLOGIA:** GraduaÇÃo administraÇÃo. [S.I.], 2012.

SILVA, M. A.; PEREIRA, T. S. Sustentabilidade e racionalização no uso de impressões em ambientes corporativos. **Revista Gestão & Sustentabilidade**, v. 4, n. 2, p. 101–115, 2018. Acesso em: 29 ago. 2025. Disponível em: <[https://www.researchgate.net/publication/373886075\\_EDUCACAO\\_AMBIENTAL\\_E\\_SUSTENTABILIDADE\\_PRATICAS\\_REFLEXOES\\_E\\_FAZERES](https://www.researchgate.net/publication/373886075_EDUCACAO_AMBIENTAL_E_SUSTENTABILIDADE_PRATICAS_REFLEXOES_E_FAZERES)>.

SOFTWARE, P. **About mobile and tablet printing — PaperCut Hive & Pocket.** 2024. Acesso em: 04 set. 2025. Disponível em: <<https://www.papercut.com/help/manuals/pocket-hive/overview/easy-printing/about-mobile-and-tablet-printing/>>.

\_\_\_\_\_. **Dashboard — PaperCut NG/MF Admin web interface.** 2024. Acesso em: 04 set. 2025. Disponível em: <<https://www.papercut.com/help/manuals/ng-mf/common/sys-dashboard/>>.

\_\_\_\_\_. **Find-Me Printing — setup.** 2024. Acesso em: 04 set. 2025. Disponível em: <<https://www.papercut.com/help/manuals/ng-mf/applicationserver/find-me-printing-setup-mf/>>.

\_\_\_\_\_. **Overview — PaperCut NG/MF (features overview).** 2024. Acesso em: 04 set. 2025. Disponível em: <<https://www.papercut.com/help/manuals/ng-mf/common/feature/>>.

SOMMERVILLE, IAN. **Engenharia de Software.**: 6a.edição, addison-wesley/pearson, 2003. [S.I.], 2013.

SOUZA, P. R. Políticas de sustentabilidade e uso consciente de recursos gráficos em universidades públicas. **Revista de Administração Pública e Sustentabilidade**, v. 6, n. 1, p. 33–48, 2020. Acesso em: 29 ago. 2025. Disponível em: <[https://www.researchgate.net/publication/342086907\\_Educacao\\_Ambiental\\_e\\_sustentabilidade\\_uma\\_intervencao\\_emergente\\_na\\_escola](https://www.researchgate.net/publication/342086907_Educacao_Ambiental_e_sustentabilidade_uma_intervencao_emergente_na_escola)>.

TOBALDINI, R. G. **Aplicação do Estilo Arquitetural REST a um Sistema de Congressos.** Accessed: 29 de ago. de 2022. Disponível em: <[https://repositorio.ufsc.br/bitstream/handle/123456789/183993/TCCRicardoGhisiTobaldini\\_final.pdf?sequence=-1&isAllowed=y](https://repositorio.ufsc.br/bitstream/handle/123456789/183993/TCCRicardoGhisiTobaldini_final.pdf?sequence=-1&isAllowed=y)>., 2021.

TSURUTA, M. **UM ESTUDO SOBRE A RELAÇÃO ENTRE QUALIDADE E ARQUITETURA DE SOFTWARE.**: Dissertação (mestrado) — universidade de são paulo - usp, são paulo - sp, 2010. Accessed: 29 de ago. de 2022. Disponível em:<<https://www.teses.usp.br/teses/disponiveis/12/12139/tde-29032011-201659/publico/MauricioTsuruta.pdf>>., 2010.

VINÍCIUS, MARCOS. **DESENVOLVIMENTO DE APIS BASEADAS EM REST PARA INTEGRAÇÃO E CONSTRUÇÃO DE APLICAÇÕES**: Curso de bacharelado em análise de sistemas e tecnologia da informação. Accessed: 29 de ago. de 2022. Disponível em:., 2013.

## APÊNDICE A – REQUISITOS DA APLICAÇÃO

### *User Stories*

Esta seção apresenta as funcionalidades que são necessárias, ou interessantes, para os usuários da aplicação e que dão origem a alguns dos Requisitos funcionais.

<b>US01 - Solicitação de impressão</b>
--

Como um professor e futuro usuário da aplicação MecanoSystem, gostaria de poder criar solicitações de impressão.
--

<b>US02 - Gerar relatório</b>
-------------------------------

Como um mecânico e futuro usuário da aplicação MecanoSystem, gostaria de poder gerar relatórios das impressões realizadas.
--

<b>US03 - Publicar notificações</b>
-------------------------------------

Como um mecânico e futuro usuário da aplicação MecanoSystem, gostaria de poder publicar notificações para os usuários da aplicação.
---

<b>US04 - Enviar mensagens</b>
--------------------------------

Como um servidor e futuro usuário da aplicação MecanoSystem, gostaria de poder enviar mensagens para o mecânico.
--

### Requisitos Funcionais

<b>Requisito</b>	RF01 - Manter conta
<b>Resumo</b>	A aplicação deve fazer a permanência das contas do usuários, bem como as operações de atualização e leitura.
<b>Dados necessários</b>	<b>Dados comuns aos módulos</b> <ul style="list-style-type: none"><li>• Matricula</li><li>• Nome</li><li>• Email</li><li>• Senha</li><li>• Setor</li><li>• Disciplina</li><li>• Telefone</li></ul> <b>Módulo Professor</b> <ul style="list-style-type: none"><li>• Disciplinas que leciona</li></ul>

<b>Requisito</b>	RF02 - Manter solicitação
<b>Resumo</b>	A aplicação deve fazer a permanência das solicitações de impressões dos usuários, bem como as operações de atualização, leitura e exclusão.
<b>Dados necessários</b>	<ul style="list-style-type: none"> <li>• Id do servidor que solicitou a impressão</li> <li>• Tipo serviço, que tipo de arquivo o servidor deseja imprimir</li> <li>• Número de original, total de páginas que possui o arquivo que o servidor deseja imprimir</li> <li>• Número de cópias do arquivo original que o servidor deseja imprimir</li> <li>• Total de cópias, total de páginas que serão impressas</li> <li>• Frente verso, se a impressão será frente e verso</li> <li>• Grampear, se a impressão deve ser grampeada</li> <li>• Status, qual o atual estado da impressão</li> <li>• Data solicitação que o servidor realizou a solicitação de impressão</li> <li>• Data em que foi realizada a entrega do documento impresso</li> <li>• Recebido por, nome da pessoa que recebeu o documento impresso</li> <li>• Documento, arquivo que será impresso</li> </ul>

<b>Requisito</b>	RF03 - Emitir Relatório
<b>Resumo</b>	A aplicação deve permitir que o mecanógrafo possa emitir relatórios com todas as suas devidas informações já preenchidas.
<b>Dados necessários</b>	

<b>Requisito</b>	RF04 - Manter Publicação de Notificação
<b>Resumo</b>	A aplicação deve fazer a permanência das Publicação de Notificação do mecanógrafo, bem como as operações de atualização, leitura e exclusão.
<b>Dados necessários</b>	<ul style="list-style-type: none"> <li>• Id do mecanógrafo que fez a publicação</li> <li>• Conteúdo da publicação</li> <li>• Data em que foi realizada a publicação</li> </ul>

