

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Pedro Paulo Silva Paiva Araújo

**Projeto e Implementação de um Sistema de
Informação pra Acompanhamento do Percurso
dos Estudantes no PPGCO**

Uberlândia, Brasil

2016

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Pedro Paulo Silva Paiva Araújo

**Projeto e Implementação de um Sistema de Informação
pra Acompanhamento do Percurso dos Estudantes no
PPGCO**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, como parte dos requi-
sitos exigidos para a obtenção título de Ba-
charel em Sistemas de Informação.

Orientador: Rafael Dias Araújo

Universidade Federal de Uberlândia – UFU
Faculdade de Computação
Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2016



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Faculdade de Computação

Av. João Naves de Ávila, nº 2121, Bloco 1A - Bairro Santa Mônica, Uberlândia-MG, CEP
38400-902

Telefone: (34) 3239-4144 - <http://www.portal.facom.ufu.br/> facom@ufu.br



ATA DE DEFESA - GRADUAÇÃO

Curso de Graduação em:	Bacharelado em Sistemas de Informação				
Defesa de:	FACOM31802 – Trabalho de Conclusão de Curso 2				
Data:	14/05/2025	Hora de início:	17:05	Hora de encerramento:	18:08
Matrícula do Discente:	11621BSI242				
Nome do Discente:	Pedro Paulo Silva Paiva Araújo				
Título do Trabalho:	Projeto e Implementação de um Sistema de Informação pra Acompanhamento do Percurso dos Estudantes no PPGCO				
A carga horária curricular foi cumprida integralmente?	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não				

Reuniu-se remotamente, por meio da Plataforma Microsoft Teams, a Banca Examinadora, designada pelo Colegiado do Curso de Graduação em Sistemas de Informação, assim composta: Professores: Dr. Bruno Augusto Nassif Travençolo (FACOM/UFU); Dr. Rodrigo Sanches Miani (FACOM/UFU); Dr. Rafael Dias Araújo (FACOM/UFU), orientador do candidato.

Iniciando os trabalhos, o presidente da mesa, Dr. Rafael Dias Araújo, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao discente a palavra, para a exposição do seu trabalho. A duração da apresentação do discente e o tempo de arguição e resposta foram conforme as normas do curso.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado. Nota [85].

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.



Documento assinado eletronicamente por **Rafael Dias Araújo, Professor(a) do Magistério Superior**, em 14/05/2025, às 18:07, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Bruno Augusto Nassif Travençolo, Professor(a) do Magistério Superior**, em 14/05/2025, às 18:07, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Rodrigo Sanches Miani, Professor(a) do Magistério Superior**, em 14/05/2025, às 18:07, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site
https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **6338783** e o código CRC **B93AE48C**.

Referência: Processo nº 23117.032038/2025-04

SEI nº 6338783

Resumo

A crescente demanda por sistemas que otimizem a gestão de projetos acadêmicos motivou a proposta de uma plataforma integrada, capaz de auxiliar estudantes, orientadores e coordenadores de pós-graduação. O sistema permite aos discentes organizar tarefas e metas, facilitando a interação com docentes e o cumprimento de prazos. Para os orientadores, oferece uma visão centralizada do progresso dos alunos, permitindo *feedback* em tempo real e intervenções mais eficazes. Já os coordenadores podem monitorar o atendimento aos requisitos programáticos, como os marcos regulatórios do PPGCO, que são automaticamente incorporados aos projetos cadastrados. Ao centralizar o acompanhamento de atividades, publicações e revisões, a plataforma busca agilizar processos, reduzir falhas e garantir conformidade com as normas institucionais, melhorando a eficiência na gestão acadêmica e a qualidade das pesquisas desenvolvidas.

Palavras-chave: Desenvolvimento, Sistemas, PPGCO, Gerenciamento, Projetos, Gerenciamento de Projetos, Pós-Graduação, Engenharia de Software.

