

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Projeto e Desenvolvimento de um Protótipo para o Novo Sistema de Gestão de Trabalhos de Conclusão de Curso do DCC/UFJF

Matheus Henrique Rubio de Melo

JUIZ DE FORA
AGOSTO, 2025

Projeto e Desenvolvimento de um Protótipo para o Novo Sistema de Gestão de Trabalhos de Conclusão de Curso do DCC/UFJF

MATHEUS HENRIQUE RUBIO DE MELO

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Sistemas de Informação
Orientador: Prof. Luciano Jerez Chaves

JUIZ DE FORA
AGOSTO, 2025

PROJETO E DESENVOLVIMENTO DE UM PROTÓTIPO PARA O NOVO SISTEMA DE GESTÃO DE TRABALHOS DE CONCLUSÃO DE CURSO DO DCC/UFJF

Matheus Henrique Rubio de Melo

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Aprovada por:

Prof. Luciano Jerez Chaves
Doutor em Ciência da Computação

Prof. Marco Antônio Pereira Araújo
Doutor em Engenharia de Sistemas e Computação

Prof. Ronney Moreira de Castro
Doutor em Informática

JUIZ DE FORA
14 DE AGOSTO, 2025

Aos meus amigos e irmãos.

Aos pais, pelo apoio e sustento.

À minha esposa, pelo companheirismo incondicional.

Resumo

Este trabalho apresenta o planejamento e as etapas iniciais do desenvolvimento de um novo sistema para a gestão dos Trabalhos de Conclusão de Curso (TCCs) dos cursos de graduação do Departamento de Ciência da Computação (DCC) da Universidade Federal de Juiz de Fora (UFJF). O novo sistema tem como objetivo substituir o *tcc-web*, sistema legado desenvolvido há mais de uma década, que apresenta sérias limitações decorrentes do uso de tecnologias obsoletas, ausência de documentação e falta de padronização no código. Para o planejamento do novo sistema, foram conduzidas reuniões com os principais usuários para levantamento de requisitos, seguidas da modelagem da arquitetura, do banco de dados, das interfaces do usuário e da escolha do conjunto tecnológico. O projeto adota tecnologias modernas e ferramentas de integração contínua, com foco em escalabilidade, segurança e facilidade de manutenção. Este trabalho também descreve o planejamento das atividades, o fluxo de trabalho adotado para o projeto e as primeiras implementações realizadas. Como trabalhos futuros, estão previstos o desenvolvimento das funcionalidades restantes, a configuração de ferramentas de monitoramento e automação, e a migração dos dados do sistema atual para a nova aplicação.

Palavras-chave: Sistemas legados, engenharia de software, modelagem de software, trabalhos de conclusão de curso, modernização de sistemas.

Abstract

This work presents the planning and initial stages of the development of a new system for managing the undergraduate thesis (TCCs) of the Computer Science Department (DCC) at the Federal University of Juiz de Fora (UFJF). The new system aims to replace *tcc-web*, a legacy system developed over a decade ago, which faces serious limitations due to the use of outdated technologies, lack of documentation, and absence of standardized coding practices. For the planning of the new system, meetings were held with key users to specify the system requirements, followed by the modeling of the system architecture, database, user interfaces, and selection of the technology stack. The project adopts modern technologies and continuous integration tools, focusing on scalability, security, and maintainability. This work also describes the activity planning, the workflow adopted for the project, and the initial implementations carried out. Future work includes the development of the remaining features, configuration of monitoring and automation tools, and migration of data from the current system to the new application.

Keywords: Legacy systems, software engineering, software documentation, undergraduate thesis, system modernization, software architecture, software requirements.

Agradecimentos

À Deus, pela sabedoria e a determinação para concluir esta etapa da minha vida.

Aos meus pais, por todo o apoio durante essa árdua jornada, todos os conselhos e ensinamentos de vida que me fizeram chegar até aqui.

Aos meus irmãos, por mesmo de longe sempre estarem presentes e por sempre acreditarem em mim.

À todos os amigos e familiares que, antes mesmo do início dessa jornada, me incentivaram a sair de Espera Feliz e ir atrás dos meus sonhos.

Aos amigos que fiz durante toda essa jornada, que permitiram que essa jornada em uma nova cidade fosse mais leve e tranquila.

Ao professor Luciano pela orientação, amizade e principalmente, pela paciência, sem a qual este trabalho não se realizaria.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

À minha esposa, que antes de todos acreditou no meu potencial e se dispôs a correr atrás desse sonho lado a lado comigo, por todo o apoio incondicional e paciência durante todos esses anos.

“Eu não luto achando que vou vencer, eu luto porque tenho que vencer, é por isso que continuo lutando!”.

Tite Kubo (Kurosaki Ichigo - Bleach)

Sumário

Lista de Figuras	7
Lista de Abreviações	8
1 Introdução	9
2 O Sistema Legado	12
2.1 Descrição do sistema	12
2.2 Problemas e desafios	14
3 Modelagem do Novo Sistema	16
3.1 Requisitos do sistema	16
3.1.1 Levantamento e análise de requisitos	16
3.1.2 Especificação de requisitos	19
3.2 Modelagem do sistema	20
3.2.1 Arquitetura do sistema	20
3.2.2 Fluxo do processo	21
3.2.3 Interfaces do usuário	23
3.2.4 Banco de dados	27
3.3 Planejamento tecnológico	30
3.3.1 Linguagem de programação	30
3.3.2 Frontend	30
3.3.3 Backend	31
3.3.4 Banco de dados	32
3.3.5 Controle de versão	33
3.3.6 Infraestrutura	34
4 Desenvolvimento Inicial do Novo Sistema	36
4.1 Gestão de atividades	36
4.2 Implementações em andamento	38
5 Considerações Finais	42
A Especificação dos Requisitos	43
A.1 Requisitos Funcionais (RF)	43
A.2 Requisitos Não Funcionais (RNF)	66
Referências Bibliográficas	69

Lista de Figuras

2.1	Tela de listagem pública dos trabalhos no sistema <i>tcc-web</i>	13
2.2	Tela de detalhes de um trabalho no sistema <i>tcc-web</i>	13
2.3	Tela de autenticação no sistema <i>tcc-web</i>	14
3.1	Arquitetura do novo sistema.	21
3.2	Fluxo do processo de desenvolvimento do projeto no TCC I.	24
3.3	Fluxo do processo de desenvolvimento do trabalho no TCC II.	25
3.4	Protótipo da nova tela de listagem pública dos TCCs.	26
3.5	Protótipo da nova tela de detalhes de um TCC.	27
3.6	Modelo do banco de dados do sistema <i>tcc-web</i>	28
3.7	Modelo do banco de dados do novo sistema.	29
4.1	Quadro <i>Kanban</i> do projeto no <i>Jira</i>	37
4.2	Fluxo de trabalho do projeto no <i>Jira</i>	38
4.3	Tela inicial do novo sistema na versão para computador.	39
4.4	Tela inicial do novo sistema na versão para dispositivos móveis.	39
4.5	História do usuário para autenticação no novo sistema.	40

Lista de Abreviações

API	<i>Application Programming Interface</i>
CD	<i>Continuous Delivery</i>
CI	<i>Continuous Integration</i>
CPF	Cadastro de Pessoa Física
DCC	Departamento de Ciência da Computação
HTTP	<i>Hypertext Transfer Protocol</i>
ICE	Instituto de Ciências Exatas
JS	<i>JavaScript</i>
NPM	<i>Node Package Manager</i>
NRC	Núcleo de Recursos Computacionais
ORM	<i>Object-Relational Mapping</i>
PDF	<i>Portable Document Format</i>
RF	Requisito Funcional
RNF	Requisito Não Funcional
SIGA	Sistema Integrado de Gestão Acadêmica
TCC	Trabalho de Conclusão de Curso
TS	<i>TypeScript</i>
UFJF	Universidade Federal de Juiz de Fora

1 Introdução

Os softwares desempenham um papel cada vez mais notável em nosso cotidiano, tornando-se elementos essenciais para o funcionamento eficiente de organizações e instituições, além de facilitarem diversas atividades do dia a dia das pessoas, contribuindo para a produtividade, comunicação e o acesso à informação.

No âmbito da gestão acadêmica, o Departamento de Ciência da Computação (DCC) da Universidade Federal de Juiz de Fora (UFJF) possui hoje um sistema para realizar o acompanhamento do desenvolvimento e a disponibilização pública dos Trabalhos de Conclusão de Curso (TCCs) elaborados pelos discentes ao final de suas graduações. O sistema tem como objetivo auxiliar estudantes dos cursos de Sistemas de Informação e Ciência da Computação do DCC no fluxo entre as etapas de desenvolvimento de seus TCCs, desde o projeto até a aprovação do trabalho final. Para isso, o sistema viabiliza o armazenamento e a troca de informações entre estudantes, coordenadores de curso e professores orientadores. O sistema pode ser acessado no endereço <http://www.monografias.ice.ufjf.br/tcc-web>.

Sistemas de apoio à gestão acadêmica, como o descrito, desempenham um papel relevante na organização e no acompanhamento de atividades institucionais. No entanto, observa-se uma escassez de estudos na literatura científica que tratem especificamente do desenvolvimento, uso ou impacto desses sistemas em contextos universitários. Um exemplo é o trabalho de Franco e Guzman (2016), que propôs e implementou um sistema web para coordenação de TCCs no Departamento de Tecnologia da Computação de uma universidade nas Filipinas. Outro exemplo é o trabalho de Xiaoping, Fanqi e Yan (2017), que descreve o desenvolvimento de um sistema voltado à organização e divulgação dos temas de TCC disponíveis, com o objetivo de facilitar a interação entre alunos e professores no processo de escolha de orientadores e assuntos a serem desenvolvidos. Ambos os trabalhos reforçam a importância de soluções tecnológicas para aprimorar a gestão acadêmica dos TCCs, ainda que com enfoques distintos sobre as etapas do processo.

Atualmente, o gerenciamento e a manutenção do sistema de gestão de TCCs

do DCC estão sob responsabilidade das coordenações dos cursos de graduação que o utilizam. Os coordenadores dos cursos são responsáveis pelo planejamento das atividades de manutenção e melhoria, contando, para isso, com o apoio de estudantes vinculados a projetos de treinamento profissional custeados pela UFJF. Cabe a esses estudantes, de fato, a implementação das alterações no sistema, sob a supervisão dos coordenadores de curso.

Com o passar dos anos, o sistema evoluiu de forma desorganizada e, hoje, apresenta uma série de limitações. Desenvolvido há mais de uma década, carece de um padrão claro de codificação, possui documentação inadequada e utiliza tecnologias desatualizadas. Como consequência, a resolução de problemas e a implementação de novas funcionalidades tornaram-se tarefas complexas e demoradas, comprometendo sua capacidade de atender às necessidades atuais dos cursos.

Esse cenário é característico de um sistema legado — termo utilizado para descrever sistemas que, embora tenham desempenhado papel importante no passado, passaram a representar um entrave à modernização tecnológica. Tais sistemas, ainda presentes em muitas organizações, distinguem-se pelo uso de tecnologias obsoletas, dificuldades de manutenção e documentação escassa (FRITOLA; SANTANDER, 2021). Embora tenham trazido benefícios inicialmente, acabam exigindo grande esforço e recursos para serem mantidos e adaptados às demandas atuais.

Neste contexto, este trabalho descreve o planejamento e as etapas iniciais do desenvolvimento de um novo sistema para o acompanhamento de TCCs no DCC, com o propósito de substituir o sistema atual. Até o momento, o projeto conta com os requisitos, a arquitetura e a estrutura inicial de desenvolvimento definidos, e encontra-se na fase de implementação das primeiras funcionalidades. Para atingir o objetivo proposto, foram adotadas rigorosamente as melhores práticas da engenharia de software, de modo a assegurar que o novo sistema, desde a concepção até o seu ciclo de vida contínuo, seja desenvolvido de forma a evitar que se torne obsoleto com o passar dos anos (SOMMERVILLE, 2019).

Para o planejamento do novo sistema, foi conduzido um levantamento de requisitos através de reuniões semanais com os coordenadores dos cursos de graduação, com

o objetivo de compreender as necessidades e expectativas sobre o novo sistema. A partir disso, foram identificados os requisitos funcionais e não funcionais do sistema, estabelecendo uma base sólida para o projeto. Em seguida, foi realizada toda a modelagem do novo sistema, como a definição da arquitetura, a modelagem do banco de dados, a modelagem das interfaces do usuário voltadas à experiência do usuário e a escolha das melhores tecnologias para o desenvolvimento do software. Por fim, o planejamento e a distribuição das tarefas foram organizados utilizando um quadro *Kanban* com auxílio da ferramenta *Jira*, permitindo que a implementação inicial do novo sistema fosse realizada de forma prática e eficiente, servindo como base para as próximas fases de desenvolvimento.

O restante deste documento está organizado da seguinte forma, o Capítulo 2 descreve o sistema atualmente em uso, o Capítulo 3 apresenta o planejamento e a modelagem do novo sistema, o Capítulo 4 fala sobre as etapas iniciais do desenvolvimento do novo sistema e, por fim, o Capítulo 5 traz as considerações finais e os trabalhos futuros.

2 O Sistema Legado

Este capítulo apresenta uma visão geral do sistema atualmente em uso. Na Seção 2.1 são descritas as principais características do sistema, bem como sua importância para os seus usuários e na Seção 2.2 são apresentados os problemas e desafios enfrentados pela equipe de desenvolvimento do sistema.

2.1 Descrição do sistema

O sistema atual de gestão de TCCs do DCC da UFJF, conhecido pela equipe de desenvolvedores como *tcc-web*, é uma aplicação web que tem como objetivo auxiliar orientadores e coordenadores de curso no acompanhamento do fluxo de desenvolvimento dos trabalhos de conclusão de curso dos graduandos do departamento. O sistema abrange todo o ciclo de vida dos TCCs, desde a criação do projeto até a entrega da versão final juntamente com a ata da defesa, permitindo que os envolvidos interajam com as informações pertinentes aos trabalhos.

O desenvolvimento do *tcc-web* foi iniciado em 2011 e, desde então, mais de 20 estudantes e professores do DCC contribuíram com sua construção e manutenção. A principal linguagem de programação utilizada no sistema é o *Java* versão 11 (JAVA, 2025). O *frontend* da aplicação é construído com o *framework* *ZK* versão 7.0.0 (ZK, 2025), um *framework* de código aberto em *Java* para auxiliar no desenvolvimento de interfaces para aplicações web interativas. O banco de dados utilizado é o *MySQL Server* versão 5.5.54 (MYSQL, 2025).

Para ilustrar a interface atual do sistema, a Figura 2.1 apresenta a tela inicial do sistema, acessível sem necessidade de autenticação, que exibe a lista pública dos TCCs já concluídos e aprovados. A Figura 2.2 mostra a tela de detalhes de um TCC, onde é possível visualizar mais informações sobre um trabalho em específico, como o título, autor, orientador, resumo, palavras-chave, bem como uma visualização do trabalho completo em Formato Portátil de Documento (do inglês, *Portable Document Format* – PDF). Por fim,

a Figura 2.3 mostra a tela para realizar a autenticação (*login*) dos usuários no sistema. A autenticação acontece por meio de uma Interface de Programação de Aplicações (do inglês, *Application Programming Interface* – API) conectada ao Sistema Integrado de Gestão Acadêmica (SIGA) da UFJF. A identificação do usuário é através de seu número de Cadastro de Pessoa Física (CPF) e a senha é a mesma utilizada para acessar o SIGA.

The image shows a web interface for the Universidade Federal de Juiz de Fora. At the top, there's a header with the university name, a link to 'Validar Certificado Digital', and a button 'Entrar no sistema'. Below this is a section titled 'Lista Pública de Trabalhos Acadêmicos'. It features a filter for 'Curso' (set to 'Todos (trabalhos mais recentes)') and 'Ano' (set to 'Todos'), a search bar, and a 'Filtrar' button. A table displays a list of academic works. Each row includes the title of the work, the author's name, a PDF icon, and an 'Extras' icon. An 'Entrar no sistema' modal is open over the table, showing fields for 'Identificador' and 'Senha', and an 'Entrar' button.