<b>Requisito</b>	RF05 - Manter mensagens
<b>Resumo</b>	A aplicação deve fazer a permanência das mensagens enviadas entre o mecanógrafo e os servidores, bem como as operações de atualização, leitura e exclusão.
<b>Dados necessários</b>	<ul style="list-style-type: none"> <li>• Id do mecanógrafo que enviou ou recebeu mensagens</li> <li>• Id do Servido que enviou ou recebeu mensagens</li> <li>• Conteudo da mensagem enviada pelo mecanógrafo ou servidor</li> <li>• Data qua foi enviada</li> </ul>

<b>Requisito</b>	RF06 - Manter conta do Administrador
<b>Resumo</b>	A aplicação deve fazer a permanência da conta do administrador, bem como as operações de atualização e exclusão.
<b>Dados necessários</b>	<ul style="list-style-type: none"> <li>• Nome do servidor que ira administrar a aplicação</li> <li>• Matricula do servidor que ira administrar a aplicação</li> <li>• Senha para realizar o login</li> </ul>

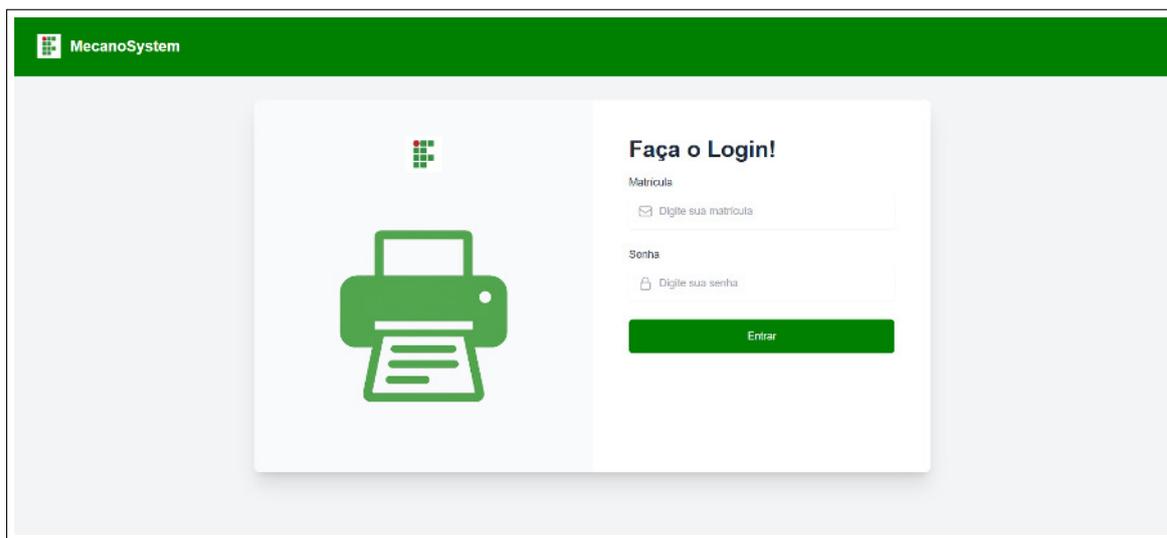
<b>Requisito</b>	RF07 - Manter setor
<b>Resumo</b>	A aplicação deve fazer a permanência dos setores cadastrados pelo administrador, bem como as operações de atualização, leitura e exclusão.
<b>Dados necessários</b>	<ul style="list-style-type: none"><li>• Identificador do administrador que cadastrou o setor</li><li>• Nome do setor cadastrado pelo administrador</li><li>• Sigla do setor cadastrado pelo administrador</li></ul>



## APÊNDICE C – IMAGENS DAS TELAS DO SISTEMA *WEB*

A Figura 18, apresenta a tela onde os usuários realizam o login digitando sua matrícula e senha.

Figura 18 – Tela de Login - *Web*



Fonte: Elaborado pelo autor

A Figura 19, apresenta a tela de cadastrar setores, essa tela é apenas para o usuário Administrador. Para um novo setor ser cadastrado na aplicação o administrador deve preencher os campos: nome do setor e sigla, e depois clicar no botão salvar setor para cadastrar o setor

**Figura 19 – Tela de Cadastrar Setores - Web**

**Cadastrar um novo Setor**  
Painel do Administrador / Setores / Criar

Nome do Setor

Sigla

Cancelar Salvar Setor

Fonte: Elaborado pelo autor

A Figura 20, apresenta a tela de gerenciamento dos setores cadastrados, essa tela é apenas para o usuário Administrador. Com funcionalidades para editar e excluir setores.

**Figura 20 – Tela de Gerenciamento de Setores - Web**

**Gerenciamento de Setores**  
Painel do Administrador / Setores

+ Novo Setor

Nome ▾  Pesquisar

Nome do Setor	Sigla	Ações
Setor de Mecanografia	SM	 
Setor dos Professores	SP	 
Setor dos Servidores	SS	 

Página 1 de 1 (3 setores no total)

Fonte: Elaborado pelo autor

A Figura 22, apresenta a tela de cadastrar usuários, essa tela é apenas para o usuário Administrador. Para um novo usuário ser cadastrado na aplicação o administrador deve preencher os campos: matrícula, nome completo, telefone, e-mail, senha, tipo de usuário, o setor a qual o novo usuário irá pertencer e a disciplina lecionada caso o usuário seja um professor. Preenchidos os dados o usuário deve clicar no botão salvar usuário para que seja cadastrado.

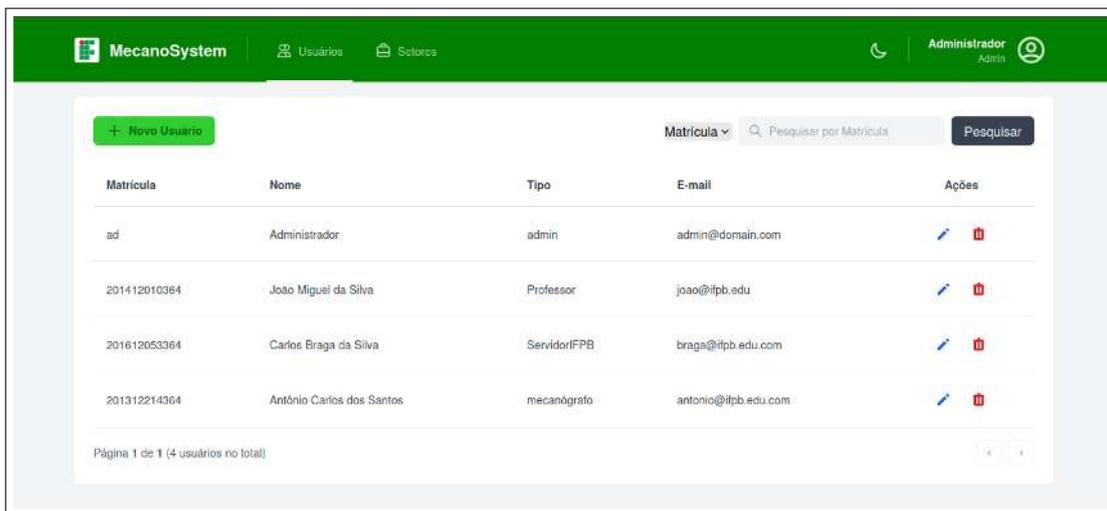
**Figura 21 – Tela de Cadastrar Usuários - Web**

A imagem mostra a interface de usuário para o cadastro de um novo usuário no sistema MecanoSystem. O cabeçalho verde contém o logo 'MecanoSystem', links para 'Usuários' e 'Setores', e o perfil do usuário 'Administrador Admin'. O título da página é 'Cadastrar um novo usuário' com o caminho de navegação 'Painel do Administrador / Usuários / Cria'. O formulário principal possui os seguintes campos: 'Matrícula', 'Nome Completo', 'Telefone', 'E-mail', 'Senha', 'Tipo de Usuário' (menu suspenso com 'Professor' selecionado), 'Setor' (menu suspenso com 'Selecione um setor'), e 'Disciplina'. Na base do formulário, há dois botões: 'Cancelar' e 'Salvar Usuário'.