Lista de ilustrações

Figura 1 – Diagrama de Caso de Uso	23
Figura 2 – Diagrama de Entidade e Relacionamento	40
Figura 3 – Organização de Pastas API	42
Figura 4 – Organização de Pastas WEB	44
Figura 5 – Fluxo de cadastro de Estudantes	45
Figura 6 – Fluxo de cadastro de Marcos Temporais	46
Figura 7 – Fluxo de edição de Marcos Temporais	47
Figura 8 – Listagem de Estudantes	50
Figura 9 – Formulário de Estudantes	51
Figura 10 – Listagem de Professores	52
Figura 11 – Formulário de Professores	53
Figura 12 – Listagem de Publicações	54
Figura 13 – Formulário de Publicações	55
Figura 14 – Listagem de Linhas de Pesquisa	55
Figura 15 – Formulário de Linhas de Pesquisa	56
Figura 16 – Listagem de Marcos Temporais	57
Figura 17 – Formulário de Marcos Temporais	58
Figura 18 – Listagem de Disciplinas	59
Figura 19 – Formulário de Disciplinas	60

Lista de abreviaturas e siglas

API	Application Programming Interface
AWS	Amazon Web Services
CLI	Command Line Interface
CSS	Cascading Style Sheets
FACOM	Faculdade de Computação
HTML	HyperText Markup Language
JS	Javascript
MSC	Model Service Controller
MVC	Model View Controller
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
PPGCO	Programa de Pós-Graduação em Computação
REST	Representational State Transfer
SAPE-PPGCO	Sistema de Acompanhamento do Percurso dos Estudantes do Programa de Pós-Graduação em Computação
SASS	Syntactically Awesome Style Sheets
SNPG	Sistema Nacional de Pós-Graduação
SQL	Structured Query Language
SSG	Static Site Generation
UC	Use Case
UFU	Universidade Federal de Uberlândia
UML	Unified Modeling Language
WWW	World Wide Web

Sumário

1	INTRODUÇÃO	10
1.1	Justificativa	11
1.2	Objetivos	11
1.3	Organização do Texto	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	Programação para Internet	13
2.1.1	Desenvolvimento de Aplicações Web	13
2.2	Arquitetura de Software	14
2.2.1	Arquitetura Cliente-Servidor	14
2.2.1.1	Arquitetura Orientada a Serviços (SOA)	14
2.2.1.2	API RESTful	15
2.2.1.3	Arquitetura MSC (Model-Service-Controller)	15
2.2.1.4	MySQL	16
2.3	JavaScript (JS)	16
2.4	Node.js	17
2.5	NestJS	17
3	TRABALHOS RELACIONADOS	18
4	DESENVOLVIMENTO	21
4.1	Requisitos	22
4.1.1	Requisitos Funcionais	22
4.1.2	Requisitos Não-Funcionais	23
4.1.3	Casos de Uso	23
4.1.3.1	UC01 - Entrar	23
4.1.3.2	UC02 - Ativar Conta	24
4.1.3.3	UC03 - Recuperar Senha	25
4.1.3.4	UC04 - Autenticar	25
4.1.3.5	UC05 - Gerenciar dados de Perfil	26
4.1.3.6	UC06 - Gerenciar Estudantes	26
4.1.3.7	UC07 - Gerenciar Professores	28
4.1.3.8	UC08 - Gerenciar Disciplinas	29
4.1.3.9	UC09 - Gerenciar Marcos Temporais	30
4.1.3.10	UC10 - Gerenciar Marcos Temporais Padrão	31
4.1.3.11	UC11 - Gerenciar Publicações	32

4.1.3.12	UC12 - Gerenciar Linhas de Pesquisa	33
4.1.3.13	UC13 - Gerenciar Notificações	35
4.1.3.14	UC14 - Gerenciar Secretários	36
4.1.3.15	UC15 - Gerenciar Coordenadores	37
4.1.3.16	UC16 - Visualizar Dashboard	38
4.2	Prototipação	38
4.3	Ferramentas e Tecnologias	39
4.3.1	Banco de Dados	39
4.3.2	API	39
4.3.3	WEB	39
4.4	Organização de Pastas e Arquitetura (API)	41
4.5	Organização de Pastas e Arquitetura (WEB)	41
4.6	Diagramas	45
4.6.1	Fluxogramas	45
4.6.1.1	Fluxo de cadastro de estudantes	45
4.6.1.2	Fluxo de cadastro marcos temporais	46
4.6.1.3	Fluxo de Edição marcos temporais	47
4.6.2	Telas do Sistema	47
4.6.2.1	Categorias de Telas	47
4.6.2.1.1	Telas de Listagem	47
4.6.2.1.2	Telas de Formulário	48
4.6.2.1.3	Regras de Validação e Padrões Visuais	48
4.6.2.2	Gerenciamento de Estudantes	48
4.6.2.2.1	Listagem	48
4.6.2.2.2	Cadastro/Edição	49
4.6.2.3	Gerenciamento de Professores	51
4.6.2.3.1	Listagem	51
4.6.2.3.2	Cadastro/Edição	52
4.6.2.4	Gerenciamento de Publicações	53
4.6.2.4.1	Listagem	53
4.6.2.4.2	Cadastro/Edição	54
4.6.2.5	Gerenciamento de Linhas de Pesquisa	55
4.6.2.5.1	Listagem	55
4.6.2.5.2	Cadastro/Edição	56
4.6.2.6	Gerenciamento de Marcos Temporais	56
4.6.2.6.1	Listagem	56
4.6.2.6.2	Cadastro/Edição	57
4.6.2.7	Gerenciamento de Disciplinas	58
4.6.2.7.1	Listagem	58