Trabalho	Autor	PDF	Extras
Simulação e Controle Preditivo de um Ciclista Virtual	VÍTOR AGOSTINHO MOURO		
Zero-Trust e Controle de Acesso Dinâmico Aplicados na Segurança de Arquivos	LUCCA OLIVEIRA SCHRÖDER		
Geração Procedural de Mapas por Cadeias	RIANO EITERER VEIRA		
"Build-block Architecture" – Uma Abordagem para iOS	LIPE ISRAEL DE VEIRA VIDAL		
Método de Diferenças Finitas ADI Combinado de Navier-Stokes	PHAEEL NOGUEIRA ZENDE LAROCA PINTO		
Estimador de Sombras em Realidade Aumentada	DRÉ LUÍS BRAGA COSTA		
Computação Desplugada para Apoio ao Ensino da Computação em Escolas Públicas	LAÍS FIGUEIREDO LINHARES		
Uso de Arduino no Ensino de Algoritmos para Autistas	MARCELO GONÇALVES DE SOUZA COSTA		
Desenvolvimento de uma Extensão de Suporte para Especialistas em ServiceNow	MARCUS VINÍCIUS RODRIGUES DA SILVA		
Identificação de Habilidades no Código-Fonte	MARA DE LEMOS GOMES		
Uma análise exploratória da influência de bad smells e refatorações no volume de comentários	KLEITON EWERTON DE OLIVEIRA		

Figura 2.3: Tela de autenticação no sistema *tcc-web*.

2.2 Problemas e desafios

Apesar de o sistema *tcc-web* existir há mais de uma década, ele não recebeu adequadamente as atualizações tecnológicas durante o seu ciclo de vida. O fato de o sistema não adotar um gerenciamento de dependências com ferramentas automatizadas, como o *Maven* (MAVEN, 2025), contribuiu para a falta de modularidade e escalabilidade do mesmo.

No que diz respeito ao *framework* *ZK*, utilizado no *frontend* da aplicação, a versão atualmente em uso pelo sistema (versão 7.0.0, de novembro de 2013) encontra-se

desatualizada. A versão atual do *framework* é a 10.2.1, de julho de 2025. Além disso, o *ZK* apresenta algumas limitações em termos de popularidade e suporte da comunidade.

Por exemplo, enquanto o *Vaadin Flow* (VAADIN, 2025), que é um outro *framework* para *frontend*, possui 701 estrelas no GitHub¹, contra 456 do *ZK*², e conta com 195 colaboradores ativos, frente a apenas 49 no *ZK*. Essa diferença se reflete diretamente no dia a dia de desenvolvimento, pois a menor base de colaboradores, a baixa frequência de atualizações e a documentação mais limitada do *ZK* aumentam a curva de aprendizado para novos colaboradores e dificultam a manutenção do *frontend*.

Outro aspecto crítico é a versão do banco de dados utilizado no sistema (versão 5.5.54, de dezembro de 2016) que já não possui mais suporte oficial. Essa versão desatualizada do banco de dados apresenta vulnerabilidades conhecidas, o que representa um risco significativo para a segurança e integridade dos dados armazenados (SECURITY, 2024). Além disso, a falta de atualizações e suporte torna o sistema suscetível a falhas e limitações que podem comprometer sua funcionalidade.

Além dos aspectos técnicos, o projeto encontra dificuldades relacionadas ao gerenciamento de seu processo de desenvolvimento. O sistema é essencialmente mantido por estudantes dos cursos de Ciência da Computação e Sistemas de Informação do DCC da UFJF, proporcionando oportunidades de aprendizado prático aos envolvidos. Entretanto, muitos desses estudantes possuem apenas conhecimentos básicos sobre as tecnologias, com pouca ou nenhuma experiência prévia no desenvolvimento de sistemas web complexos com as tecnologias adotadas. Essa situação, associada à falta de documentação adequada e à ausência de uma padronização da estrutura do código-fonte do *tcc-web*, aumenta consideravelmente o tempo necessário para que os novos colaboradores possam efetivamente contribuir com o projeto.

Esses fatores combinados têm resultado em um custo maior em termos de tempo e recursos para correções de erros e na implementação de novas funcionalidades no sistema *tcc-web*, caracterizando-o como um sistema legado. Dessa forma, o desenvolvimento de um novo sistema para substituir o atual, utilizando tecnologias atuais e seguindo princípios da engenharia de software, surgiu como uma iniciativa estratégica e necessária.

¹<<https://github.com/vaadin/flow>>

²<<https://github.com/zkoss/zk>>

3 Modelagem do Novo Sistema

Este capítulo apresenta a modelagem do novo sistema, visando garantir um processo de desenvolvimento mais estruturado, eficiente e sustentável. A aplicação desses fundamentos se mostra essencial no desenvolvimento de sistemas confiáveis e de fácil manutenção, evidenciando a importância da engenharia de software. Segundo Sommerville (2019), essa importância se deve principalmente a dois fatores: o aumento da necessidade da sociedade por softwares mais complexos e o fato de que, a longo prazo, é mais econômico manter e evoluir um software que seguiu um processo de engenharia de software adequado.

A Seção 3.2 descreve o levantamento, análise e especificação dos requisitos para o novo sistema. A Seção 3.2 apresenta a modelagem do sistema, incluindo a arquitetura, o fluxo do processo, as interfaces gráficas do usuário e o modelo de dados. Por fim, a Seção 3.3 apresenta o planejamento tecnológico para o desenvolvimento e a implantação do novo sistema.

3.1 Requisitos do sistema

Esta seção apresenta o processo de levantamento, análise e especificação dos requisitos do novo sistema. Na Subseção 3.1.1 será descrito o processo de levantamento e análise, enquanto na Subseção 3.1.2 será apresentada a forma adotada para a especificação final dos requisitos do sistema.

3.1.1 Levantamento e análise de requisitos

Para o projeto e desenvolvimento do software proposto, foi imprescindível levantar os requisitos necessários à sua construção. Essa etapa foi fundamental para compreender as necessidades dos usuários finais, estabelecer as funcionalidades essenciais do sistema e garantir que a solução atenda de maneira integral às expectativas e demandas previstas.

Entrevista

Como parte fundamental do processo de levantamento de requisitos para a especificação do sistema proposto, durante o primeiro semestre de 2024, foram conduzidas entrevistas semanais com os professores Gleiph Ghiotto, Leonardo Vieira e Luciano Chaves. Estes professores são os atuais coordenadores dos cursos de graduação em Ciência da Computação (integral), Ciência da Computação (noturno) e Sistemas de Informação, respectivamente. Essa iniciativa teve como objetivo obter todas as informações necessárias sobre as expectativas e funcionalidades desejadas para o sistema. Durante as entrevistas, foram esclarecidos os objetivos do sistema:

- Sistematizar e automatizar a troca de informações entre orientandos, orientadores, coordenadores de curso e membros de bancas de avaliação ao longo do projeto e do desenvolvimento dos TCCs.
- Servir de agenda pública para as apresentações de TCCs agendadas.
- Servir de repositório público para os TCCs finalizados.

Identificação dos requisitos

Já com os dados resultantes das entrevistas, foi realizado um levantamento abrangente de todos os requisitos do sistema, englobando tanto os Requisitos Funcionais (RFs) quanto os Requisitos Não Funcionais (RNFs) (SOMMERVILLE, 2019). Os RFs descrevem as funcionalidades do sistema, abrangendo as interações entre os usuários (estudantes, orientadores e coordenadores) e as principais funcionalidades do sistema, como a criação de projetos de TCC, agendamento de defesas e a disponibilização pública dos trabalhos concluídos. Já os RNFs tratam de aspectos como desempenho, segurança, usabilidade e manutenibilidade do sistema, garantindo que ele atenda a padrões de qualidade e eficiência.

Essa fase foi essencial para garantir a qualidade do software que será desenvolvido, proporcionando uma compreensão detalhada das expectativas dos usuários e dos parâmetros que irão garantir o funcionamento do sistema.

Análise dos requisitos identificados

Em seguida, foi realizada uma análise dos requisitos identificados durante o levantamento, visando garantir a consistência e clareza dos requisitos, identificando possíveis conflitos entre os requisitos, duplicidades e a semelhança entre as funcionalidades. Além disso, requisitos considerados redundantes, irrelevantes ou que não agregassem valor significativo foram desconsiderados.

Ao final dessa análise, obteve-se uma compreensão mais refinada e precisa dos requisitos do sistema, estabelecendo uma base sólida para as fases subsequentes do desenvolvimento do software.

Avaliação dos requisitos

Por fim, foi realizada uma avaliação dos requisitos identificados para o sistema, conduzida em conjunto com os principais *stakeholders*, a fim de garantir que todos os interesses e necessidades fossem devidamente contemplados.

Durante essa etapa, foram analisados diversos aspectos, como a viabilidade técnica e operacional de cada requisito, os custos e esforços envolvidos na implementação, o nível de prioridade em relação aos objetivos estratégicos do projeto e o impacto direto na qualidade do software a ser desenvolvido. Também foram discutidas possíveis restrições e dependências entre requisitos, visando identificar riscos e antecipar eventuais problemas que poderiam comprometer o desenvolvimento do novo sistema.

Essa avaliação aprofundada foi fundamental para assegurar que os requisitos definidos estivessem alinhados com a visão do projeto, com as expectativas dos usuários e com os recursos disponíveis para a implementação do sistema.

Como resultado, estabeleceu-se uma base sólida e validada para as próximas fases do desenvolvimento, reduzindo riscos, evitando retrabalho e contribuindo para a entrega de um software de maior qualidade e com maior aderência às necessidades do departamento.

3.1.2 Especificação de requisitos

Os requisitos foram documentados de forma clara, precisa e objetiva. Sua especificação foi estruturada em requisitos do usuário que, posteriormente, foram subespecificados em requisitos do sistema, de forma a detalhar seu comportamento e funcionalidades esperadas. Abaixo, são apresentados alguns exemplos de requisitos funcionais e não funcionais que foram especificados para o novo sistema:

- **RF01:** O visitante pode visualizar, na página inicial do sistema, uma lista pública com todos os TCCs concluídos e aprovados.
 - **RF01.1:** O sistema deve exibir, para cada TCC da lista, o título do trabalho, o nome do autor, o ano da defesa e ícones que permitam o acesso direto ao documento e à página de detalhes do TCC.
 - **RF01.2:** Ao clicar sobre o ícone do documento em um trabalho na lista, o sistema deve exibir ao visitante o documento em PDF.
 - **RF01.3:** Ao clicar sobre o ícone de detalhes de um trabalho na lista, o sistema deve redirecionar o visitante para a página com os detalhes do trabalho.
- **RF24:** Ao selecionar um perfil de aluno, o usuário deve ver em sua página inicial a linha do tempo com todas as etapas relativas ao desenvolvimento do TCC, destacando a etapa em que se encontra no momento.
 - **RF24.1:** O sistema deve exibir, para cada etapa da linha do tempo, um texto descritivo que oriente o aluno sobre o que acontece na etapa, bem como destacar quem é o ator responsável pelas ações da etapa (aluno, coordenador ou orientador).
 - **RF24.2:** O sistema deve destacar a etapa em que TCC do aluno se encontra no momento, diferenciando visualmente as etapas já concluídas da etapa atual e das etapas futuras.
 - **RF24.3:** Caso existam ações a serem executadas pelo aluno na etapa atual de seu TCC, o sistema deve exibir botões de atalho para a execução da ação

(ex. botão de solicitar matrícula em TCC I para o aluno que está na primeira etapa).

- **RF24.4:** Caso o TCC já esteja concluído e disponível na lista pública, o sistema deve exibir um botão que direciona o aluno à página pública de detalhes de seu TCC.
- **RNF04:** O sistema deve garantir a proteção das credenciais e das informações pessoais de seus usuários.
 - **RNF04.1:** O sistema deverá utilizar o *Redis* para armazenar e proteger informações sigilosas de acessos não autorizados.

A especificação completa de todos os requisitos levantados para o novo sistema está documentada e disponível em <<http://bit.ly/4oevcuB>>.

3.2 Modelagem do sistema

Esta seção apresenta os principais aspectos da modelagem do novo sistema. A Subseção 3.2.1 apresenta a organização arquitetural do sistema; a Subseção 3.2.2 apresenta o modelo de fluxo do processo de desenvolvimento dos TCCs, com os principais atores e as etapas do processo; a Subseção 3.2.3 apresenta o protótipo de interface do usuário do sistema; e, por fim, a Subseção 3.2.4 detalha o modelo proposto para o banco de dados, com suas entidades e relacionamentos.

3.2.1 Arquitetura do sistema

O projeto de arquitetura de software é uma etapa essencial, pois define como o sistema será organizado e conecta o projeto aos seus requisitos. Nessa fase, são estabelecidos os principais componentes estruturais e seus relacionamentos, determinando a estrutura do sistema e a forma como esses componentes irão interagir (SOMMERVILLE, 2019).

A Figura 3.1 apresenta a arquitetura em camadas do novo sistema, que separa as responsabilidades para facilitar a organização e a manutenção do sistema ao longo de seu ciclo de vida. Dessa forma, foram definidas três camadas:

- **Camada de aplicação**, que é a camada com quem o usuário interage diretamente;
- **Camada do servidor**, que é responsável por processar e responder as requisições feitas pela camada de aplicação;
- **Camada de persistência**, que é responsável por armazenar os dados do sistema e fornecer acesso a esses dados para a camada do servidor.

Na figura também é possível observar as tecnologias adotadas em cada camada do sistema, cujos detalhes serão apresentados na Seção 3.3.

3.2.2 Fluxo do processo

Atualmente, o desenvolvimento de um TCC por discentes dos cursos de graduação do DCC da UFJF ocorre ao longo de duas disciplinas: TCC I e TCC II. Dessa forma, o fluxo de desenvolvimento dos TCCs é dividido em duas fases: projeto e trabalho, respectivamente.

Durante o desenvolvimento de um TCC, três atores principais interagem com o sistema: o aluno, o orientador e o coordenador de curso. As responsabilidades de cada ator no processo de desenvolvimento dos TCCs, em ordem cronológica, são:

- **Aluno**
 1. Solicitar matrícula na disciplina de TCC I;
 2. Desenvolver o projeto;

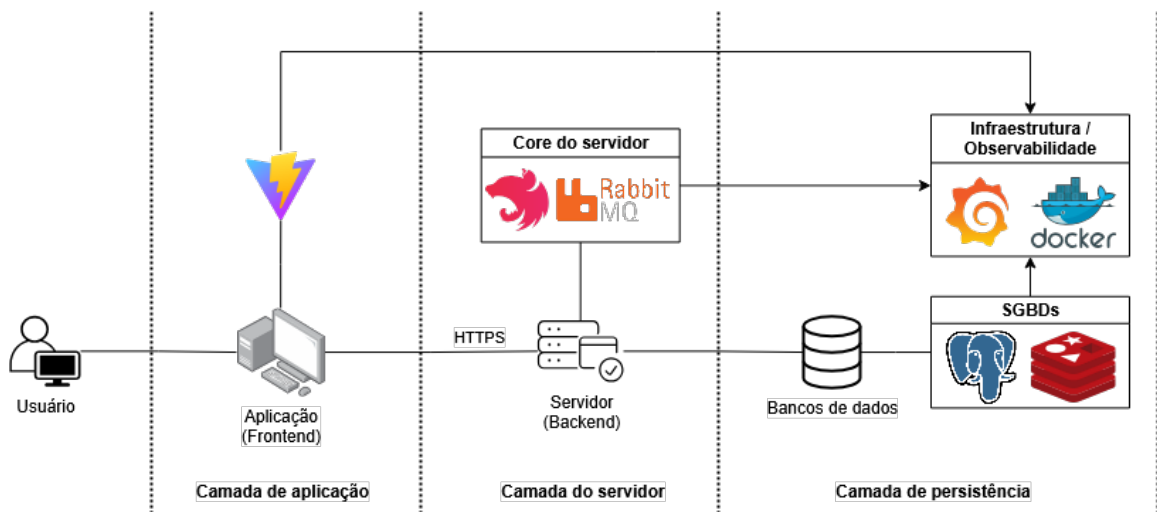


Figura 3.1: Arquitetura do novo sistema.