Fonte: Elaborado pelo autor

A Figura 22, apresenta a tela de gerenciamento de usuários cadastrados, essa tela é apenas para o usuário Administrador. Com funcionalidades para editar e excluir usuários.

Figura 22 – Tela de Gerenciamento de Usuários - Web



Fonte: Elaborado pelo autor

A Figura 23, apresenta a tela de editar solicitações, ela primeiramente exibi as informações da solicitação que o usuário deseja editar bloqueadas, e caso o usuário deseje editar qualquer informação da solicitação, deve clicar no botão de editar fazer as modificações e salvar.

**Figura 23 – Tela de Editar Solicitações - Web**

**Editar Solicitação**  
Painel / Solicitação de Impressão / Detalhes

Procedência (Setor de Origem)  
Setor de Mecanografos

Solicitante  
Professor

Pessoa a Retirar  
Professor

Tipo de Impressão  
Provas

Número de Páginas (Originais)  
#

Número de Cópia por Original  
2

Quero impresso em (Data)  
28/08/2025

Quero impresso em (Hora)  
02:00

Frente e Verso

Grampear

Arquivo  
2023 - gabarito.pdf

Observações

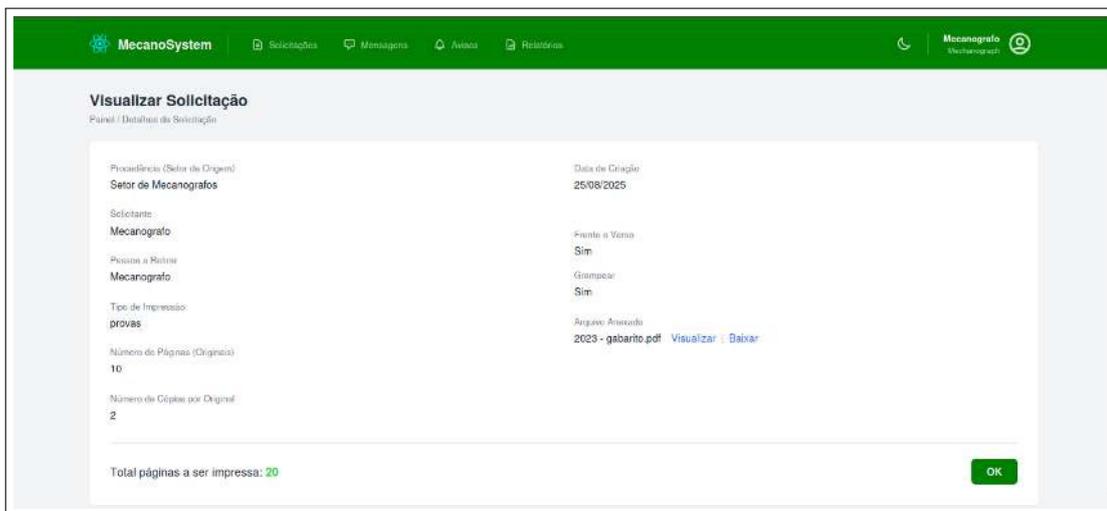
Total páginas a ser impressa: 0

OK Editar

Fonte: Elaborado pelo autor

A Figura 24, apresenta a tela onde os usuários podem visualizar as informações clicando no botão de visualizar em uma das solicitações listadas na página de solicitações abertas.

Figura 24 – Tela de Visualizar Solicitações - Web



Fonte: Elaborado pelo autor

A Figura 25, apresenta a tela para a publicação de avisos pelo usuário mecano-  
nografo, nela temos os dois campos o primeiro deve ser digitado o título do aviso e  
o segundo o conteúdo, quando preenchidos os dois campos o usuário deve enviar  
o aviso clicando no botão de enviar. Nessa interface também é listado os avisos já  
publicados, podendo ser editados ou excluídos.

Figura 25 – Tela de Publicação de Avisos - Web

**MecanoSystem** Solicitações Mensagens Avisos Relatórios Mecanógrafo Mechanograph

### Gerenciamento de Avisos

Panel / Avisos

#### Publicar Novo Aviso

Digite um Título

Digite um Aviso

**Enviar**

#### Avisos Publicados

Postado em: 24/08/2025, 10:24:08

**Aviso Teste!**

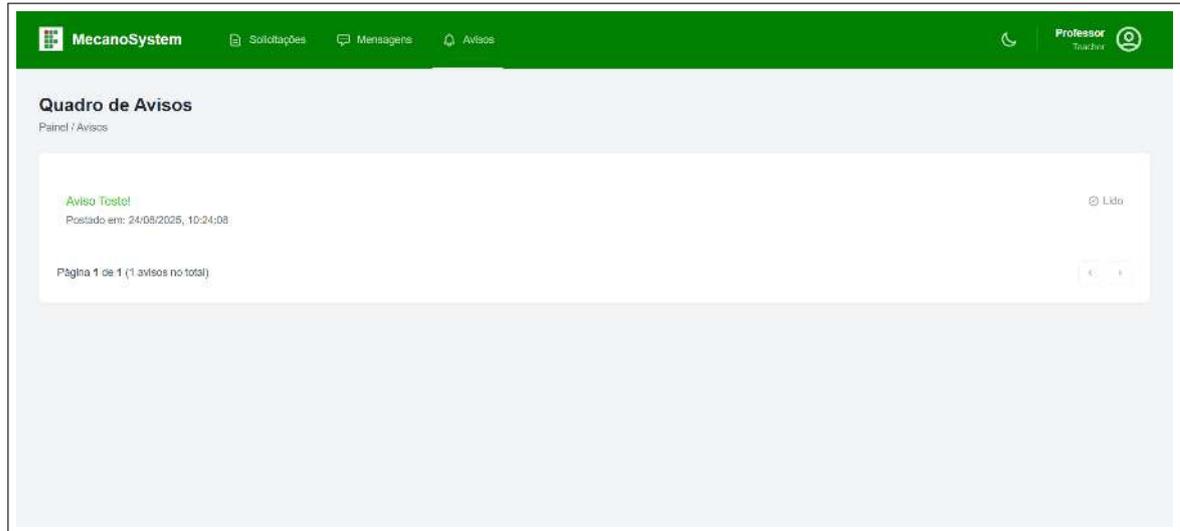
Esse aviso é apenas um aviso teste!