4.6.2.7.2	Cadastro/Edição	58
5	CONSIDERAÇÕES FINAIS	61
	REFERÊNCIAS	62

1 Introdução

O ensino superior é organizado em dois níveis: a graduação e a pós-graduação. No Brasil, o sistema nacional de pós-graduação (SNPG) gerencia a oferta desses cursos em todo o território nacional, incluindo uma avaliação periódica realizada a cada quatro anos. Essa avaliação analisa a qualidade dos cursos e das instituições de ensino, com base em insumos derivados de dados dos estudantes ([CAPES, 2023](#)).

Atualmente, o processo de coleta e controle desses insumos demanda muito esforço por parte da secretaria e coordenação dos programas de pós-graduação, uma vez que falhas podem comprometer a precisão dos resultados e, consequentemente, a avaliação final ([CAPES, 2023](#)).

O sucesso dos cursos de pós-graduação e o desempenho de seus estudantes dependem não apenas de uma gestão acadêmica eficiente, mas também de uma gestão de projetos bem estruturada. Projetos bem planejados e executados contribuem diretamente para os resultados positivos na avaliação dos cursos, uma vez que impactam a qualidade da produção científica.

A complexidade dos projetos de pesquisa acadêmica, especialmente nos níveis de mestrado e doutorado, requer uma maior organização e gestão eficiente por parte dos envolvidos. Projetos dessa magnitude exigem uma boa gestão de tempo, acompanhamento constante, cumprimento de prazos e colaboração entre diferentes atores do processo acadêmico.

A Universidade Federal de Uberlândia ainda utiliza sistemas fragmentados e processos manuais para essa gestão, o que pode resultar em falhas de comunicação, perda de prazos e, consequentemente, comprometer a qualidade das pesquisas.

Com a alta demanda por plataformas que facilitem esse processo, a criação de um sistema de gerenciamento de projetos acadêmicos surge como uma solução prática e eficiente. Este tipo de plataforma permite aos estudantes organizarem melhor suas tarefas e metas, ao mesmo tempo em que facilita a interação com orientadores e coordenadores.

Além disso, oferece aos professores uma visão centralizada do progresso dos alunos, possibilitando intervenções mais assertivas e pareceres em tempo real. Para os coordenadores, a plataforma auxilia no controle de prazos e requisitos formais dos programas de pós-graduação, garantindo uma maior conformidade com os regulamentos internos e externos.

Os programas de pós-graduação possuem uma série de normas e regulamentos internos que devem ser respeitados para atender às especificidades do programa. Por exem-

plo, no caso do PPGCO, para concluir um de seus cursos de pós-graduação, o discente deve cumprir alguns marcos definidos no [Regulamento Interno do PPGCO](#). O sistema proposto considera os marcos padrão definidos no regulamento e os adiciona automaticamente ao projeto assim que é cadastrado, visando facilitar o trabalho da coordenação e da secretaria e também facilitar o acompanhamento e cumprimento desses marcos por parte do estudante.

A plataforma proposta visa simplificar e centralizar o desenvolvimento e a manutenção de projetos do PPGCO, permitindo que os estudantes possam registrar suas atividades, fazer publicações e que recebam opiniões contínuas dos orientadores. Os professores, por sua vez, podem acompanhar o progresso dos seus orientandos, recebendo alertas sobre prazos e tendo acesso facilitado às revisões necessárias. Além disso, coordenadores de programas de pós-graduação podem monitorar o cumprimento dos requisitos do curso, garantindo que todos os projetos sigam as diretrizes estabelecidas.

1.1 Justificativa

A criação dessa plataforma se justifica pela necessidade de organizar e centralizar o fluxo de trabalho dos envolvidos, viabilizando uma comunicação mais fluida e um acompanhamento mais próximo do progresso acadêmico.