3. Enviar o projeto para revisão do orientador;
4. Apresentar o projeto;
5. Solicitar matrícula na disciplina de TCC II;
6. Desenvolver o trabalho;
7. Enviar o trabalho para revisão do orientador;
8. Agendar defesa do trabalho;
9. Apresentar o trabalho;
10. Proceder com as correções na versão final do trabalho;
11. Enviar a versão final do trabalho;

- **Professor orientador**

1. Responder à solicitação de orientação de projeto;
2. Autorizar a apresentação do projeto;
3. Revisar o trabalho;
4. Autorizar a apresentação do trabalho;
5. Informar a deliberação da banca examinadora;
6. Aprovar o documento final do trabalho;

- **Coordenador de curso**

1. Matricular o aluno na disciplina de TCC I;
2. Agendar apresentação do projeto;
3. Lançar conceito do projeto no SIGA;
4. Matricular o aluno na disciplina de TCC II;
5. Verificar ata de defesa e formatação do documento final;
6. Lançar conceito do trabalho no SIGA;

Para uma melhor compreensão do fluxo extenso de desenvolvimento dos TCCs e suas etapas, foram elaborados dois diagramas de atividades, um para cada fase do

processo. A Figura 3.2 apresenta o fluxo do processo de desenvolvimento do projeto no TCC I, que engloba as etapas de criação da solicitação de orientação e matrícula pelo aluno, aceitação de orientação pelo orientador, registro da matrícula no SIGA pela coordenação, desenvolvimento do projeto pelo aluno, aprovação do projeto pelo orientador, agendamento da apresentação pela coordenação, apresentação do projeto pelo aluno e lançamento do conceito final no SIGA pela coordenação.

A Figura 3.3 apresenta o fluxo do processo de desenvolvimento do trabalho no TCC II, que inclui a solicitação de orientação e matrícula pelo aluno, aceitação de orientação pelo orientador, registro da matrícula no SIGA pela coordenação, o desenvolvimento do trabalho pelo aluno, o agendamento da defesa pelo aluno, a autorização da defesa pelo orientador, a defesa do trabalho pelo aluno, o registro do resultado da defesa pelo orientador, o envio da versão final do trabalho pelo aluno, a aprovação da versão final pelo orientador, a verificação da ata e da formatação pela coordenação e o lançamento do conceito final no SIGA pela coordenação.

3.2.3 Interfaces do usuário

Protótipos visuais de alta fidelidade são representações detalhadas da interface final de um sistema, incluindo *layout*, tipografia, cores e elementos interativos. Eles desempenham papel estratégico no processo de desenvolvimento, pois funcionam como documentação visual clara e como uma prévia realista da experiência do usuário (KRUG, 2014). Além de reduzir ambiguidades e retrabalho, permitem validar decisões de design antes da implementação, tornando o desenvolvimento mais ágil e assertivo.

Para o desenvolvimento do projeto da interface do novo sistema, foi utilizado o *Figma*, uma ferramenta gratuita de *design* colaborativo amplamente adotada na indústria de software (FIGMA, 2025). Com o *Figma* foi possível criar protótipos de alta fidelidade, visualizar e interagir com a interface gráfica do sistema antes mesmo do início do desenvolvimento.

Os protótipos de interface do usuário foram desenvolvidos com base na interface do *tcc-web* e nos requisitos especificados para o novo sistema. O objetivo foi a criação de uma interface moderna, intuitiva e responsiva, corrigindo os problemas da interface

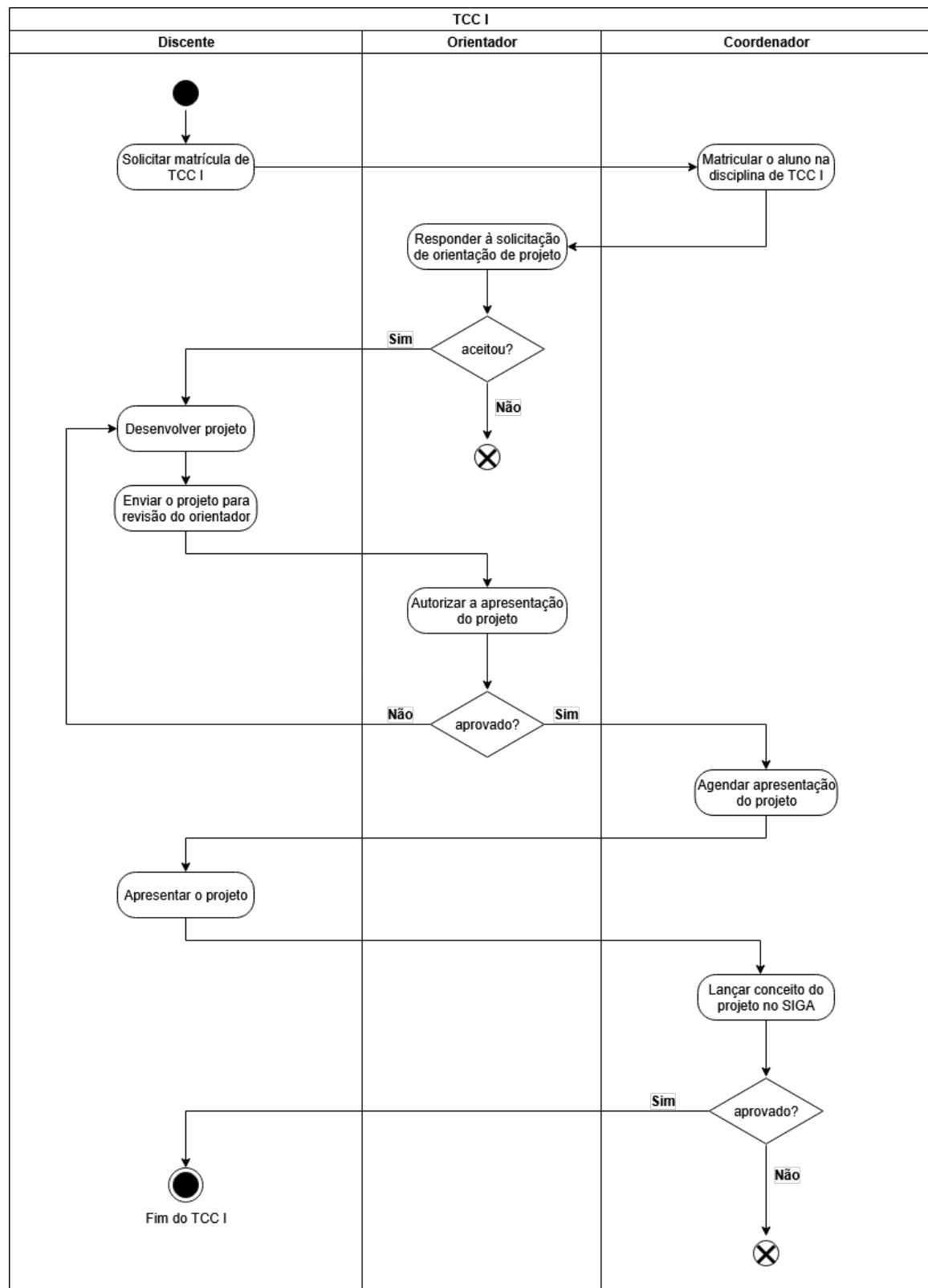


Figura 3.2: Fluxo do processo de desenvolvimento do projeto no TCC I.

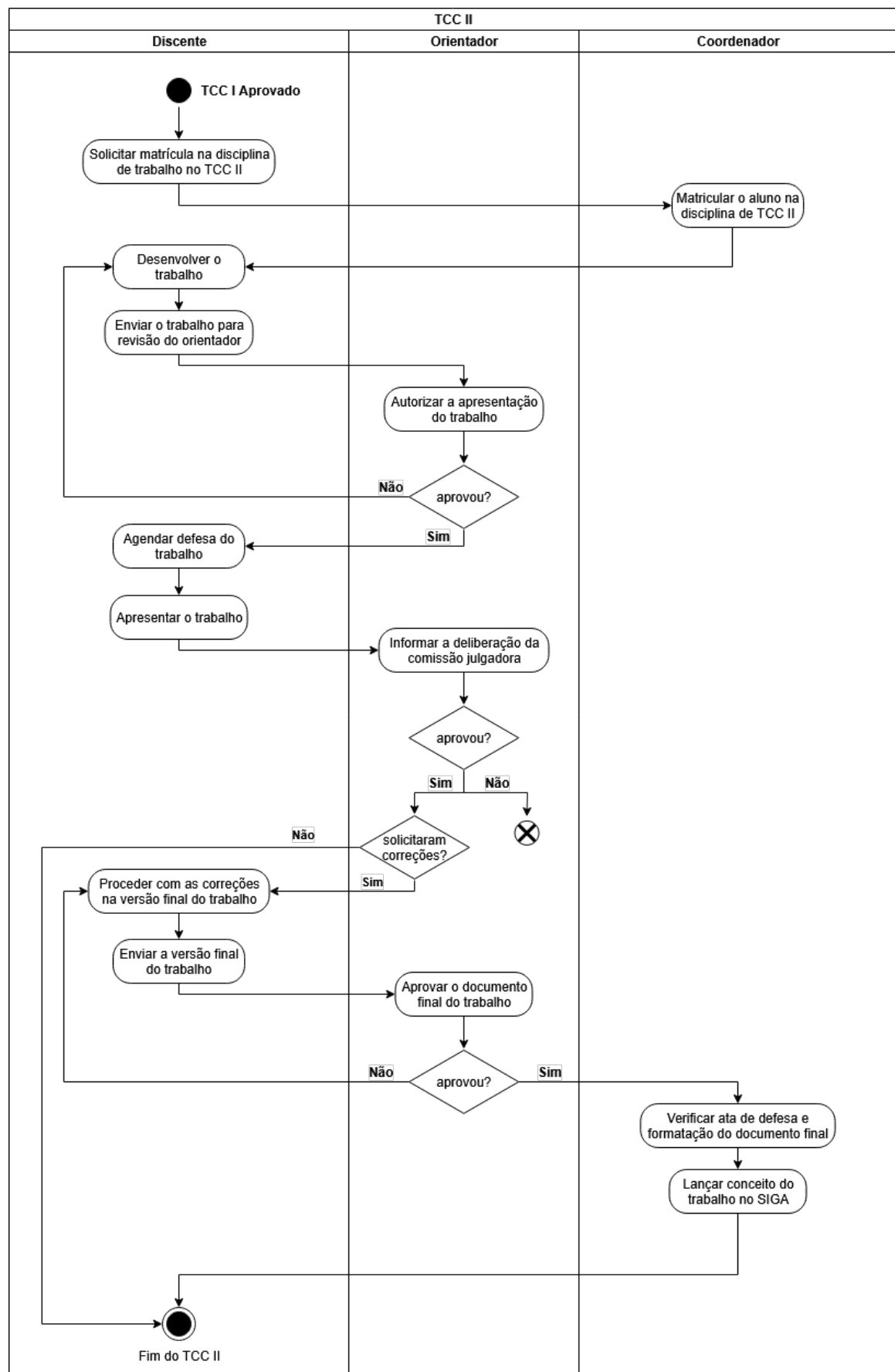


Figura 3.3: Fluxo do processo de desenvolvimento do trabalho no TCC II.

antiga e permitindo aos usuários interagir com o novo sistema de forma agradável.

A Figura 3.4 apresenta o protótipo da tela de listagem pública dos TCCs. Em comparação com a tela apresentada na Figura 2.1, o novo protótipo apresenta a listagem pública de TCCs e permite também a visualização das defesas de trabalhos agendadas. A identidade visual do novo sistema tem uma paleta de cores alinhada com as cores da UFJF, com destaque para as ações disponíveis no sistema, como a visualização detalhada de um trabalho, o *download* do trabalho completo em PDF e a possibilidade da expansão das listas de defesas agendadas e da listagem de TCCs.

A Figura 3.5 apresenta o protótipo da tela de detalhes de um TCC. Em comparação com a tela apresentada na Figura 2.2, o novo protótipo traz mais destaque para as informações gerais do TCC, deixando claro para o usuário a visão geral do trabalho sem remover a pré-visualização do trabalho completo em PDF, que também é importante. A tela também corrige problemas da versão atual, como a ausência de um botão que permita ao usuário voltar para a tela de listagem pública dos TCCs.



Figura 3.4: Protótipo da nova tela de listagem pública dos TCCs.



Figura 3.5: Protótipo da nova tela de detalhes de um TCC.

3.2.4 Banco de dados

Embora um novo sistema esteja sendo desenvolvido, é fundamental considerar o modelo de banco de dados atual para garantir uma migração eficiente e sem perda de informações. O sistema *tcc-web* utiliza um modelo entidade-relacionamento que foi criado de forma desorganizada, o que impõe desafios adicionais. Além dos problemas já apresentados no Capítulo 2, essas limitações da modelagem anterior também impactam diretamente o processo de transição para a nova versão.

A Figura 3.6, gerada pela ferramenta *Dbeaver* de gerenciamento de banco de dados (DBEAVER, 2025), apresenta o modelo do banco de dados atual do sistema *tcc-web*. A partir dessa figura, podemos ilustrar alguns dos problemas do modelo atual:

- Os atributos `emailAlertaEnviados`, `idSala`, `salaDefesa` e `linkSala` não deveriam estar presentes na tabela `tcc`, pois violam a 2ª forma normal. Esses atributos deveriam estar em outra(s) tabela(s), relacionada à tabela `tcc` por meio de uma chave estrangeira.
- As tabelas `tcc` e `usuario` possuem o atributo `idOrientador` em comum. Isso representa redundância de dados, pois o orientador de um TCC é um usuário do

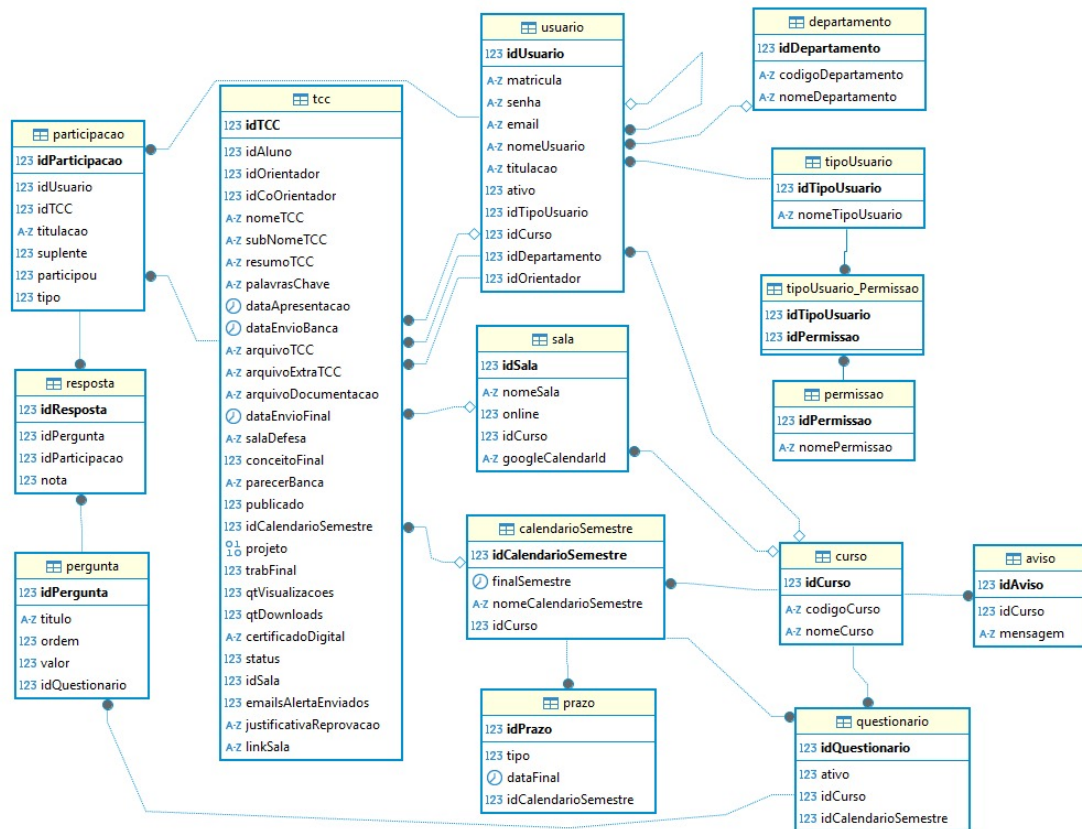


Figura 3.6: Modelo do banco de dados do sistema *tcc-web*.

sistema e, portanto, já possui um registro na tabela **usuario**. O ideal seria que a tabela **tcc** armazenasse apenas o identificador do orientador, referenciando o registro correspondente na tabela **usuario**.