Página 1 de 1 (1 avisos no total)

Fonte: Elaborado pelo autor

A Figura 26, apresenta a tela onde os outros usuários recebem os avisos publicados pelo mecanógrafo, são organizados em uma lista e marcado como lido quando o usuário tenha lido o aviso.

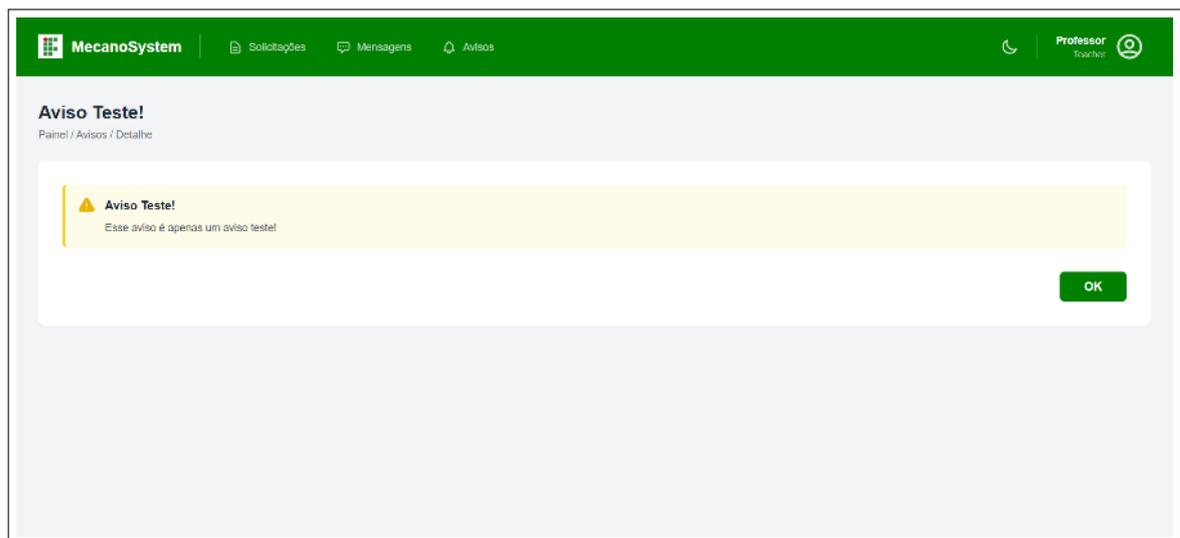
**Figura 26 – Tela de Lista de Avisos - Web**



Fonte: Elaborado pelo autor

A Figura 27, apresenta a tela onde o conteúdo dos avisos são lidos pelos usuários, após ter clicado em um aviso da lista.

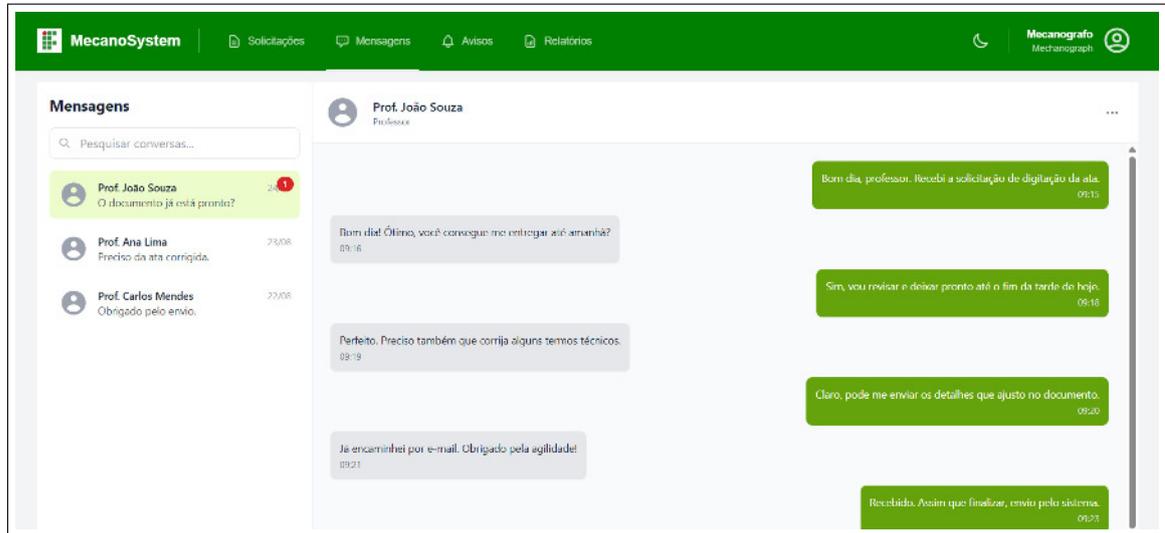
**Figura 27 – Tela para Ler um Aviso - Web**



Fonte: Elaborado pelo autor

A Figura 28, apresenta a tela onde acontece a comunicação entre o usuário que solicitou uma impressão e o mecanógrafo, através de mensagens.

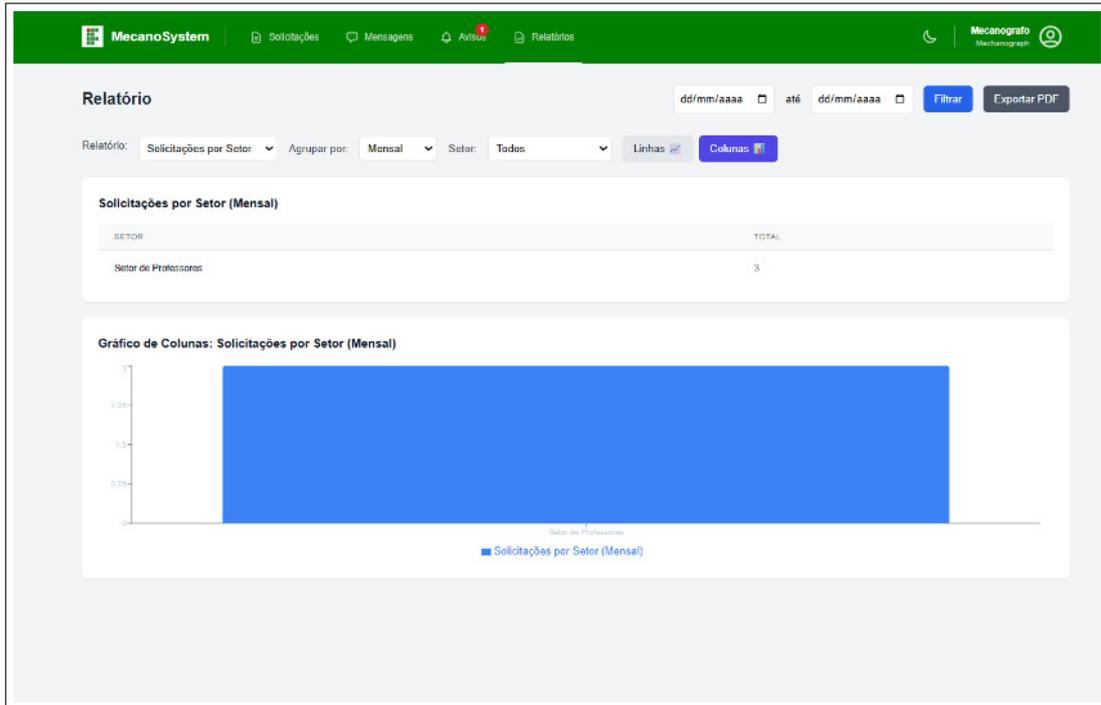
**Figura 28 – Tela de Mensagens - Web**



Fonte: Elaborado pelo autor

A Figura 29, apresenta a tela onde os usuários podem ver relatórios sobre todas as solicitações realizadas na aplicação no caso do mecanógrafo e sobre suas próprias solicitações no caso dos outros usuários.