Para os estudantes, isso significa possuir uma ferramenta para a organização pessoal, permitindo a gestão de prazos e tarefas de forma eficiente, o que pode reduzir o estresse e aumentar a produtividade. Para os orientadores e coordenadores, o sistema facilita o monitoramento do desenvolvimento dos projetos, garante o cumprimento das normas e melhora a qualidade do acompanhamento e da supervisão acadêmica.

Além disso, a plataforma proposta visa contribuir para a inovação tecnológica no ambiente acadêmico, oferecendo uma solução que atende às demandas modernas de digitalização e automação de processos. Dessa forma, a justificativa deste projeto está baseada em sua capacidade de atender às necessidades práticas do cotidiano acadêmico, facilitando a experiência de todos os envolvidos e promovendo a excelência nas atividades de pesquisa e orientação.

1.2 Objetivos

Projetar e desenvolver uma solução computacional para controle e gestão de trabalhos de pós-graduação, criando um sistema de gerenciamento de projetos. Para tal, o sistema possui uma versão iOS ([PAULA, 2024](#)), uma versão Web e também possui um Banco de Dados Relacional e uma Interface de Programação de Aplicativos (API). Este trabalho abordará o desenvolvimento da aplicação WEB, do Banco de Dados e da API.

A aplicação teve como propósito oferecer ao coordenador um ambiente que possibilasse o cadastro e a gestão de projetos, alunos e orientadores, além da definição de marcos (prazos e entregas) e o acompanhamento dos resultados. O sistema permitia que os alunos realizassem publicações científicas e gerissem seus dados pessoais, enquanto os orientadores podiam acompanhar o progresso dos alunos e prestar apoio no desenvolvimento dos projetos.

O sistema tem como premissa atender às necessidades dos indivíduos envolvidos, separando-os em papéis - Coordenador, Professores, Secretários e Estudantes. A aplicação WEB está focada na visão do Coordenador e dos Secretários; entretanto, a visão de Estudantes e Professores pode ser trabalhada em projetos futuros.

Especificamente, busca-se:

- Facilitar o monitoramento do progresso dos projetos pelos orientadores, oferecendo uma visão clara e atualizada das atividades dos alunos.
- Automatizar o controle de prazos e envio de notificações para garantir o cumprimento dos cronogramas acadêmicos.
- Proporcionar aos estudantes uma ferramenta para gestão de tarefas, submissão de documentos e comunicação direta com os orientadores.
- Oferecer aos coordenadores de programas de pós-graduação um sistema para monitorar o andamento geral dos projetos, assegurando que as exigências acadêmicas e institucionais sejam atendidas.

Assim, o projeto visa contribuir para a eficiência do processo de gestão acadêmica, promovendo uma experiência mais integrada e organizada para todos os envolvidos.

1.3 Organização do Texto

O Capítulo 2 contempla assuntos e conceitos necessários para o entendimento deste trabalho; o Capítulo 3 traz alguns trabalhos que abordam contextos relacionados com o abordado neste trabalho; o Capítulo 4 detalha todo o processo de desenvolvimento da proposta idealizada; por fim, no Capítulo 5 são apresentadas as considerações finais sobre o trabalho.

2 Fundamentação teórica

A fundamentação teórica deste trabalho está baseada em conceitos de gestão de projetos, ambientes colaborativos e tecnologias aplicadas à educação. A plataforma proposta integra princípios de gerenciamento de projetos, como controle de cronogramas, acompanhamento de tarefas e gestão de recursos, adaptados ao contexto acadêmico de mestrado e doutorado.

2.1 Programação para Internet

A programação para internet engloba técnicas de desenvolvimento voltadas à rede mundial de computadores, permitindo acesso a serviços remotos e comunicação entre aplicações e usuários geograficamente distribuídos.

Seu desenvolvimento iniciou-se na década de 1980, porém ganhou maior impulso em 1989 com a criação da *World Wide Web (WWW)* por Tim Berners-Lee, que estabeleceu padrões para distribuição de documentos hipertexto através do protocolo HTTP ([ROCK CONTENT, 2020](#)). Atualmente, a programação para internet abrange diversas modalidades, incluindo desenvolvimento Mobile e *IoT* ([SERVICES, N.D.a](#)).