- As tabelas **participacao**, **resposta**, **pergunta** e **questionario** relacionam a defesa de um TCC com os professores que participaram da banca, as perguntas feitas nos questionários de avaliação e a nota de cada resposta. No entanto, a forma como estão estruturadas não reflete adequadamente essa relação, resultando em uma modelagem de difícil compreensão.

A construção do novo modelo de banco de dados precisou se basear no modelo atualmente em uso, uma vez que a migração dos dados do sistema antigo para o novo será uma etapa crucial no desenvolvimento. Embora o modelo existente apresente diversas oportunidades de melhoria, o novo modelo foi projetado não apenas para corrigir essas deficiências, mas também para se adequar às novas regras de negócio e funcionalidades previstas, garantindo uma estrutura mais eficiente, moderna e escalável.

A Figura 3.7 apresenta o modelo para o novo sistema. Esse modelo foi desenvolvido com o auxílio da ferramenta online *Vertabelo* para construção de modelos entidade-relacionamento (VERTABELO, 2025).

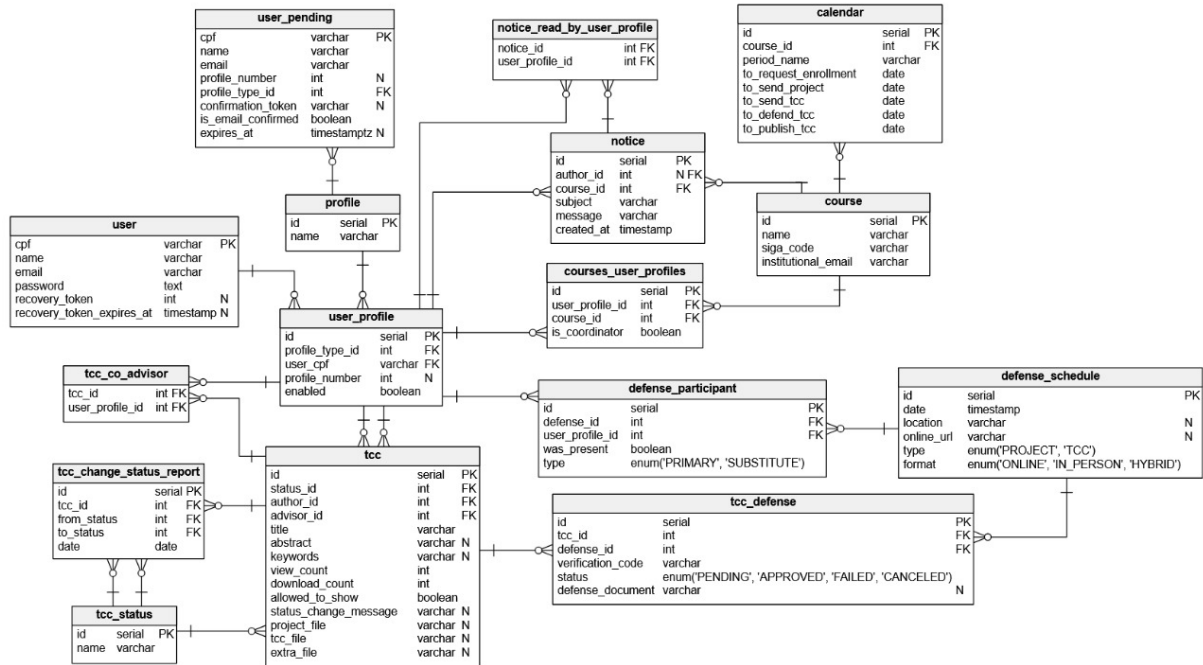


Figura 3.7: Modelo do banco de dados do novo sistema.

No modelo proposto, as entidades **user**, **user_profile**, **profile** e **user_pending** estão relacionadas com o processo de autenticação e gerenciamento de usuários do sistema.

As entidades **course** e **course_user_profiles** representam os cursos cadastrados no sistema e os perfis de usuários associados a cada curso, respectivamente. A entidade **calendar** também está relacionada diretamente aos cursos, pois é responsável por armazenar as datas importantes definidas pela coordenação de curso que devem ser seguidas pelos discentes ao longo do semestre letivo, como a data para envio do projeto, a data para envio da versão final e a data para agendamento da defesa.

Para tudo que envolva os avisos/notificações do sistema, as entidades **notice** e **notice_read_by_user_profile** são responsáveis por armazenar as notificações enviadas aos usuários e registrar quais usuários já leram cada notificação, respectivamente.

No que diz respeito ao fluxo de desenvolvimento dos TCCs, as entidades **tcc**, **tcc_status**, **tcc_change_status_report** entre outras presentes na figura, armazenam todas as informações relacionadas aos TCCs (e.g., seus status, mudanças de status,

(co)orientadores, defesas e participantes das bancas).

3.3 Planejamento tecnológico

Para o desenvolvimento do novo sistema, estão sendo utilizadas tecnologias modernas e amplamente adotadas na indústria de software. A escolha dessas tecnologias foi feita levando em consideração a familiaridade da equipe de desenvolvimento e as estatísticas de popularidade disponíveis no site oficial do gerenciador de pacotes do *Node.js*³, visando garantir a eficiência, escalabilidade e facilidade de manutenção do sistema.

3.3.1 Linguagem de programação

Como principal linguagem de programação, é utilizado o *TypeScript* (TS)⁴. Esta linguagem de programação foi desenvolvida pela Microsoft como um superconjunto da linguagem *JavaScript* (JS), ou seja, tudo que é *Javascript* também é válido no *TypeScript*, mas não o inverso. O *TypeScript* une as características de uma linguagem amplamente utilizada no desenvolvimento web, com a segurança da tipagem estática, oferecendo uma experiência de desenvolvimento mais segura e confiável (TYPESCRIPT, 2025). Atualmente, o *TypeScript* registra aproximadamente 90.503.131 downloads semanais no *npm*, demonstrando sua ampla adoção na indústria.

Para o projeto, o *TypeScript* desempenha um papel central como linguagem base para o desenvolvimento do software. A escolha estratégica de empregar *TypeScript* é respaldada pela segurança que o desenvolvimento de um software com uma linguagem fortemente tipada traz e pela utilização de bibliotecas especializadas para o desenvolvimento web: *React* para o desenvolvimento do *frontend* e *Node.js* para o *backend*.

3.3.2 Frontend

Para o desenvolvimento do *frontend* é adotado o *React*⁵, uma poderosa biblioteca para a construção de interfaces de usuário interativas (REACT, 2025). Baseado em componentes

³<<https://www.npmjs.com/>>

⁴<<https://www.npmjs.com/package/typescript>>

⁵<<https://www.npmjs.com/package/react>>

reutilizáveis, o *React* simplifica o desenvolvimento de aplicações web dinâmicas, oferecendo uma abordagem eficaz para a criação de interfaces modernas e responsivas. Atualmente, o *React* contabiliza aproximadamente 43.546.734 downloads semanais no *npm*, reforçando sua popularidade e ampla utilização.

A adoção do *React* neste projeto é estratégica para a construção de uma interface de usuário eficiente e escalável. Sua arquitetura centrada em componentes alinha-se perfeitamente com a modularidade desejada, facilitando o desenvolvimento e manutenção do *frontend* da aplicação (REACT, 2025).

Associado ao *React*, também é utilizado o *Vite*⁶, uma ferramenta de construção de projetos de *frontend* moderna e otimizada para aplicações web (VITE, 2025). O *Vite* se destaca por sua velocidade e eficiência, proporcionando uma experiência de desenvolvimento fluida e responsiva. Atualmente, o *Vite* registra aproximadamente 149.195.072 downloads semanais no *npm*, evidenciando seu crescimento e adoção na comunidade.

3.3.3 Backend

No lado do *backend*, é utilizado o *Node.js*, uma plataforma de execução de *JavaScript* no lado do servidor (JAVASCRIPT, 2025). Reconhecida por sua eficiência e escalabilidade, o *Node.js* transforma o desenvolvimento de aplicações *backend*, permitindo a criação de servidores altamente performáticos e responsivos.

A escolha do *Node.js* como plataforma *backend* é estratégica para alcançar uma arquitetura coesa e eficiente. Sua capacidade de lidar com operações assíncronas e orientação a eventos é especialmente vantajosa, garantindo um *backend* ágil e responsivo.

No sistema, também é utilizado o *NestJS* (NESTJS, 2025), para a construção de uma API de Transferência Representacional de Estado (do inglês, *Representational State Transfer* – REST), que será responsável por gerenciar as requisições e respostas entre o *frontend* e o *backend* do sistema. A API REST é uma abordagem arquitetural utilizada no desenvolvimento de aplicações web, por sua simplicidade e compatibilidade com o protocolo *Hypertext Transfer Protocol* (HTTP) (Red Hat, 2020). O *framework* feito em *TypeScript*, *NestJS*, associada a essa abordagem, permite a construção de APIs escaláveis

⁶<<https://www.npmjs.com/package/vite>>

e bem organizadas, utilizando conceitos de modularização, injeção de dependências e programação orientada a objetos (GOWDA; GOWDA, 2024).

Já integrado nativamente no *NestJS*, o sistema também utiliza o *Jest*, que é um *framework* de testes para *JavaScript* que permite a execução de diversos tipos de testes, como os testes unitários que serão implementados de forma automatizada no novo sistema (JEST, 2025). Sua utilização no projeto é essencial para garantir a qualidade dos códigos desenvolvidos e a confiabilidade do sistema.

Para a realização de tarefas assíncronas e o gerenciamento dessas tarefas em filas de processamento, será utilizado o *RabbitMQ*, que é um sistema de mensagens robusto e amplamente adotado na indústria de software (RABBITMQ, 2025). No contexto do novo sistema, o *RabbitMQ* permite que tarefas demoradas, como o envio de notificações por e-mail, sejam processadas de forma assíncrona, melhorando o tempo de resposta da aplicação e garantindo uma experiência de usuário mais fluida.

3.3.4 Banco de dados

Na camada de persistência do sistema, são utilizados dois sistemas de gerenciamento de banco de dados: o *PostgreSQL* e o *Redis*. O *PostgreSQL* é um robusto sistema de gerenciamento de banco de dados relacional de código aberto (POSTGRES, 2025). Já o *Redis* é um banco de dados chave-valor, também de código aberto, que armazena dados em memória, amplamente utilizado para armazenamento de dados temporários e para realizar *cache* de dados (REDIS, 2025).

Conhecido por sua confiabilidade, extensibilidade e conformidade com padrões, o *PostgreSQL* é uma escolha preferencial para projetos que exigem armazenamento eficiente e gerenciamento avançado de dados. A arquitetura robusta do *PostgreSQL*, combinada com seu suporte a recursos avançados, como transações e personalização de tipos, oferece uma base sólida para a persistência e manipulação de informações do projeto.

Já o *Redis* é importante no projeto pois irá armazenar os dados de sessão dos usuários, garantindo segurança para as informações sensíveis dos usuários autenticados no sistema, que não serão expostas na camada de aplicação e que o sistema possa ter acesso a essas informações de forma rápida e eficiente em cada requisição.

Associada à utilização do *PostgreSQL*, é utilizado o *Prisma*⁷, que faz o Mapeamento Objeto-Relacional (do inglês, *Object-Relational Mapping* – ORM) de maneira moderna e eficiente para *Node.js* e *TypeScript* (PRISMA, 2025). Atualmente, o *Prisma* contabiliza cerca de 4.667.189 downloads semanais no *npm*, refletindo sua relevância no ecossistema de desenvolvimento. Para o projeto, o *Prisma* simplifica a interação com o banco de dados, e levando em consideração o fato de que os desenvolvedores do projeto são principalmente estudantes, o *Prisma* irá facilitar a construção de consultas dos dados, manipulações e transações de forma intuitiva e segura. Além disso, o *Prisma* também irá fornecer uma camada adicional de abstração no gerenciamento estrutural do banco de dados, permitindo que o desenvolvimento e a manutenção do sistema sejam mais produtivos.

3.3.5 Controle de versão

Para o controle de versão do código-fonte do sistema, está sendo utilizado o *Git*, que é um sistema de controle de versão distribuído gratuito e de código aberto, amplamente utilizado na indústria de software (GIT, 2025). O *Git* permite que os desenvolvedores acompanhem as alterações no código-fonte ao longo do tempo, facilitando a colaboração e o gerenciamento de versões do software.

Associado diretamente ao *Git*, todo o histórico do código-fonte e seus arquivos estão sendo armazenados no *GitHub*, que é uma plataforma baseada na web que se destaca como um repositório de código-fonte, viabilizando o compartilhamento e a colaboração eficiente entre desenvolvedores (GITHUB, 2025). Desde sua criação, o *GitHub* evoluiu para além de um simples sistema de controle de versão, tornando-se um ecossistema essencial para o desenvolvimento de software moderno.

A escolha do *Git* e do *GitHub* neste projeto não apenas reflete a excelência em controle de versão, mas também reconhece sua influência no aprimoramento da colaboração e na eficácia do desenvolvimento de software. Através da utilização do *GitHub*, busca-se promover um ambiente de trabalho integrado, onde a gestão eficiente do código-fonte e a colaboração contínua são fundamentais para o sucesso do projeto.

⁷<<https://www.npmjs.com/package/prisma>>

3.3.6 Infraestrutura

O sistema será hospedado em um servidor disponibilizado pelo Núcleo de Recursos Computacionais (NRC) do ICE, que é o setor responsável por prover uma infraestrutura tecnológica de ponta para o Instituto. A infraestrutura do novo sistema, de forma geral, será gerenciada através de *containers* em *Docker*. O *Docker* é uma plataforma de código aberto que permite empacotar, distribuir e executar aplicações de forma isolada e padronizada (DOCKER, 2025). O uso do *Docker* é uma escolha estratégica para garantir que o sistema seja facilmente implantável, atualizável e extremamente portátil.

Estruturalmente, no servidor, o sistema será organizado em dois ambientes distintos: um ambiente de homologação, onde novas funcionalidades e correções serão validadas antes de serem lançadas; e o ambiente de produção, onde o sistema estará disponível para o usuário final. Essa separação de ambientes é uma prática que permite melhores testes do ponto de vista do usuário e garante a estabilidade do sistema em produção.

A utilização do *Docker* traz mais praticidade e eficiência no processo de desenvolvimento e atualização do sistema, o que no *tcc-web* é um obstáculo, visto que o processo de atualização do sistema é complexo e totalmente manual, principalmente levando em consideração que muitos participantes do projeto não possuem experiência prévia na implantação de sistemas web.

Associado à utilização do *Docker*, será utilizado o *Jenkins*, que é uma ferramenta de integração e entrega contínuas (do inglês, Continuous Integration/Continuous Delivery – CI/CD) amplamente adotada na indústria de software (JENKINS, 2025). Com essa poderosa ferramenta, será possível automatizar todo o processo de construção e implantação do sistema, desde a compilação do código até a execução de testes. Com sua utilização, o novo sistema não irá sofrer com o mesmo problema do *tcc-web* que, como dito anteriormente, possui um processo de implantação manual e propenso a erros.

Com a funcionalidade de execução de rotinas automatizadas do *Jenkins*, também será possível implementar duas rotinas importantes para o novo sistema: o *backup* periódico do banco de dados e dos arquivos de TCC armazenados no servidor. Essas rotinas serão essenciais para garantir a integridade e a segurança dos dados do sistema, permitindo que, em caso de falhas ou problemas como a corrupção dos dados, seja possível

restaurar o sistema para um estado anterior, minimizando os danos sofridos.