Figura 29 – Tela de Relatórios - Web



Fonte: Elaborado pelo autor

## APÊNDICE D – IMAGENS DAS TELAS DO SISTEMA *MOBILE*

Por as telas apresentadas a seguir possuir o mesmo funcionamento da versão *Web* e para evitar repetição, será escrito apenas uma pequena descrição sobre elas. Para saber mais detalhes Apêndice C.

Na Figura 30, apresenta a tela onde os usuários realizam o login digitando sua matrícula e senha.

Figura 30 – Tela de Login - *Mobile*



Fonte: Elaborado pelo autor

Na Figura 31, apresenta a tela de editar uma solicitação.

**Figura 31 – Tela de Editar Solicitação - Mobile**



The screenshot displays the 'Editar Solicitação' (Edit Request) screen on a mobile device. The interface is clean and uses a light gray color scheme for the form fields. At the top, there is a green header bar with a back arrow and the title 'Editar Solicitação'. The status bar at the very top shows the time as 06:36, signal strength, Wi-Fi, and a 21% battery level. The main content area is titled 'Editar Solicitação' and contains several sections:

- Solicitante:** A text field containing 'Professor'.
- Procedência (Setor de Origem):** A text field containing 'Setor de Professores'.
- Pessoa a Retirar:** A text field with a person icon and the text 'Professor'.
- Tipo de Impressão:** A dropdown menu currently showing 'Selecione...'.
- Nº de Páginas (Originals):** A text field with a document icon and the value '10'.
- Nº de Cópias por Original:** A text field with a document icon and the value '2'.
- Printing Options:** Two toggle switches. 'Frente e Verso' is currently turned off, and 'Grampear' is currently turned on.
- Observações:** A text area with the placeholder text 'Escreva aqui, observações sobre a impressão...'.

Fonte: Elaborado pelo autor

Na Figura 32, apresenta a tela onde os usuários podem ver detalhes sobre uma solicitação criada.

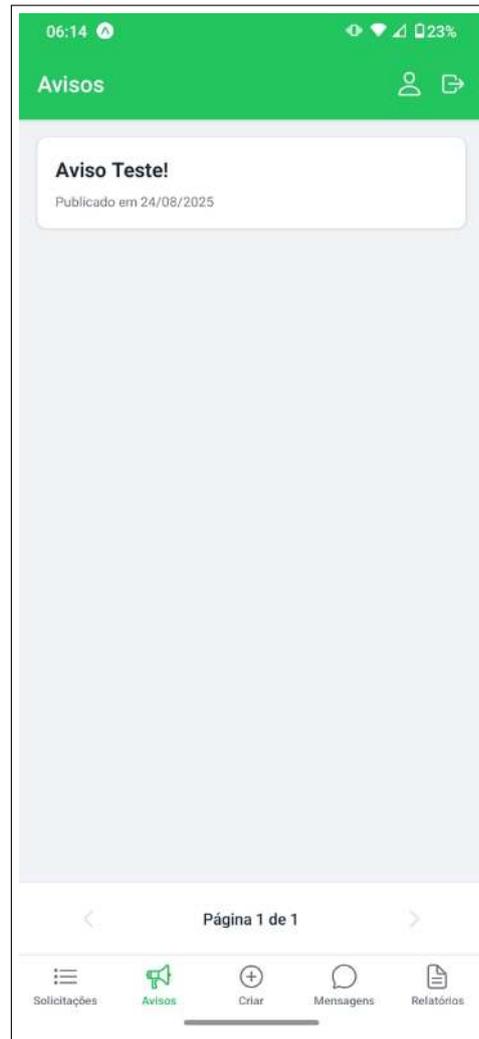
**Figura 32 – Tela de Visualizar Solicitação - Mobile**



Fonte: Elaborado pelo autor

Na Figura 33, apresenta a tela onde é listados os avisos publicados pelo mecânico para todos os usuários em forma de lista.

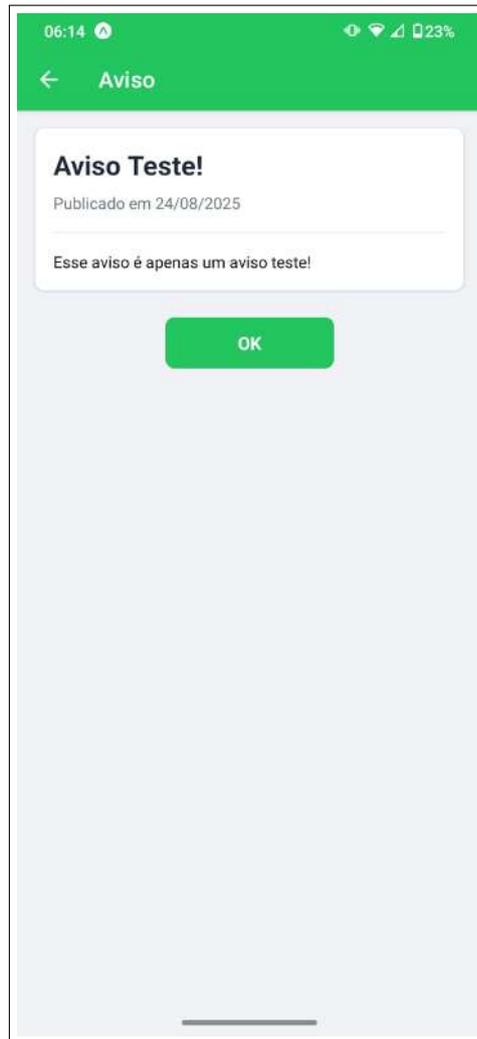
**Figura 33 – Tela de Lista de Avisos - Mobile**



Fonte: Elaborado pelo autor

Na Figura 34, apresenta a tela onde os usuários podem ler o conteúdo de um aviso clicando na lista de avisos.