Além disso, será implementada a integração do *Grafana* com o sistema, que é um *software* de código aberto feito para que os sistemas possam ser monitorados de forma mais eficiente (GRAFANA, 2025). O *Grafana* irá coletar e exibir métricas de desempenho dos *containers* do sistema, permitindo que haja um monitoramento em tempo real do sistema de forma prática e eficiente. A ferramenta também permite a criação de alertas automáticos que podem ser enviados diretamente aos administradores do sistema quando um problema for detectado, além de permitir o armazenamento dos *logs* do sistema, facilitando a identificação e resolução de problemas. A utilização do *Grafana* resolve um dos problemas do *tcc-web*, que é a dificuldade de monitoramento do ambiente de produção.

4 Desenvolvimento Inicial do Novo Sistema

Este capítulo apresenta os principais aspectos do desenvolvimento inicial do novo sistema. Enquanto a Seção 4.1 detalha como o desenvolvimento do novo sistema está sendo gerenciado, com a criação e distribuição de tarefas, a Seção 4.2 descreve o estado atual do desenvolvimento, com as implementações já concluídas e as que estão em andamento.

4.1 Gestão de atividades

Para um melhor gerenciamento do desenvolvimento do novo sistema, foi criado um projeto na ferramenta de gestão de projetos *Jira* (JIRA, 2025). O *Jira* facilita a gestão de atividades de um projeto, como a atribuição de responsáveis a uma determinada tarefa e o acompanhamento do progresso do projeto de forma eficiente.

No *Jira*, foi criado um quadro *Kanban*, que é uma ferramenta de gerenciamento de projeto que auxilia na visualização do fluxo de trabalho e na organização das atividades. Tais atividades foram categorizadas de forma que possam ser facilmente identificadas e gerenciadas. Os tipos de atividades definidos no quadro são:

- **História:** representa uma funcionalidade esperada pelo usuário no sistema, compreendendo um ou mais requisitos e estar relacionada com uma ou mais tarefas;
- **Tarefa:** representa uma atividade que irá envolver entrega de código do sistema, seja de uma funcionalidade nova ou de uma melhoria;
- **Spike:** representa uma atividade que não irá envolver entrega de código, como o estudo de uma tecnologia ou planejamento de uma nova funcionalidade do sistema;
- **Bug:** representa uma atividade que visa corrigir um defeito ou comportamento inesperado encontrado no sistema.

As atividades presentes no quadro *Kanban* são movidas entre as colunas existentes conforme seu progresso, permitindo que todos os membros da equipe tenham uma visão clara do estado atual do projeto. Tais colunas foram definidas da seguinte forma:

- **A fazer:** atividades que ainda não foram iniciadas;
- **Em andamento:** atividades que estão sendo desenvolvidas;
- **Em revisão:** atividades já concluídas e estão aguardando revisão da *pull request*;
- **Desenvolvido:** atividades já concluídas e revisadas, mas que ainda não foram implantadas no servidor de homologação;
- **Em homologação:** atividades já implantadas no servidor de homologação e que estão aguardando validação;
- **Concluído:** atividades já validadas no servidor de homologação e que já foram implantadas no servidor de produção.

A Figura 4.1 ilustra o quadro do projeto no *Jira*, enquanto a Figura 4.2 apresenta o fluxo de trabalho adotado. Quando uma nova atividade é registrada, ela é inicialmente posicionada na coluna *a fazer*, onde permanece até que um membro da equipe inicie seu desenvolvimento. A partir daí, a atividade é movida para a coluna *em andamento*. Após a conclusão da etapa de codificação, ela é encaminhada para a coluna *em revisão*, no qual outro integrante da equipe realiza a análise crítica do que foi produzido. Se a revisão for aprovada, a atividade avança para a coluna *desenvolvido*, aguardando a implantação no ambiente de homologação. Caso sejam identificados problemas ou a necessidade de ajustes, a tarefa retorna à coluna anterior até que todas as correções sejam realizadas e aprovadas. Com a implantação no ambiente de homologação, a atividade passa para

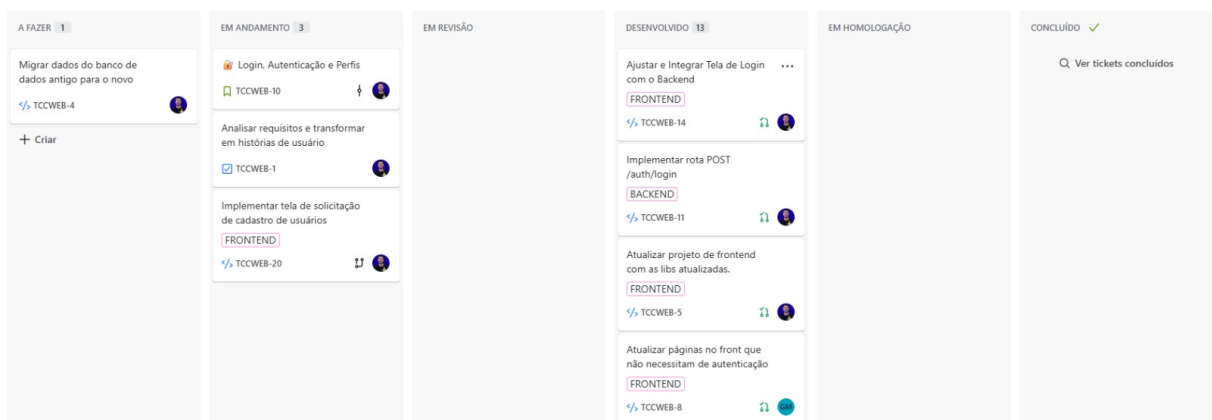


Figura 4.1: Quadro *Kanban* do projeto no *Jira*.



Figura 4.2: Fluxo de trabalho do projeto no *Jira*.

a coluna *em homologação*, onde permanece até ser validada por quem solicitou ou por um responsável pela verificação funcional. Após a validação bem-sucedida e a posterior implantação em produção, a atividade é finalmente movida para a coluna *concluído*, sendo considerada encerrada. Para atividades que não envolvem desenvolvimento de código, como as *spikes*, o fluxo de trabalho é mais simples, passando apenas pelas colunas *a fazer*, *em andamento* e *concluído*, o que reflete seu caráter exploratório e não funcional.

As atividades do projeto estão sendo especificadas e inseridas no quadro com base nos requisitos levantados na Seção 3.1 e na modelagem do sistema apresentada na Seção 3.2. Cada atividade é detalhada com uma descrição clara do que precisa ser feito e seus critérios de aceitação.

4.2 Implementações em andamento

O desenvolvimento do novo sistema encontra-se em andamento. No *backend*, a estrutura inicial do projeto foi construída com base no *framework NestJS*, já estando configurada também a ferramenta de mapeamento objeto-relacional *Prisma*. O banco de dados *PostgreSQL* foi devidamente configurado, e as entidades do sistema foram criadas de acordo com a modelagem definida na Subseção 3.2.4, servindo como base para a implementação das funcionalidades futuras.

No *frontend*, a estrutura do projeto também já foi estabelecida. Foram realizadas as configurações iniciais do *React* e do *Vite*, além da implementação do sistema de rotas, permitindo a navegação entre as páginas da aplicação. Algumas interfaces já foram desenvolvidas, embora ainda sem integração com o *backend*. Entre elas, destacam-se a tela de listagem pública dos TCCs e os formulários responsáveis pela autenticação de usuários,

recuperação de senha e solicitação de cadastro. As Figuras 4.3 e 4.4 ilustram a tela inicial do sistema, nas versões para computadores e dispositivos móveis, respectivamente.

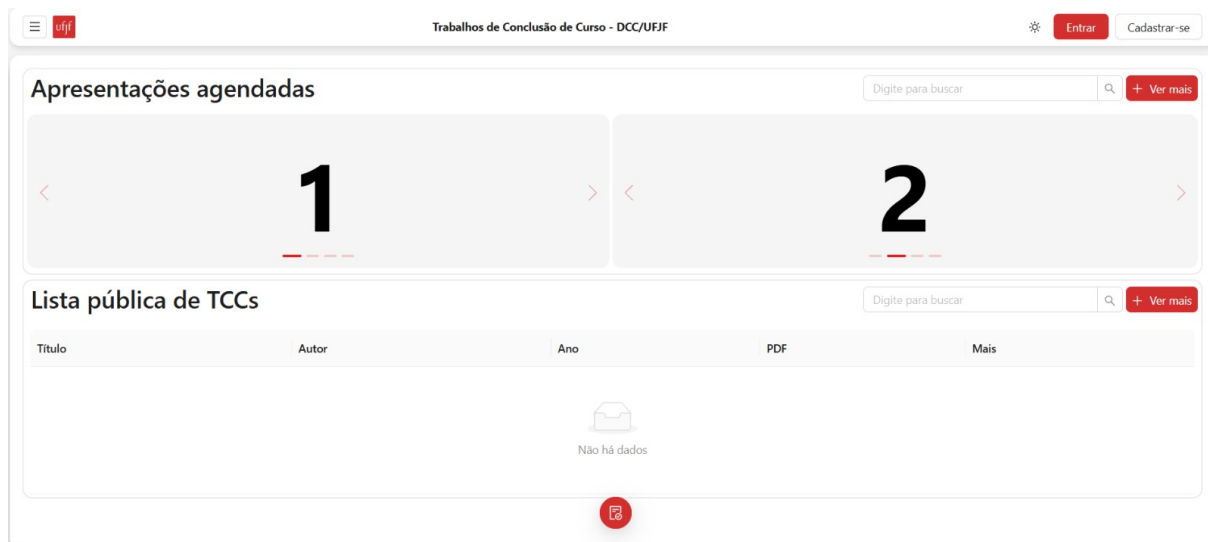


Figura 4.3: Tela inicial do novo sistema na versão para computador.



Figura 4.4: Tela inicial do novo sistema na versão para dispositivos móveis.

Seguindo o fluxo de desenvolvimento estabelecido no quadro *Kanban* do *Jira*, a primeira história de usuário especificada foi intitulada “Login, Autenticação e Perfis”. Essa história abrange toda a lógica de autenticação do sistema, incluindo a solicitação de cadastro de novos usuários, o processo de login e a troca de perfis de usuário. Conforme

ilustrado na Figura 4.5, foram criadas dez tarefas relacionadas a essa funcionalidade, das quais nove já se encontram no status *desenvolvido*, aguardando a implantação no ambiente de homologação para posterior validação.

Login, Autenticação e Perfis



▼ **Descrição**

Como usuário do sistema,
quero me autenticar com meu e-mail e senha, recuperar meu acesso caso esqueça a senha e alternar entre perfis vinculados à minha conta
para acessar minhas funcionalidades, manter minha segurança e atuar conforme diferentes permissões dentro do sistema.

▼ **Tickets vinculados** +

relates to

 TCCWEB-11 Implementar rota POST /auth/login	DESENVOLVIDO ▼		=
 TCCWEB-12 Implementar rota POST /auth/recover-password	DESENVOLVIDO ▼		=
 TCCWEB-13 Implementar rota POST /auth/recover-password/:token	DESENVOLVIDO ▼		=
 TCCWEB-14 Ajustar e Integrar Tela de Login com o Backend	DESENVOLVIDO ▼		=
 TCCWEB-15 Implementar telas de recuperação de senha	DESENVOLVIDO ▼		=
 TCCWEB-16 Implementar rota GET /auth/recover-password/:auth	DESENVOLVIDO ▼		=
 TCCWEB-17 Implementar rota POST /auth/change-profile/:profileId	DESENVOLVIDO ▼		=
 TCCWEB-18 Implementar troca de perfis no frontend	DESENVOLVIDO ▼		=
 TCCWEB-19 Implementar rota POST /auth/signup	DESENVOLVIDO ▼		=
 TCCWEB-20 Implementar tela de solicitação de cadastro de usuários	EM ANDAMENT...		=

Figura 4.5: História do usuário para autenticação no novo sistema.

Durante a implementação da autenticação, foram adotadas diversas práticas de segurança para proteger os dados sensíveis dos usuários e reduzir possíveis vulnerabilidades. Entre essas práticas, destaca-se a utilização de *cookies HttpOnly* para o armazenamento de tokens de sessão, prevenindo o acesso a essas informações por meio de *scripts* maliciosos executados no navegador (*XSS*). Além disso, foram estabelecidos requisitos

para a criação de senhas fortes, exigindo uma combinação mínima de caracteres, números e símbolos, a fim de aumentar a robustez contra ataques de força-bruta. Por fim, os dados sensíveis relacionados à sessão dos usuários são armazenados de forma segura no *Redis*, garantindo maior desempenho e proteção na gestão das credenciais.

Adicionalmente, todas as rotas da API que já foram desenvolvidas possuem testes unitários implementados, cobrindo as principais funcionalidades do sistema. Essa prática garante maior confiabilidade no código, facilita a detecção de falhas, reduz custos de manutenção e acelera o desenvolvimento de novas funcionalidades, uma vez que possíveis regressões são identificadas de forma antecipada durante o processo de integração contínua.

5 Considerações Finais

Este trabalho descreveu o projeto e o estágio atual de desenvolvimento de um novo sistema para gerenciar os TCCs dos cursos de graduação do Departamento de Ciência da Computação da UFJF. A nova solução tem como objetivo substituir o sistema legado, que apresenta sérias limitações técnicas que dificultam a implementação de novas funcionalidades e tornam o processo de correção de erros lento e complexo.

O desenvolvimento do novo sistema foi iniciado com base em uma análise dos requisitos e das limitações do sistema atual, adotando tecnologias modernas e consolidadas na indústria de software. A modelagem foi elaborada com foco na eficiência e, sobretudo, na facilidade de manutenção, considerando que os responsáveis pela sua evolução ao longo do tempo serão, em grande parte, estudantes com pouca ou nenhuma experiência prévia em desenvolvimento de sistemas web.

Para a continuidade no desenvolvimento do novo sistema, novas histórias de usuário serão criadas no *Jira*, com base nos requisitos ainda não implementados. As tarefas serão distribuídas entre os três integrantes da equipe que continuarão no projeto, dando sequência ao trabalho em ciclos acompanhados por reuniões semanais. Também está prevista a configuração do *Grafana* para monitoramento do sistema, além da construção de *pipelines* de integração e entrega contínuas (CI/CD) no *Jenkins*, com o objetivo de automatizar os processos de construção, testes e implantação.

Concluído o desenvolvimento, será realizada a migração dos dados do sistema legado para a nova aplicação, assegurando a preservação das informações dos TCCs e a continuidade do uso do sistema sem interrupções para os usuários.

A Especificação dos Requisitos

Este apêndice compila, em formato estruturado, os requisitos funcionais e não funcionais do sistema, conforme o *Documento de Requisitos* disponibilizado.

A.1 Requisitos Funcionais (RF)

RF1: O visitante pode visualizar, na página inicial do sistema, uma lista pública com todos os TCCs concluídos e aprovados.

1.1. O sistema deve exibir, para cada TCC da lista, o título do trabalho, o nome do autor, o ano da defesa e ícones que permitam o acesso direto ao documento e à página de detalhes do TCC.

1.2. Ao clicar sobre o ícone do documento em um trabalho na lista, o sistema deve exibir ao visitante o documento em PDF.

1.3. Ao clicar sobre o ícone de detalhes de um trabalho na lista, o sistema deve redirecionar o visitante para a página com os detalhes do trabalho.

RF2: O visitante pode pesquisar por TCCs na lista pública.

2.1. O sistema deve exibir uma caixa de texto única que permita a pesquisa nos campos: título, autor, curso, orientador, coorientador (se houver), resumo, palavras-chave, ano da defesa e conteúdo do trabalho.

RF3: O visitante pode visualizar os detalhes de um TCC da lista pública, incluindo o acesso ao documento na íntegra.

3.1. O sistema deve exibir, em uma página de detalhes do TCC, as informações completas do TCC: título, autor, curso, orientador, coorientador (se houver), resumo, palavras-chave, e dados da defesa (data, horário, local, banca examinadora e link para a ata preenchida e assinada). A página também deve incluir uma pré-visualização do documento em formato PDF e links para download

de todos os arquivos relacionados, como a ata de defesa, anexo (se existir) e o próprio TCC em PDF.

RF4: O visitante pode exportar a citação bibliográfica de um TCC da lista pública.

4.1. O sistema deve exibir, na página de detalhes do TCC, um botão que permite exportar a citação no formato BibTeX.

RF5: O visitante pode visualizar, na página inicial, um carrossel de cartões com todas as apresentações de TCCs programadas.

5.1. O sistema deve exibir, em cada cartão, as principais informações da apresentação: título do trabalho, autor, data, horário e local.

5.2. Ao clicar em um cartão, o sistema deve redirecionar o visitante para a página com os detalhes da apresentação do TCC.

5.3. Caso não haja apresentações agendadas, o sistema deve exibir uma mensagem informativa no lugar do carrossel.

RF6: O visitante pode pesquisar por apresentações de TCCs no carrossel.

6.1. O sistema deve exibir uma caixa de texto única que permita a pesquisa nos campos: título, autor, curso, orientador, coorientador (se houver), resumo, palavras-chave, data, horário, local e banca examinadora.

RF7: O visitante pode visualizar os detalhes da apresentação de um TCC.

7.1. O sistema deve exibir, em uma página específica, as informações completas da apresentação do TCC: título, autor, curso, orientador, coorientador (se houver), resumo, palavras-chave, data, horário, local e banca examinadora.

RF8: O visitante pode exportar as informações da apresentação de um TCC.

8.1. O sistema deve exibir, na página de detalhes da apresentação, um botão que permite exportar as informações do evento em formato .ical.

RF9: O visitante pode visualizar, na página inicial, um carrossel de cartões com todos os seminários de apresentação de projetos programados.

- 9.1. O sistema deve exibir, em cada cartão, as principais informações dos seminários: data, horário, local e a lista (com título e autor) dos projetos a serem apresentados naquele seminário.
- 9.2. Ao clicar em um cartão, o sistema deve redirecionar o visitante para a página com os detalhes do seminário.
- 9.3. Caso não haja seminários agendados, o sistema deve exibir uma mensagem informativa no lugar do carrossel.

RF10: O visitante pode pesquisar por seminários de apresentação de projetos no carrossel.

- 10.1. O sistema deve exibir uma caixa de texto única que permita a pesquisa nos campos: título, autor, curso, orientador, coorientador (se houver), resumo, palavras-chave, data, horário, local e banca examinadora.

RF11: O visitante pode visualizar os detalhes do seminário de apresentação de projetos.

- 11.1. O sistema deve exibir, em uma página específica, as informações completas do seminário: data, horário, local, banca examinadora e a lista dos projetos a serem apresentados naquele seminário.
 - Para cada projeto da lista, o sistema deve exibir: título, autor, curso, orientador, coorientador (se houver), resumo e palavras-chave.

RF12: O visitante pode exportar as informações de um seminário de apresentação de projetos.

- 12.1. O sistema deve exibir, na página de detalhes do seminário de apresentação de projetos, um botão que permite exportar as informações do evento em formato .ical.

RF13: O visitante pode verificar na página inicial a autenticidade de documentos emitidos pelo sistema, como cartas de agradecimento por participação em bancas, usando códigos disponíveis nesses documentos.

- 13.1. O sistema deve exibir um campo para inserção do código de validação e um botão para iniciar a verificação.

13.2. Após a verificação, o sistema deve exibir uma mensagem de texto informando se o documento é válido ou desconhecido.

13.3. Para códigos válidos, deve exibir também um link ou botão para direcionar o visitante à página de detalhes do TCC correspondente.

RF14: O visitante pode solicitar seu cadastro no sistema com um novo perfil de aluno, professor, servidor ou externo.

14.1. O sistema deve exibir, na página inicial, um botão de cadastro que redireciona o visitante para a página de solicitação de cadastro.

14.2. Na página de cadastro, o sistema deve solicitar inicialmente o CPF do usuário, utilizado como identificador único. Caso não exista um usuário com esse CPF, o sistema solicitará os dados comuns a todos os perfis: nome completo, e-mail institucional e senha de acesso. Caso o CPF já esteja associado a um usuário, essas informações não serão solicitadas novamente.

14.3. Em seguida, o sistema deve solicitar o tipo de perfil que o usuário deseja cadastrar (aluno, professor, servidor ou externo). O sistema deve exibir texto informativo sobre os tipos de perfis para evitar cadastro em perfil incorreto.

14.4. De acordo com o perfil selecionado, o sistema deverá solicitar os dados adicionais específicos:

- Para o perfil aluno: o sistema deve solicitar o curso de graduação (selecionado a partir de uma lista de cursos cadastrados) e o número de matrícula no curso (identificador único do perfil).
- Para o perfil professor: o sistema deve solicitar o número de SIAPE (identificador único do perfil).
- Para o perfil servidor: nenhum dado adicional é exigido (usar o CPF como identificador único do perfil).
- Para o perfil externo: nenhum dado adicional é exigido (usar o CPF como identificador único do perfil).

14.5. O sistema deve efetuar as seguintes validações nos dados informados pelo solicitante

- Não pode haver perfil duplicado para um mesmo usuário (de acordo com o identificador único do perfil)
 - CPF válido;
 - Senha forte (mínimo de 8 caracteres com 1 letra maiúscula, 1 letra minúscula, 1 número e 1 caractere especial);
 - Verificação da senha (segundo campo para digitação da senha)
 - Para o perfil professor: e-mail institucional terminado em ufjf.br e SIAPE com 7 dígitos;
 - Para o perfil servidor: e-mail institucional terminado em ufjf.br;
 - Para o perfil aluno: e-mail institucional terminado em ufjf.br e número de matrícula válido e compatível com o curso selecionado (números de matrícula possuem padrão XXXXYZZZAA, onde XXXX é o ano de ingresso, YY é o código numérico do curso, ZZZ é um contador numérico e AA são uma ou duas letras (opcionais) para alunos com cursos de segundo ciclo).
- 14.6.** Após o preenchimento e validação dos dados, o sistema deve enviar um token de verificação para o e-mail informado e o solicitante deve inserir esse token na tela de cadastro para concluir o pedido.
- 14.7.** Ao concluir o pedido, o sistema deve informar ao solicitante que o pedido de cadastro foi registrado com sucesso e que será encaminhado para aprovação. O sistema também deve informar ao solicitante que, uma vez que o cadastro seja aprovado ou reprovado, ele será notificado por e-mail.
- 14.8.** O sistema deve encaminhar a solicitação de cadastro para aprovação por algum coordenador ou secretário de curso. No caso de alunos de graduação, a solicitação será encaminhada aos coordenadores e secretários do referido curso. Para os demais perfis, a solicitação será encaminhada a todos os coordenadores e secretários, podendo qualquer um deles fazer a aprovação/reprovação do pedido.

RF15: O usuário com perfil previamente cadastrado e aprovado pode se autenticar (fazer login) no sistema.

- 15.1. O sistema deve realizar a autenticação utilizando o CPF do usuário e senha por ele escolhida.
- 15.2. O sistema deve oferecer ao usuário um link para recuperação de senha esquecida.
- 15.3. O sistema deve bloquear o acesso do usuário caso ele apenas possua perfis pendentes de aprovação.
- 15.4. Para usuários que ocupem função de coordenador ou secretário de curso de graduação, o sistema deve realizar autenticação em dois fatores, enviando token de verificação para o e-mail cadastrado que deverá ser inserido no sistema pelo usuário para concluir a autenticação.
- 15.5. Após uma autenticação com dois fatores bem sucedida, o sistema deverá dispensar o usuário do segundo fator de autenticação por um período de 12 horas, desde que a autenticação aconteça a partir do mesmo endereço IP.
- 15.6. Uma vez que o usuário se autentique com sucesso, o sistema deverá direcioná-lo para a página inicial de seu perfil.

RF16: O usuário autenticado pode se desautenticar (fazer logout) do sistema.

- 16.1. O sistema deve finalizar completamente a sessão ativa do usuário (sem necessidade de confirmação), e redirecionar o mesmo para a página pública inicial do sistema.
- 16.2. O sistema deve finalizar automaticamente a sessão ativa do usuário em caso de inatividade superior a 30 minutos.

RF17: O usuário autenticado pode consultar seus dados pessoais e informações de perfis.

- 17.1. O sistema deve exibir ao usuário já autenticado seus dados pessoais comuns a todos os perfis (CPF, nome completo e endereço de e-mail), bem como a lista de perfis cadastrados e seus respectivos dados adicionais (incluindo o identificador único do perfil).
- 17.2. O sistema deve diferenciar perfis já aprovados daqueles pendentes de aprovação.

RF18: O usuário autenticado pode alterar seu endereço de e-mail.

- 18.1. O sistema deve permitir que o usuário já autenticado faça a alteração do endereço de e-mail.
- 18.2. O sistema deve aplicar as mesmas regras de validação de e-mail empregadas durante o cadastro dos usuários.
- 18.3. Após preenchimento do novo endereço de e-mail, o sistema deve enviar um token de verificação para o e-mail informado e o solicitante deve inserir esse token na tela de cadastro para concluir o processo.

RF19: O usuário autenticado pode alterar sua senha.

- 19.1. O sistema deve permitir que o usuário já autenticado faça o cadastro de uma nova senha.
- 19.2. O sistema deve aplicar as mesmas regras de validação da nova senha que são empregadas durante o cadastro dos usuários.

RF20: O usuário autenticado pode solicitar a recuperação da senha esquecida.

- 20.1. O sistema deve permitir que o usuário não autenticado que tenha esquecido sua senha solicite a recuperação a partir de seu CPF.
- 20.2. A partir da identificação do usuário pelo CPF, o sistema deve enviar uma mensagem para o e-mail cadastrado do usuário com um link temporário único para acesso à página de cadastro de nova senha.
- 20.3. O sistema deve construir o link a partir de um token de autenticação temporário único, com validade de 3 horas.
- 20.4. O sistema deve permitir o acesso à página de cadastro de nova senha unicamente através do link enviado por e-mail, dentro do período de validade do token.
- 20.5. O sistema deve aplicar as mesmas regras de validação da nova senha que são empregadas durante o cadastro dos usuários.

RF21: O usuário autenticado pode escolher/modificar o perfil com o qual irá interagir com o sistema.

21.1. O sistema deve exibir em um menu a lista de perfis para o usuário, contendo o nome do perfil e seu identificador único. Perfis que estejam pendentes de aprovação devem ser listados de maneira visualmente diferenciada dos demais.

21.2. O sistema deve impedir que perfis pendentes de aprovação sejam selecionados para uso.

21.3. Se o usuário tiver mais de um perfil, o sistema deve, no momento da autenticação, selecionar inicialmente o perfil utilizado por último pelo usuário.

RF22: O usuário autenticado pode visualizar, em página específica, uma lista completa de usuários cadastrados no sistema.

22.1. O sistema deve exibir, para cada usuário, seu nome completo, seu endereço de e-mail, os primeiros e últimos dígitos do CPF (os dígitos intermediários devem ser ocultados por questões de privacidade e proteção de dados pessoais), e um ícone para acesso às informações dos perfis do usuário.

22.2. Ao clicar sobre o ícone de perfis, o sistema deve exibir a lista de perfis para o referido usuário, listando o tipo do perfil e parte dos dígitos do identificador único do perfil (ocultando outra parte por questões de privacidade e proteção de dados pessoais).

RF23: O usuário autenticado pode pesquisar por outros usuários na lista de usuários cadastrados no sistema.

23.1. O sistema deve exibir uma caixa de texto única que permita a pesquisa nos campos: nome, e-mail, CPF e identificador único dos perfis.

RF24: Ao selecionar um perfil de aluno, o usuário deve ver em sua página inicial a linha do tempo com todas as etapas relativas ao desenvolvimento do TCC, destacando a etapa em que se encontra no momento.

- 24.1.** O sistema deve exibir, para cada etapa da linha do tempo, um texto descritivo que oriente o aluno sobre o que acontece na etapa, bem como destacar quem é o ator responsável pelas ações da etapa (aluno, coordenador ou orientador).
- 24.2.** O sistema deve destacar a etapa em que TCC do aluno se encontra no momento, diferenciando visualmente as etapas já concluídas da etapa atual e das etapas futuras.
- 24.3.** Caso existam ações a serem executadas pelo aluno na etapa atual de seu TCC, o sistema deve exibir botões de atalho para a execução da ação (ex. botão de solicitar matrícula em TCC I para o aluno que está na primeira etapa).
- 24.4.** Caso o TCC já esteja concluído e disponível na lista pública, o sistema deve exibir um botão que direciona o aluno à página pública de detalhes de seu TCC.

RF25: O aluno pode, após sua autenticação no sistema, visualizar os dados cadastrados e os documentos de seu próprio TCC, independente da etapa do fluxo de desenvolvimento em que o projeto/trabalho esteja.

- 25.1.** O sistema deve exibir, em uma página específica, as informações atualmente disponíveis sobre o TCC do aluno como título, autor, curso, orientador, coorientador (se existir), resumo, palavras-chave, etapa atual do projeto/trabalho no fluxo de desenvolvimento do TCC, pré-visualização do documento em formato PDF (do projeto ou do trabalho, de acordo com a etapa atual), informações sobre a apresentação do projeto e defesa do trabalho (data, horário, local, banca examinadora), além de uma e links para download de todos os documentos relacionados ao TCC (ata de defesa – se existir, anexo – se existir, e os próprios PDFs do projeto e do trabalho).

RF26: O aluno que possui um projeto na etapa inicial de “solicitação de matrícula em TCC I” pode, após sua autenticação no sistema, solicitar sua matrícula na disciplina de TCC I.

- 26.1.** O sistema deve solicitar que o aluno informe o título provisório do projeto (que

pode ser apenas um tema ou mesmo área de concentração) e o professor orientador, a ser escolhido na lista de usuários com perfil de professor cadastrados no sistema.

- 26.2.** Após o envio das informações pelo aluno, o sistema deve mover o projeto para a etapa de “aguardando aceite de orientação do projeto” e notificar o professor orientador para que este manifeste ciência do pedido de orientação.

RF27: O aluno que possui um projeto na etapa de “aguardando aceite de orientação do projeto” ou “aguardando a matrícula em TCC I no SIGA” pode, após sua autenticação no sistema, cancelar sua solicitação de matrícula em TCC I.

- 27.1.** O sistema deve notificar o orientador e a coordenação do cancelamento da solicitação de matrícula e mover o projeto de volta à etapa inicial de “solicitação de matrícula em TCC I”.

RF28: O aluno que possui um projeto na etapa de “desenvolvimento do projeto” pode, após sua autenticação no sistema, editar os dados cadastrados e os documentos de seu projeto.

- 28.1.** O sistema deve exibir, em uma página específica, todas as informações atualmente cadastradas para o projeto, bem como o arquivo em PDF do projeto previamente anexado (se existir).

- 28.2.** O sistema deve permitir que o aluno modifique as informações de identificação do projeto: título, resumo e palavras-chave.

- 28.3.** O sistema deve permitir que o aluno envie 1 (um) arquivo PDF do projeto, sobrescrevendo o que existir no sistema (caso exista).

- 28.4.** Caso o aluno modifique o orientador do projeto, o sistema deve solicitar confirmação imediata do aluno e, em caso de confirmação positiva, mover o projeto para a etapa de “aguardando aceite de orientação do projeto” e notificar o antigo orientador da mudança bem como o novo orientador para que este manifeste ciência do pedido de orientação.

- 28.5. Caso o aluno queira editar a lista de coorientadores do projeto, o sistema deve permitir a escolha a partir da lista de usuários com perfil de professor, servidor ou externo.
 - 28.6. O sistema deve limitar a lista de coorientadores em um máximo de 2 usuários.
 - 28.7. O sistema deve permitir que o aluno informe que seu projeto está pronto para a apresentação, validando se as seguintes informações obrigatórias estão cadastradas: título, orientador, resumo, palavras-chave e o documento PDF principal do projeto anexado.
 - 28.8. Quando o aluno informar que o projeto está pronto para apresentação, o sistema deve mover o projeto para a etapa de “aguardando autorização para apresentação do projeto” e notificar o professor orientador para que este autorize a apresentação.
- RF29:** O aluno que possui um projeto na etapa de “apresentação do projeto” pode, após sua autenticação no sistema, visualizar as informações sobre o seminário de apresentação ao qual foi designado.
- 29.1. O sistema deve exibir as informações de data, horário, local e composição da banca examinadora para o seminário de apresentação ao qual o projeto foi alocado.
- RF30:** O aluno que possui um trabalho na etapa de “solicitação de matrícula em TCC II” pode, após sua autenticação no sistema, solicitar sua matrícula na disciplina de TCC II.
- 30.1. O sistema deve exibir o título e o professor orientador atualmente cadastrados, permitindo que o aluno edite estas informações antes de confirmar a solicitação de matrícula.
 - 30.2. Caso o aluno solicite alteração de orientador, o sistema deve mover o trabalho para a etapa de “aguardando aceite de orientação do trabalho” e notificar o antigo orientador da mudança bem como o novo orientador para que este manifeste ciência do pedido de orientação.