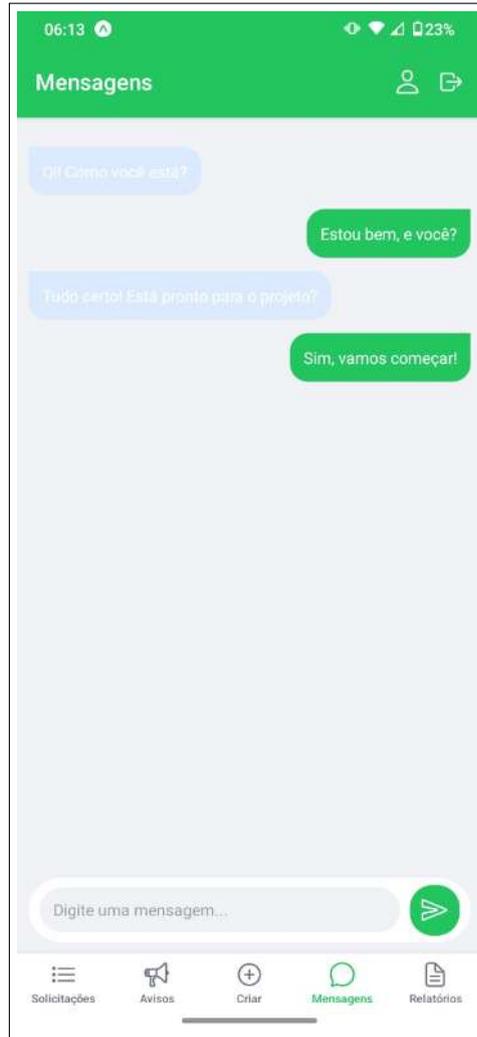
**Figura 34 – Tela de Ler Aviso - Mobile**



Fonte: Elaborado pelo autor

Na Figura 35, apresenta a tela para envio de mensagens entre o usuário solicitante e o mecanógrafo.

**Figura 35 – Tela de Mensagens - Mobile**



Fonte: Elaborado pelo autor

Na Figura 36, apresenta a tela onde os usuários podem ver relatórios sobre as suas solicitações.

Figura 36 – Tela de Relatórios - *Mobile*



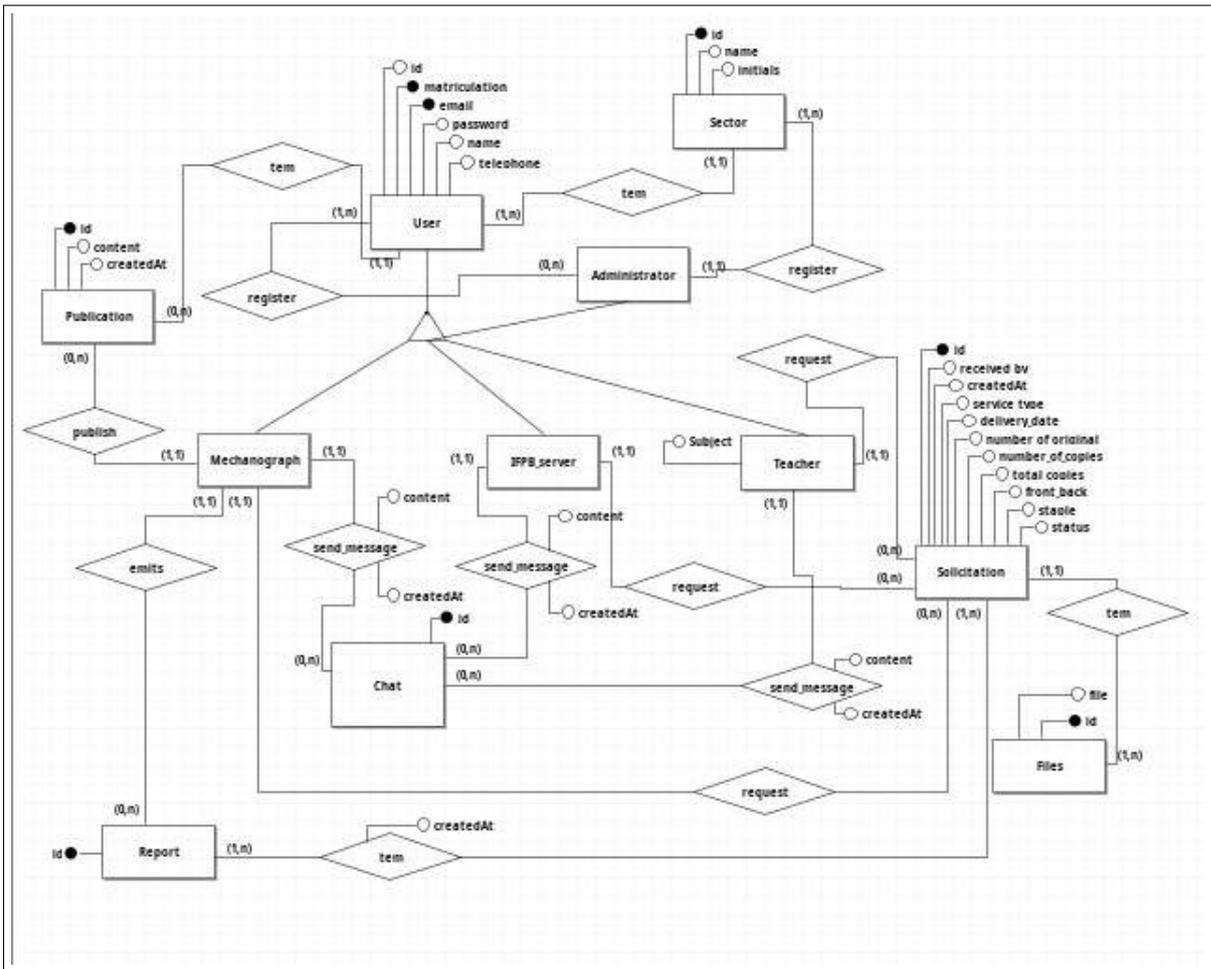
Fonte: Elaborado pelo autor

## **APÊNDICE E – ARQUITETURA DO BANCO DE DADOS DO SISTEMA**

### **Descrição do Mini-Mundo**

A base de dados irá armazenar as informações da aplicação MecanoSystem, A aplicação se trata de um software para auxiliar o mecanógrafo no gerenciamento das solicitações de impressão para a mecanografia do IFPB - Campus Cajazeiras, e servidores do campus a acompanharem suas solicitações de impressão. Na aplicação é necessário armazenar os dados de várias entidades que serão cruciais para o seu funcionamento, tais como: Administrador, Usuário, Mecanógrafo, Servidor, Professor, Publicação, Setor e várias outras. As entidades restantes e os dados que devem ser preenchidos para cada uma estão apresentados no modelo lógico do banco de dados abaixo.