30.3. Caso o aluno não solicite alteração de orientador, o sistema deve mover o trabalho para a etapa de “aguardando matrícula em TCC II no SIGA” e notificar a coordenação de curso para que esta efetue a matrícula.

RF31: O aluno que possui um trabalho na etapa de “aguardando aceite de orientação do trabalho” ou “aguardando matrícula em TCC II no SIGA” pode, após sua autenticação no sistema, cancelar sua solicitação de matrícula em TCC II.

31.1. O sistema deve notificar o orientador e a coordenação do cancelamento da solicitação de matrícula e mover o trabalho para a etapa inicial de “solicitação de matrícula em TCC II”.

RF32: O aluno que possui um trabalho na etapa de “desenvolvimento do trabalho” pode, após sua autenticação no sistema, editar os dados cadastrados e os documentos de seu trabalho.

32.1. O sistema deve exibir, em uma página específica, todas as informações atualmente cadastradas para o trabalho, bem como os arquivos previamente anexados (o que inclui o projeto, anexado durante o fluxo do projeto).

32.2. O sistema deve permitir que o aluno modifique as informações de identificação do trabalho: título, resumo e palavras-chave.

32.3. O sistema deve permitir que o aluno envie 1 (um) arquivo PDF do trabalho, sobrescrevendo o que existir no sistema (caso exista).

- O PDF do trabalho é diferente do PDF do projeto, sendo que o trabalho não deve sobrescrever o projeto.

32.4. O sistema deve permitir que o aluno envie 1 (um) arquivo de anexos, em formato compactado (.zip, .rar, .tar.gz ou .tar.bz2), sobrescrevendo o que existir no sistema (caso exista).

32.5. Caso o aluno modifique o orientador do trabalho, o sistema deve solicitar confirmação imediata do aluno e, em caso de confirmação positiva, mover o trabalho para a etapa de “aguardando aceite de orientação do trabalho” e notificar

o antigo orientador da mudança bem como o novo orientador para que este manifeste ciência do pedido de orientação.

- 32.6.** Caso o aluno queira editar a lista de coorientadores do trabalho, o sistema deve permitir a escolha a partir da lista de usuários com perfil de professor, servidor ou externo.
- 32.7.** O sistema deve limitar a lista de coorientadores em um máximo de 2 usuários.
- 32.8.** Caso o aluno queira editar a lista de integrantes da banca examinadora, o sistema deve permitir a escolha a partir da lista de usuários com perfil de professor, servidor ou externo.
- 32.9.** O sistema deve solicitar sempre que o aluno indique, para cada integrante da banca examinadora, se sua participação será como titular ou suplente.
- 32.10.** O sistema deve incluir automaticamente o orientador e os coorientadores como titulares na lista de participantes da banca examinadora, impedindo que o aluno os remova desta lista.
- 32.11.** O sistema deve limitar a lista de participantes da banca examinadora em um máximo de 5 participantes titulares e 2 participantes suplentes.
- 32.12.** O sistema deve permitir que o aluno informe data e horário da defesa, a modalidade da defesa (presencial, online ou híbrida), bem como o local da mesma (sala para defesa presencial, link para defesa online, ou os dois para defesa híbrida).
- 32.13.** O sistema deve permitir que o aluno informe que seu trabalho está pronto para a defesa, validando se as seguintes informações obrigatórias estão cadastradas: título, orientador, resumo, palavras-chave, o documento PDF do trabalho anexo, a data, horário e local da defesa, e a lista de participantes da banca examinadora.
 - A banca examinadora é composta obrigatoriamente pelo orientador, pelos coorientadores (se existirem), por no mínimo dois outros membros titulares e por no mínimo um membro suplente.

- 32.14.** Quando o aluno informar que o trabalho está pronto para defesa, o sistema deve mover o trabalho para a etapa de “aguardando autorização para defesa do trabalho” e notificar o professor orientador para que este autorize a defesa.

RF33: O aluno que possui um trabalho na etapa de “defesa do trabalho” ou “aguardando lançamento de deliberação da banca examinadora” pode, após sua autenticação no sistema, gerar o PDF da ata de defesa, que deve incluir a data, horário e local da sessão pública de defesa, a composição da banca examinadora (apenas os nomes dos integrantes que efetivamente participaram da defesa), a indicação de quem foi o presidente, o nome do aluno, o título do TCC, o espaço para indicação se o trabalho foi aprovado ou reprovado, e o espaço para as assinaturas dos integrantes da banca examinadora e também do aluno.

- 33.1.** O sistema deve solicitar que o aluno indique quais integrantes da banca examinadora, de fato, participaram/participarão da defesa, para que os nomes destes sejam incluídos na ata.

- Por padrão, o sistema deve pré-selecionar os titulares como participantes da defesa, permitindo que o aluno faça a troca de titulares por suplentes, se necessário.

- 33.2.** Caso o orientador não seja selecionado como um participante da defesa, o sistema deve solicitar que o aluno indique qual integrante foi/será o presidente da banca examinadora.

- 33.3.** Mesmo após a geração com sucesso de um PDF da ata, o sistema deve permitir que o aluno faça alterações nos integrantes da banca examinadora que participaram/participarão da defesa para a emissão de nova ata, se necessário.

RF34: O aluno que possui um trabalho na etapa de “envio da versão final do trabalho” pode, após sua autenticação no sistema, editar parte dos dados cadastrados e enviar os documentos da versão final de seu trabalho.

- 34.1.** O sistema deve exibir, em uma página específica, todas as informações atualmente cadastradas para o trabalho, bem como os arquivos previamente anexados.

- 34.2.** O sistema deve permitir que o aluno modifique as informações de identificação do trabalho: título, resumo e palavras-chave.
- 34.3.** O sistema deve permitir que o aluno envie 1 (um) arquivo PDF com a versão final do trabalho, sobrescrevendo o que existir no sistema.
- 34.4.** O sistema deve permitir que o aluno envie 1 (um) arquivo de anexos, em formato compactado (.zip, .rar, .tar.gz ou .tar.bz2), sobrescrevendo o que existir no sistema (caso exista).
- 34.5.** O sistema deve permitir que o aluno envie 1 (um) arquivo PDF com a ata da defesa, sobrescrevendo a ata que existir no sistema (caso exista).
- 34.6.** O sistema deve permitir que o aluno informe que a versão final de seu trabalho está pronto para a publicação, validando se as seguintes informações obrigatórias estão cadastradas: título, orientador, resumo e palavras-chave, bem como se os documentos em PDF do trabalho e da ata de defesa estão anexados.
- 34.7.** Quando o aluno informar que a versão final do trabalho está pronta, o sistema deve mover o trabalho para a etapa de “aguardando aprovação do documento final” e notificar o professor orientador para que este aprove a versão final do trabalho.

RF35: O aluno que possui um trabalho na etapa de “correções na ata de defesa ou formação do documento final” pode, após sua autenticação no sistema, editar parte dos dados cadastrados e enviar os documentos da versão final de seu trabalho.

- 35.1.** O sistema deve exibir, em uma página específica, todas as informações atualmente cadastradas para o trabalho, bem como os arquivos previamente anexados.
- 35.2.** O sistema deve permitir que o aluno modifique as informações de identificação do trabalho: título, resumo e palavras-chave.
- 35.3.** O sistema deve permitir que o aluno envie 1 (um) arquivo PDF com a versão final do trabalho, sobrescrevendo o que existir no sistema.
- 35.4.** O sistema deve permitir que o aluno envie 1 (um) arquivo PDF com a ata da defesa, sobrescrevendo a ata que existir no sistema.

35.5. O sistema deve permitir que o aluno informe que as correções na ata de defesa ou na formatação do documento final estão prontas, validando se os documentos em PDF do trabalho e da ata de defesa estão anexados.

35.6. Quando o aluno informar que as correções na ata de defesa ou na formatação do documento final estão prontas, o sistema deve mover o trabalho para a etapa de “aguardando verificação da ata de defesa e da formatação do documento final” e notificar a coordenação de curso para que esta proceda com a verificação destes documentos.

RF36: Ao selecionar um perfil de professor, o usuário deve ver em sua página inicial uma lista de TCCs sob sua orientação para os quais existem ações pendentes de sua parte enquanto orientador.

36.1. O sistema deve exibir uma lista de TCCs que se encontram nas etapas listadas a seguir, em que há ações pendentes de serem executadas pelo professor enquanto orientador do trabalho:

- Aguardando aceite de orientação do projeto
- Aguardando autorização para apresentação do projeto
- Aguardando aceite de orientação do trabalho
- Aguardando autorização para defesa do trabalho
- Aguardando lançamento da deliberação da banca examinadora
- Aguardando aprovação do documento final

36.2. O sistema deve exibir minimamente, para cada TCC da lista, o título do trabalho, o nome completo do aluno, a etapa atual do projeto/trabalho no fluxo de desenvolvimento do TCC, botão de atalho para a página de visualização completa dos dados do TCC e também, quando aplicável, botão de ação rápida para aceitar ou recusar pedidos de orientação.

RF37: O professor pode, após sua autenticação no sistema, visualizar os dados cadastrados e os documentos de todos os TCCs sob sua orientação ou coorientação, independente da etapa do fluxo de desenvolvimento em que o projeto/trabalho esteja.

- 37.1.** O sistema deve exibir a lista de TCCs em uma página específica denominada “minhas orientações”.
- 37.2.** O sistema deve exibir minimamente, para cada TCC da lista, o título do trabalho, o nome completo do aluno, a etapa atual do projeto/trabalho no fluxo de desenvolvimento do TCC, a relação do professor com o TCC (se orientação ou coorientação) e botão de atalho para a página de visualização completa dos dados do TCC.
- 37.3.** A lista deve ser separada entre “TCCs em desenvolvimento” e “TCCs finalizados”, preferencialmente em abas distintas.
- 37.4.** O sistema deve permitir ordenar, filtrar e realizar buscas na lista de TCCs.

RF38: O professor pode, após sua autenticação no sistema, visualizar os dados cadastrados e os documentos de todos os projetos que estejam nas etapas “apresentação do projeto”, ou “aguardando lançamento de conceito do projeto”, nos quais ele tenha sido incluído como integrante da banca examinadora.

- 38.1.** O sistema deve exibir a lista de bancas em uma página específica denominada “minhas bancas”.

RF39: O professor pode, após sua autenticação no sistema, visualizar os dados cadastrados e os documentos de todos os trabalhos que estejam na etapa “defesa do trabalho” ou em qualquer etapa seguinte do fluxo, nos quais ele tenha sido incluído como integrante da banca examinadora.

- 39.1.** O sistema deve exibir a lista de bancas em uma página específica denominada “minhas bancas”.

RF40: O professor pode, após sua autenticação no sistema, gerar a carta de participação em uma defesa da qual ele tenha efetivamente participado como integrante da banca examinadora.

RF41: O professor pode, após sua autenticação no sistema, aceitar ou recusar um pedido de orientação de projeto que esteja na etapa “aguardando aceite de orientação do projeto”.

- 41.1. O sistema deve exibir, na página de visualização detalhada de TCC, os botões para aceitar ou recusar a solicitação de orientação do projeto.
- 41.2. Caso o professor recuse a orientação do projeto, o sistema deve mover o projeto para a etapa inicial de “solicitar matrícula em TCC I” e notificar o aluno para que este esteja ciente da decisão do professor.
- 41.3. Caso o professor aceite a orientação do projeto, o sistema deve mover o projeto para a etapa seguinte de “aguardando matrícula em TCC I no SIGA” e notificar o aluno e a coordenação de curso para que estes estejam cientes da decisão do orientador.

RF42: O professor orientador pode, após sua autenticação no sistema, autorizar ou impedir que um projeto sob sua orientação na etapa “aguardando autorização para apresentação do projeto” seja apresentado pelo discente à banca examinadora em seminário que será posteriormente agendado pela coordenação de curso.

- 42.1. O sistema deve exibir, na página de visualização detalhada de TCC, os botões para autorizar ou impedir a apresentação do projeto.
- 42.2. Caso o professor impeça a apresentação do projeto, o sistema deve mover o projeto para a etapa anterior de “desenvolvimento do projeto” e notificar o aluno para que este esteja ciente da decisão do orientador.
- 42.3. Caso o professor autorize a apresentação do projeto, o sistema deve mover o projeto para a etapa seguinte de “aguardando agendamento da apresentação do projeto” e notificar o aluno e a coordenação de curso para que estes estejam cientes da decisão do orientador.

RF43: O professor orientador pode, após sua autenticação no sistema, aceitar ou recusar um pedido de orientação de trabalho que esteja na etapa “aguardando aceite de orientação do trabalho”.

- 43.1. O sistema deve exibir, na página de visualização detalhada de TCC, os botões para aceitar ou recusar a solicitação de orientação do trabalho.

- 43.2.** Caso o professor recuse a orientação do trabalho, o sistema deve mover o trabalho para a etapa inicial de “solicitar matrícula em TCC II” e notificar o aluno para que este esteja ciente da decisão do professor.
- 43.3.** Caso o professor aceite a orientação do trabalho, o sistema deve mover o trabalho para a etapa seguinte de “aguardando matrícula em TCC II no SIGA” e notificar o aluno e a coordenação de curso para que estes estejam cientes da decisão do orientador.

RF44: O professor orientador pode, após sua autenticação no sistema, autorizar ou impedir que um trabalho sob sua orientação na etapa “aguardando autorização para defesa do trabalho” seja apresentado pelo discente à banca examinadora em sessão pública, conforme dados já cadastrados no sistema pelo discente.

- 44.1.** O sistema deve exibir, na página de visualização detalhada de TCC, os botões para autorizar ou impedir a defesa do trabalho.
- 44.2.** Caso o professor impeça a defesa do trabalho, o sistema deve mover o trabalho para a etapa anterior de “desenvolvimento do trabalho” e notificar o aluno para que este esteja ciente da decisão do orientador.
- 44.3.** Caso o professor autorize a defesa do trabalho, o sistema deve mover o trabalho para a etapa seguinte de “defesa do trabalho” e notificar o aluno, a coordenação de curso e todos os integrantes da banca examinadora para que estes estejam cientes da decisão do orientador.

RF45: O professor orientador pode, após sua autenticação no sistema, informar a deliberação da banca examinadora e anexar a ata de defesa de trabalho sob sua orientação na etapa “aguardando lançamento da deliberação da banca examinadora”.

- 45.1.** O sistema deve exibir, na página de visualização detalhada de TCC, os botões para informar a aprovação ou reprovação do trabalho, bem como um botão para informar que a defesa não foi realizada.
- 45.2.** Caso a defesa não tenha sido realizada, o sistema deve mover o trabalho para a etapa de “desenvolvimento do trabalho” e notificar o aluno e a coordenação

de curso para que estes estejam cientes de que a sessão pública de defesa não foi realizada.

45.3. Caso o professor informe que a banca aprovou ou reprovou o trabalho, o sistema deve solicitar que o professor anexe o arquivo em formato PDF com a ata da defesa, devidamente assinado pela banca examinadora e pelo aluno.

45.4. Caso o trabalho tenha sido reprovado pela banca, o sistema deve mover o trabalho para a etapa de “aguardando lançamento de conceito do trabalho” e notificar o aluno e a coordenação de curso para que estes estejam cientes da decisão da banca examinadora.

45.5. Caso o trabalho tenha sido aprovado pela banca, o sistema deve mover o trabalho para a etapa de “envio da versão final do trabalho” e notificar o aluno e a coordenação de curso para que estes estejam cientes da decisão da banca examinadora.