Figura 37 – Modelo Entidade Relacionamento

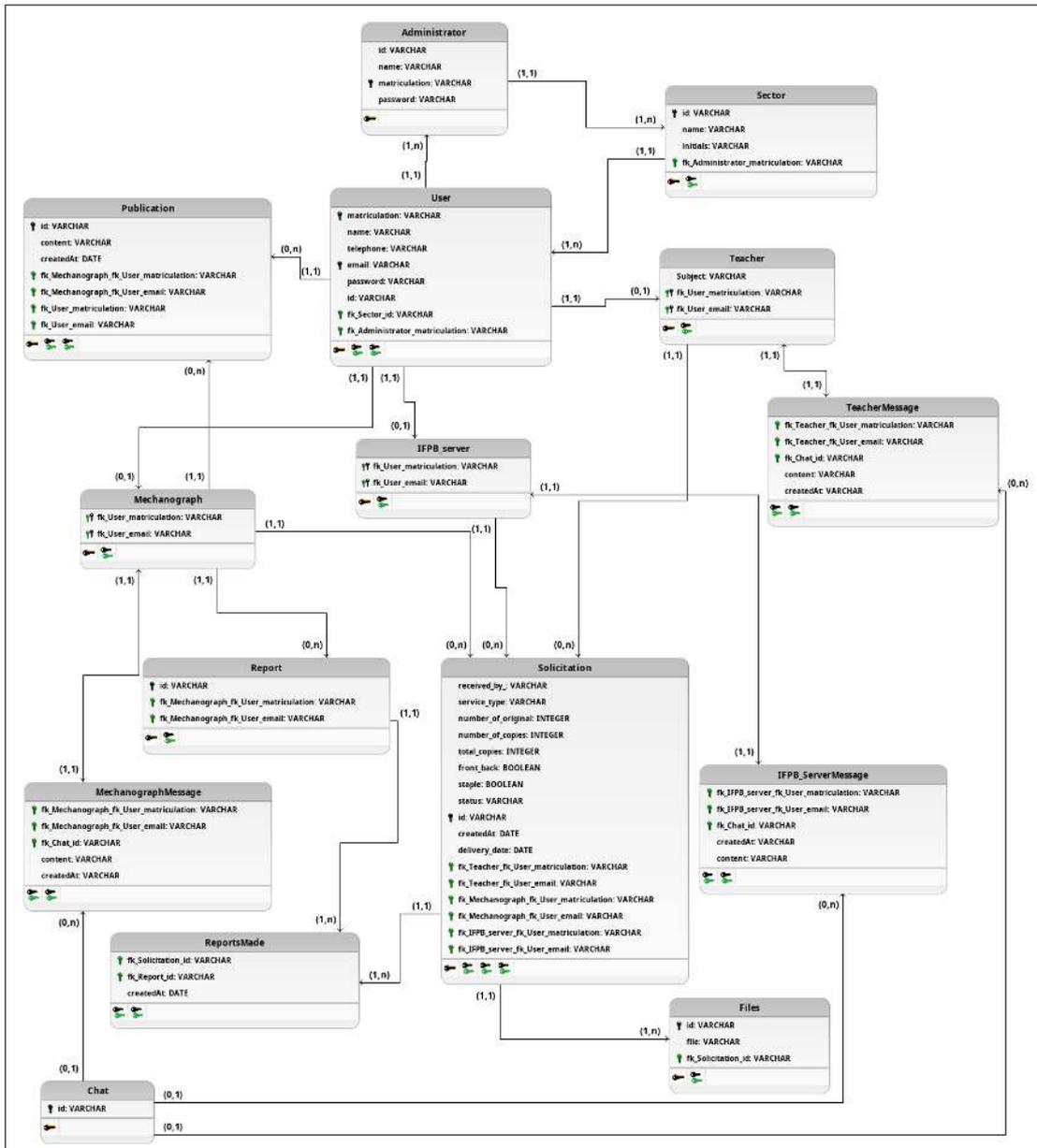


Fonte: Elaborado pelo autor

### Modelo Lógico

A representação lógica do banco de dados apresenta um modelo mais próximo da representação física do banco de dados.

Figura 38 – Modelo lógico



Fonte: Elaborado pelo autor

### Modelo Físico

Esta seção apresenta o modelo físico da base de dados gerado a partir do modelo lógico que se encontra na seção anterior. O código gerado segue os comandos e regras utilizados pelo PostgreSQL, uma vez que é a ferramenta utilizada na aplicação para a permanência dos dados.

```

1 CREATE TABLE Administrator (
2     id VARCHAR,
3     name VARCHAR,
4     matriculation VARCHAR PRIMARY KEY,
5     password VARCHAR
6 );
7
8 CREATE TABLE Sector (
9     id VARCHAR PRIMARY KEY,
10    name VARCHAR,
11    initials VARCHAR,
12    fk_Administrator_matriculation VARCHAR
13 );
14
15 CREATE TABLE User (
16    matriculation VARCHAR,
17    name VARCHAR,
18    telephone VARCHAR,
19    email VARCHAR,
20    password VARCHAR,
21    id VARCHAR,
22    fk_Sector_id VARCHAR,
23    fk_Administrator_matriculation VARCHAR,
24    PRIMARY KEY (matriculation, email)
25 );
26
27 CREATE TABLE Mechanograph (
28    fk_User_matriculation VARCHAR,
29    fk_User_email VARCHAR,
30    PRIMARY KEY (fk_User_matriculation, fk_User_email)
31 );
32
33 CREATE TABLE IFPB_server (
34    fk_User_matriculation VARCHAR,
35    fk_User_email VARCHAR,
36    PRIMARY KEY (fk_User_matriculation, fk_User_email)
37 );
38
39 CREATE TABLE Teacher (
40    Subject VARCHAR,
41    fk_User_matriculation VARCHAR,
42    fk_User_email VARCHAR,
43    PRIMARY KEY (fk_User_matriculation, fk_User_email)
44 );

```

Fonte: Elaborado pelo autor

```

1
2 CREATE TABLE Publication (
3     id VARCHAR PRIMARY KEY,
4     content VARCHAR,
5     createdAt DATE,
6     fk_Mechanograph_fk_User_matriculation VARCHAR,
7     fk_Mechanograph_fk_User_email VARCHAR,
8     fk_User_matriculation VARCHAR,
9     fk_User_email VARCHAR
10 );
11
12 CREATE TABLE Solicitation (
13     received_by_ VARCHAR,
14     service_type VARCHAR,
15     number_of_original INTEGER,
16     number_of_copies INTEGER,
17     total_copies INTEGER,
18     front_back BOOLEAN,
19     staple BOOLEAN,
20     status VARCHAR,
21     id VARCHAR PRIMARY KEY,
22     createdAt DATE,
23     delivery_date DATE,
24     fk_Teacher_fk_User_matriculation VARCHAR,
25     fk_Teacher_fk_User_email VARCHAR,
26     fk_Mechanograph_fk_User_matriculation VARCHAR,
27     fk_Mechanograph_fk_User_email VARCHAR,
28     fk_IFPB_server_fk_User_matriculation VARCHAR,
29     fk_IFPB_server_fk_User_email VARCHAR
30 );
31
32 CREATE TABLE Files (
33     id VARCHAR PRIMARY KEY,
34     file VARCHAR,
35     fk_Solicitation_id VARCHAR
36 );
37
38 CREATE TABLE Report (
39     id VARCHAR PRIMARY KEY,
40     fk_Mechanograph_fk_User_matriculation VARCHAR,
41     fk_Mechanograph_fk_User_email VARCHAR
42 );
43
44 CREATE TABLE ReportsMade (
45     fk_Solicitation_id VARCHAR,
46     fk_Report_id VARCHAR,
47     createdAt DATE
48 );

```

```

1
2 CREATE TABLE Chat (
3     id VARCHAR PRIMARY KEY
4 );
5
6 CREATE TABLE MechanographMessage (
7     fk_Mechanograph_fk_User_matriculation VARCHAR,
8     fk_Mechanograph_fk_User_email VARCHAR,
9     fk_Chat_id VARCHAR,
10    content VARCHAR,
11    createdAt VARCHAR
12 );
13
14 CREATE TABLE IFPB_ServerMessage (
15     fk_IFPB_server_fk_User_matriculation VARCHAR,
16     fk_IFPB_server_fk_User_email VARCHAR,
17     fk_Chat_id VARCHAR,
18     createdAt VARCHAR,
19     content VARCHAR
20 );
21
22 CREATE TABLE TeacherMessage (
23     fk_Teacher_fk_User_matriculation VARCHAR,
24     fk_Teacher_fk_User_email VARCHAR,
25     fk_Chat_id VARCHAR,
26     content VARCHAR,
27     createdAt VARCHAR
28 );
29
30 ALTER TABLE Sector ADD CONSTRAINT FK_Sector_2
31     FOREIGN KEY (fk_Administrator_matriculation)
32     REFERENCES Administrator (matriculation)
33     ON DELETE RESTRICT;
34
35 ALTER TABLE User ADD CONSTRAINT FK_User_2
36     FOREIGN KEY (fk_Sector_id)
37     REFERENCES Sector (id)
38     ON DELETE RESTRICT;
39
40 ALTER TABLE User ADD CONSTRAINT FK_User_3
41     FOREIGN KEY (fk_Administrator_matriculation)
42     REFERENCES Administrator (matriculation)
43     ON DELETE RESTRICT;
44
45 ALTER TABLE Mechanograph ADD CONSTRAINT FK_Mechanograph_2
46     FOREIGN KEY (fk_User_matriculation, fk_User_email)
47     REFERENCES User (matriculation, email)
48     ON DELETE CASCADE;

```

```

1
2 ALTER TABLE IFPB_server ADD CONSTRAINT FK_IFPB_server_2
3     FOREIGN KEY (fk_User_matriculation, fk_User_email)
4     REFERENCES User (matriculation, email)
5     ON DELETE CASCADE;
6
7 ALTER TABLE Teacher ADD CONSTRAINT FK_Teacher_2
8     FOREIGN KEY (fk_User_matriculation, fk_User_email)
9     REFERENCES User (matriculation, email)
10    ON DELETE CASCADE;
11
12 ALTER TABLE Publication ADD CONSTRAINT FK_Publication_2
13    FOREIGN KEY (fk_Mechanograph_fk_User_matriculation,
14                fk_Mechanograph_fk_User_email)
15    REFERENCES Mechanograph (fk_User_matriculation, fk_User_email)
16    ON DELETE CASCADE;
17
18 ALTER TABLE Publication ADD CONSTRAINT FK_Publication_3
19    FOREIGN KEY (fk_User_matriculation, fk_User_email)
20    REFERENCES User (matriculation, email)
21    ON DELETE CASCADE;
22
23 ALTER TABLE Solicitation ADD CONSTRAINT FK_Solicitation_2
24    FOREIGN KEY (fk_Teacher_fk_User_matriculation,
25                fk_Teacher_fk_User_email)
26    REFERENCES Teacher (fk_User_matriculation, fk_User_email)
27    ON DELETE CASCADE;
28
29 ALTER TABLE Solicitation ADD CONSTRAINT FK_Solicitation_3
30    FOREIGN KEY (fk_Mechanograph_fk_User_matriculation,
31                fk_Mechanograph_fk_User_email)
32    REFERENCES Mechanograph (fk_User_matriculation, fk_User_email)
33    ON DELETE CASCADE;
34
35 ALTER TABLE Solicitation ADD CONSTRAINT FK_Solicitation_4
36    FOREIGN KEY (fk_IFPB_server_fk_User_matriculation,
37                fk_IFPB_server_fk_User_email)
38    REFERENCES IFPB_server (fk_User_matriculation, fk_User_email)
39    ON DELETE CASCADE;
40
41 ALTER TABLE Files ADD CONSTRAINT FK_Files_2
42    FOREIGN KEY (fk_Solicitation_id)
43    REFERENCES Solicitation (id)
44    ON DELETE RESTRICT;

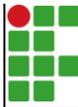
```

Fonte: Elaborado pelo autor

```

1
2 ALTER TABLE Report ADD CONSTRAINT FK_Report_2
3     FOREIGN KEY (fk_Mechanograph_fk_User_matriculation,
4                 fk_Mechanograph_fk_User_email)
5     REFERENCES Mechanograph (fk_User_matriculation, fk_User_email)
6     ON DELETE CASCADE;
7
8 ALTER TABLE ReportsMade ADD CONSTRAINT FK_ReportsMade_1
9     FOREIGN KEY (fk_Solicitation_id)
10    REFERENCES Solicitation (id)
11    ON DELETE RESTRICT;
12
13 ALTER TABLE ReportsMade ADD CONSTRAINT FK_ReportsMade_2
14     FOREIGN KEY (fk_Report_id)
15     REFERENCES Report (id)
16     ON DELETE RESTRICT;
17
18 ALTER TABLE MechanographMessage ADD CONSTRAINT
19     FK_MechanographMessage_1
20     FOREIGN KEY (fk_Mechanograph_fk_User_matriculation,
21                 fk_Mechanograph_fk_User_email)
22     REFERENCES Mechanograph (fk_User_matriculation, fk_User_email)
23     ON DELETE RESTRICT;
24
25 ALTER TABLE MechanographMessage ADD CONSTRAINT
26     FK_MechanographMessage_2
27     FOREIGN KEY (fk_Chat_id)
28     REFERENCES Chat (id)
29     ON DELETE SET NULL;
30
31 ALTER TABLE IFPB_ServerMessage ADD CONSTRAINT FK_IFPB_ServerMessage_1
32     FOREIGN KEY (fk_IFPB_server_fk_User_matriculation,
33                 fk_IFPB_server_fk_User_email)
34     REFERENCES IFPB_server (fk_User_matriculation, fk_User_email)
35     ON DELETE RESTRICT;
36
37 ALTER TABLE IFPB_ServerMessage ADD CONSTRAINT FK_IFPB_ServerMessage_2
38     FOREIGN KEY (fk_Chat_id)
39     REFERENCES Chat (id)
40     ON DELETE SET NULL;
41
42 ALTER TABLE TeacherMessage ADD CONSTRAINT FK_TeacherMessage_1
43     FOREIGN KEY (fk_Teacher_fk_User_matriculation,
44                 fk_Teacher_fk_User_email)
45     REFERENCES Teacher (fk_User_matriculation, fk_User_email)
46     ON DELETE RESTRICT;
47
48 ALTER TABLE TeacherMessage ADD CONSTRAINT FK_TeacherMessage_2
49     FOREIGN KEY (fk_Chat_id)
50     REFERENCES Chat (id)
51     ON DELETE SET NULL;

```

	<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 Restrito

### Documento Final de TCC

<b>Assunto:</b>	Documento Final de TCC
<b>Assinado por:</b>	Kaique Augusto
<b>Tipo do Documento:</b>	Anexo
<b>Situação:</b>	Finalizado
<b>Nível de Acesso:</b>	Restrito
<b>Hipótese Legal:</b>	Informação Pessoal (Art. 31 da Lei no 12.527/2011)
<b>Tipo do Conferência:</b>	Cópia Simples

Documento assinado eletronicamente por:

- **Kaique Augusto de Sousa, DISCENTE (202122010011) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 12/09/2025 13:42:25.

Este documento foi armazenado no SUAP em 12/09/2025. 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: 1607219

Código de Autenticação: ce172b5cd3