RF46: O professor orientador pode, após sua autenticação no sistema, aprovar ou reprovar o documento final de um trabalho sob sua orientação na etapa “aguardando aprovação do documento final” após as correções realizadas pelo aluno conforme sugestão da banca examinadora.

46.1. O sistema deve exibir, na página de visualização detalhada de TCC, os botões para aprovar ou reprovar o documento final do trabalho.

46.2. Caso o professor reprove o documento final do trabalho, o sistema deve mover o trabalho para a etapa anterior de “envio da versão final do trabalho” e notificar o aluno para que este esteja ciente da decisão do orientador.

46.3. Caso o professor aprove o documento final do trabalho, o sistema deve mover o trabalho para a etapa seguinte de “aguardando verificação da ata de defesa e da formatação do documento final” e notificar o aluno e a coordenação de curso para que estes estejam cientes da decisão do orientador.

RF47: O professor orientador pode, após sua autenticação no sistema, editar todos os dados cadastrados e os documentos dos TCCs sob sua orientação, exceto daqueles já concluídos e que estejam na etapa “trabalho finalizado com sucesso”.

47.1. O professor pode realizar ações e agendar defesas mesmo que estejam em desconformidade com o calendário da disciplina, desde que eles confirmem ciência.

RF48: O professor orientador pode, após sua autenticação no sistema, gerar o PDF da ata de defesa dos TCCs sob sua orientação que estejam nas etapas de “defesa do trabalho” ou “aguardando lançamento de deliberação da banca examinadora”.

RF49: O professor que estiver ocupando o cargo de coordenador de curso pode, após sua autenticação no sistema, aprovar ou reprovar cadastro pendente de usuários no sistema.

49.1. Para cadastro do tipo aluno, aprovar/reprovar apenas os pedidos do respectivo curso.

49.2. Para cadastro do tipo professor, servidor ou externo, qualquer coordenador pode aprovar qualquer cadastro.

49.3. Na hora da aprovação do cadastro, coordenador/secretário podem corrigir erros no nome e matrícula de alunos, mas não podem modificar CPF nem e-mail.

RF50: O professor que estiver ocupando o cargo de coordenador de curso pode, após sua autenticação no sistema, ter acesso às páginas de gestão de trabalhos do curso, que lista todos os TCCs dos alunos do curso.

RF51: O professor que estiver ocupando o cargo de coordenador de curso pode, após sua autenticação no sistema, editar os dados de trabalhos dos alunos do curso, inclusive daqueles já concluídos.

RF52: O professor que estiver ocupando o cargo de coordenador de curso pode, após sua autenticação no sistema, criar e editar calendários para o semestre letivo vigente.

RF53: O professor que estiver ocupando o cargo de coordenador de curso pode, após sua autenticação no sistema, enviar mensagens de avisos aos alunos que estão com TCCs em desenvolvimento no sistemas, bem como aos seus professores orientadores.

53.1. O sistema deve permitir escolher os destinatários: alunos de projetos ou de trabalhos, orientadores de projetos ou de trabalhos.

53.2. O sistema deve encaminhar as mensagens por e-mail.

RF54: O professor que estiver ocupando o cargo de coordenador de curso pode, após sua autenticação no sistema, agendar os seminários de apresentação dos projetos, podendo incluir alunos de qualquer curso.

54.1. O coordenador deve agendar um seminário de projeto definindo a data, o horário, a sala e os projetos que serão apresentados naquele seminário.

54.2. O coordenador pode gerar a ata.

54.3. O presidente da banca examinadora do seminário pode gerar a ata.

RF55: O professor que estiver ocupando o cargo de coordenador de curso pode, após sua autenticação no sistema, gerar cartas de participação em banca de qualquer trabalho de aluno de seu curso.

55.1. O sistema deve oferecer a possibilidade de fazer o download da carta ou encaminhá-la ao e-mail cadastrado no sistema.

RF56: O professor que estiver ocupando o cargo de coordenador de curso pode, após sua autenticação no sistema, excluir TCCs em andamento que tenham sido abandonados.

RF57: O professor que estiver ocupando o cargo de coordenador de curso pode, após sua autenticação no sistema, ocultar TCCs já concluídos da lista pública, desde que indique uma data limite na qual o trabalho voltará a ser público.

RF58: O professor que estiver ocupando o cargo de coordenador de curso pode, após sua autenticação no sistema, designar outros usuários do sistema para atuarem como secretário do curso, de modo a auxiliar na gestão dos TCCs do curso.

RF59: O professor que estiver ocupando o cargo de coordenador de curso pode, após sua autenticação no sistema, informar que já matriculou um aluno nas disciplinas de TCC I ou TCC II no SIGA.

- RF60:** O professor que estiver ocupando o cargo de coordenador de curso pode, após sua autenticação no sistema, informar que já lançou o conceito do aluno nas disciplinas de TCC I ou TCC II no SIGA.
- RF61:** O professor que estiver ocupando o cargo de coordenador de curso pode, após sua autenticação no sistema, aprovar ou reprovar a formatação final do documento de um trabalho.
- RF62:** O professor que estiver ocupando o cargo de coordenador de curso pode, após sua autenticação no sistema, cadastrar um TCC já defendido diretamente no sistema (sem que o mesmo precise passar pelo fluxo completo do processo).
- RF63:** Ao selecionar um perfil de servidor, o usuário deve ver em sua página inicial uma lista de TCCs em que ele possui relação de coorientação e/ou participação em banca examinadora.
- RF64:** O professor que estiver ocupando o cargo de secretário de curso pode, após sua autenticação no sistema, realizar as mesmas ações que o coordenador do respectivo curso.
- RF65:** Ao selecionar um perfil de externo, o usuário deve ver em sua página inicial uma lista de TCCs em que ele possui relação de coorientação e/ou participação em banca examinadora.
- RF66:** O administrador deve ser capaz de definir os coordenadores e secretários de um curso.
- RF67:** O administrador pode cadastrar novos cursos no sistema
- RF68:** O administrador pode editar os dados pessoais dos usuários cadastrados no sistema.
- 68.1.** Apenas o administrador pode alterar CPF e nome de usuários.
- RF69:** O administrador pode resetar senha de usuários, forçando que o usuário tenha que criar uma senha nova no próximo acesso ao sistema.

RF70: O sistema tem que mover automaticamente os projetos do status de “apresentação do projeto” para “aguardando lançamento de conceito do projeto” quando passar a data indicada de apresentação do projeto.

70.1. Verificar em rotina diária

RF71: O sistema tem que mover automaticamente os trabalhos do status de “defesa do trabalho” para “aguardando o lançamento da deliberação da banca examinadora” quando passar a data indicada de defesa do trabalho.

71.1. Verificar em rotina diária

RF72: O sistema deve solicitar confirmação de todas as ações que impliquem na mudança de etapa do fluxo de desenvolvimento de um TCC, independente de quem seja o ator responsável pela ação.

72.1. Pode-se utilizar um modal com botões de confirmar ou cancelar, sempre acompanhado de mensagem descritiva sobre o que está sendo solicitado a confirmação.

72.2. Sempre deve haver um campo de texto que permita ao ator escrever uma mensagem, que será enviada junto ao e-mail de notificação para os envolvidos.

RF73: O sistema deve, diariamente, verificar se existem e-mails pendentes de envio e realizar nova tentativa de envio dos mesmos.

RF74: Se uma defesa é reprovada, o TCC volta para as etapas anteriores. O pdf com a ata de reprovação vai continuar existindo, e quando voltar na época de definir nova defesa vai estar tudo em branco novamente pro aluno que vai cadastrar uma nova defesa (sem apagar a anterior).

A.2 Requisitos Não Funcionais (RNF)

RNF1: O sistema deve garantir a proteção das credenciais e das informações pessoais de seus usuários utilizando o *Redis*.

1.1. Não trafegar senha em texto plano.

RNF2: O frontend do sistema deve ser testado para fins de compatibilidade com os navegadores Google Chrome, Microsoft Edge, Mozilla Firefox e Safari.

2.1. Ser responsivo em celular.

RNF3: A interface gráfica da página inicial do perfil de um aluno deve dar destaque às datas e aos prazos definidos no calendário.

RNF4: O sistema deve possuir integração com serviço de e-mail do google para envio de mensagens de notificações, para autenticação de dois fatores e também para recuperação de senhas esquecidas.

4.1. O sistema deve fazer uso de uma fila para armazenar os emails que ainda não foram enviados, permitindo que sejam recuperados em caso de falha de envio.

RNF5: O sistema deve manter compatibilidade com todos os códigos de autenticação emitidos pelo sistema antigo.

RNF6: O sistema deve sempre priorizar o uso de endereço de e-mail institucional do usuário (@estudante.ufjf.br ou @ufjf.br).

RNF7: O sistema deve garantir a integridade dos dados do sistema.

7.1. O sistema deve gerar um backup do banco de dados e de documentos como atas de defesa e TCC 's em servidor remoto para proteção dos dados.

RNF8: Todo o código deve ser documentado com anotações para que possa ser gerada documentação automática.

RNF9: Todas as páginas devem possuir um menu de ajuda com orientações de usabilidade.

RNF10: Deve-se implementar rate limiter nas rotas públicas do sistema para evitar ataques de força bruta.

RNF11: O texto descritivo de cada etapa do desenvolvimento do TCC deve ser salvo em um arquivo ou em alguma variável de ambiente, permitindo que o mesmo seja editado sem a necessidade de modificar o código.

RNF12: Os calendários no sistema são para orientação/referência, mas não devem restringir ações dos atores.

12.1. O sistema não tem travas de ações em função de calendário.

12.2. Os calendários são independentes por curso.

RNF13: As informações sobre apresentações de TCCs programadas devem ter mais destaque do que a lista pública de TCCs finalizados na página inicial do sistema.

RNF14: Todas as ações que façam o projeto mudar de status deve ser confirmadas em um modal.

RNF15: As defesas tem 5 status: em preenchimento, agendada, cancelada, aprovado, reprovado.

15.1. Depois que a defesa entra no status de agendada, ele não pode mais ser excluída, devendo o fluxo levá-la para cancelada, aprovada ou reprovada.

15.2. As defesas tem um identificador único, usado pra validar ata e carta de participação.

Referências Bibliográficas

DBEAVER. *A universal database management tool for anyone who needs to work with data professionally*. 2025. Acesso em: 30 de julho de 2025. Disponível em: <<https://dbeaver.com/>>.

DOCKER. *An open platform for developing, shipping, and running applications*. 2025. Acesso em: 02 de agosto de 2025. Disponível em: <<https://www.docker.com/>>.

FIGMA. *A powerful and collaborative design tool for teams*. 2025. Acesso em: 01 de agosto de 2025. Disponível em: <<https://www.figma.com/>>.

FRANCO, G. R. L.; GUZMAN, C. Y. C. D. Design and implementation of a web-based thesis coordinator system (TCS). In: *2016 IEEE Region 10 Conference (TENCON)*. [s.n.], 2016. p. 2726–2729. Disponível em: <<https://doi.org/10.1109/TENCON.2016.7848535>>.

FRITOLA, R.; SANTANDER, V. Documentando requisitos de sistemas legados: um estudo de caso utilizando técnicas da engenharia de requisitos orientada a objetivos. In: *Anais da V Escola Regional de Engenharia de Software*. Porto Alegre, RS, Brasil: SBC, 2021. p. 139–148. Disponível em: <<https://sol.sbc.org.br/index.php/eres/article/view/18459>>.

GIT. *A free and open source distributed version control system*. 2025. Acesso em: 03 de agosto de 2025. Disponível em: <<https://git-scm.com/>>.

GITHUB. *The world's leading AI-powered developer platform*. 2025. Acesso em: 29 de julho de 2025. Disponível em: <<https://github.com/>>.

GOWDA, P.; GOWDA, A. Best practices in REST API design for enhanced scalability and security. *Journal of Artificial Intelligence, Machine Learning and Data Science*, v. 2, n. 1, p. 827–830, 2024. Acesso em: 03 de agosto de 2025. Disponível em: <<https://doi.org/10.51219/JAIMLD/priyanka-gowda/202>>.

GRAFANA. *Open source software for visualizing and analyzing metrics, logs, and traces*. 2025. Acesso em: 02 de agosto de 2025. Disponível em: <<https://grafana.com/>>.

JAVA. *The #1 programming language and development platform*. 2025. Acesso em: 30 de julho de 2025. Disponível em: <<https://www.java.com/>>.

JAVASCRIPT. *A lightweight interpreted (or just-in-time compiled) programming language with first-class functions*. 2025. Acesso em: 29 de julho de 2025. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/JavaScript>>.

JENKINS. *The leading open source automation server*. 2025. Acesso em: 02 de agosto de 2025. Disponível em: <<https://www.jenkins.io/>>.

JEST. *A delightful JavaScript testing framework with a focus on simplicity*. 2025. Acesso em: 02 de agosto de 2025. Disponível em: <<https://jestjs.io/>>.

JIRA. *A a work management tool for software teams that need to organize and track their work*. 2025. Acesso em: 02 de agosto de 2025. Disponível em: <<https://www.atlassian.com/software/jira>>.

KRUG, S. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. 3rd. ed. Berkeley, CA: New Riders, 2014.

MAVEN. *A build tool for Java projects*. 2025. Acesso em: 30 de julho de 2025. Disponível em: <<https://maven.apache.org/>>.

MYSQL. *The world's most popular open source database*. 2025. Acesso em: 30 de julho de 2025. Disponível em: <<https://www.mysql.com/>>.

NESTJS. *A progressive Node.js framework for building efficient, reliable and scalable server-side applications*. 2025. Acesso em: 30 de julho de 2025. Disponível em: <<https://nestjs.com/>>.

POSTGRES. *The world's most advanced open source database*. 2025. Acesso em: 29 de julho de 2025. Disponível em: <<https://www.postgresql.org/>>.

PRISMA. *Next-generation Node.js and TypeScript ORM*. 2025. Acesso em: 30 de julho de 2025. Disponível em: <<https://www.prisma.io/>>.

RABBITMQ. *Open source message broker software that originally implemented the Advanced Message Queuing Protocol (AMQP)*. 2025. Acesso em: 01 de agosto de 2025. Disponível em: <<https://www.rabbitmq.com/>>.

REACT. *The library for web and native user interfaces*. 2025. Acesso em: 29 de julho de 2025. Disponível em: <<https://react.dev/>>.

Red Hat. *What is a REST API?* 2020. <<https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>>. Acesso em: 29 Ago. 2025.

REDIS. *The preferred, fastest, and most feature-rich cache, data structure server, and document and vector query engine*. 2025. Acesso em: 02 de agosto de 2025. Disponível em: <<https://redis.io/>>.

SECURITY, S. *mysql-5.5 Vulnerabilities*. 2024. Acesso em: 30 de julho de 2025. Disponível em: <<https://security.snyk.io/package/linux/debian%3A8/mysql-5.5>>.

SOMMERVILLE, I. *Engenharia de Software*. 10. ed. São Paulo: Pearson, 2019. ISBN 978:85-430-2497-4.

TYPESCRIPT. *A strongly typed programming language that builds on JavaScript*. 2025. Acesso em: 29 de julho de 2025. Disponível em: <<https://www.typescriptlang.org/>>.

VAADIN. *Vaadin Flow: A Java framework for building web applications*. 2025. Acesso em: 22 de agosto de 2025. Disponível em: <<https://github.com/vaadin/flow>>.

VERTABELO. *Web based data modeler for Professionals*. 2025. Acesso em: 29 de julho de 2025. Disponível em: <<https://vertabelo.com/>>.

VITE. *A blazing fast frontend build tool powering the next generation of web applications*. 2025. Acesso em: 30 de julho de 2025. Disponível em: <<https://vite.dev/>>.

XIAOPING, F.; FANQI, W.; YAN, Z. Design and implementation of graduation thesis management system. In: . [s.n.], 2017. Disponível em: <<https://doi.org/10.2991/icemc-17.2017.250>>.

ZK. *The best open source Java framework for building enterprise web and mobile apps*. 2025. Acesso em: 30 de julho de 2025. Disponível em: <<https://www.zkoss.org/>>.