2.1.1 Desenvolvimento de Aplicações Web

O desenvolvimento Web utiliza os conceitos fundamentais da programação para a internet, operando principalmente sobre a WWW. Consiste em técnicas para criação e disponibilização de aplicações que executam na internet mediante o protocolo HTTP ([SOUZA, 2019](#)). Sua evolução pode ser dividida em:

- Primeira geração: páginas estáticas em HTML
- Gerações subsequentes: incorporação de dinamismo e interatividade

Atualmente, o desenvolvimento Web suporta protocolos avançados como *WebSocket*, que mantém conexões persistentes entre cliente e servidor ([SCHMITZ, 2021](#)). Essa tecnologia possibilitou o surgimento de:

- Sistemas de chat em tempo real
- Jogos online multijogador
- Aplicações colaborativas

2.2 Arquitetura de Software

Arquitetura de software refere-se ao modelo estrutural que define a organização de um sistema, antecipando problemas através de modularização, padronização e definição clara de fluxos e responsabilidades (GONÇALVES, 2021). A aplicação proposta neste trabalho adotará:

- Arquitetura de rede Cliente-Servidor
- Padrão Orientado a Serviços (SOA)
- *API RESTful* com estrutura *Model-Service-Controller*

2.2.1 Arquitetura Cliente-Servidor

Modelo fundamental da rede TCP/IP que estabelece:

- **Servidor:** provê recursos ou serviços continuamente
- **Cliente:** acessa os recursos de forma esporádica

O servidor requer um endereço IP público e, preferencialmente, um nome de domínio para facilitar o acesso (NET, 2023).

2.2.1.1 Arquitetura Orientada a Serviços (SOA)

Padrão arquitetural que organiza a camada de negócios em serviços independentes, com as seguintes características:

- Separação clara entre *backend* e *frontend*
- Alto grau de abstração
- Baixo acoplamento
- Interoperabilidade

Normalmente implementada como *API REST*, permite que múltiplos clientes (Web, Mobile) consumam os mesmos serviços (SERVICES, N.D.b). Representa uma evolução do modelo monolítico, onde *frontend* e *backend* eram partes integrantes da mesma aplicação.

Como desenvolvimento posterior ao SOA, surgiu a arquitetura de microsserviços, caracterizada por:

- Serviços mais especializados

- Escopo reduzido
- Menor granularidade

([SUPERO, N.D.](#))

2.2.1.2 API RESTful

Interface de programação que segue os princípios REST (*Representational State Transfer*), utilizando HTTP como protocolo subjacente. Suas características principais incluem:

- Comunicação baseada em representação de estado
- Independência de tecnologias
- Stateless por natureza

([SCANDIFFIO, 2017](#); [HAT, 2023](#))

2.2.1.3 Arquitetura MSC (Model-Service-Controller)

Padrão derivado do MVC, organizando o sistema em três camadas principais:

Tabela 1 – Camadas da Arquitetura MSC

Camada	Responsabilidade
Model	Acesso e manipulação de dados
Service	Regras de negócio
Controller	Comunicação com clientes

Vantagens desta abordagem:

- Melhor organização do código
- Facilidade de manutenção
- Segurança aprimorada
- Desenvolvimento em equipe facilitado

Adotada por diversos frameworks como Laravel, NestJS, .NET, Spring e Django ([MOHR, 2022](#)).

2.2.1.4 MySQL

Sistema gerenciador de banco de dados relacional (SGBDR) de código aberto, desenvolvido originalmente na Suécia por David Axmark, Allan Larsson e Michael Widenius ([PACIEVITCH, N.D.](#)).

Principais características:

- Alta popularidade e adoção (Facebook, YouTube, Netflix)
- Escalabilidade para grandes volumes
- Flexibilidade de implantação (nuvem, distribuído)
- Comunidade ativa e robusta

([ORACLE, N.D.](#); [CORADIN, 2023](#))

2.3 JavaScript (JS)

Linguagem de programação multiparadigma que se tornou fundamental para o desenvolvimento web moderno, suportando:

- Programação orientada a objetos
- Programação funcional
- Abordagens procedural e declarativa

([SCANDIFFIO, 2017](#); [PEREIRA, 2023](#))

Padronizada como ECMAScript (atualmente na versão 2022 ([CONTRIBUTORS, 2023](#))), é a linguagem por trás de conceitos como:

- *Single Page Applications* (SPAs)
- Aplicações isomórficas
- Desenvolvimento full-stack

([FERREIRA, 2023](#); [CONTRIBUTORS, 2022](#))

Adotada por grandes plataformas como Instagram, Facebook, Uber ([LTDA, N.D.](#)), Airbnb ([AIRBNB, N.D.](#)), e Netflix ([NOLETO, N.D.](#)).

2.4 Node.js

Ambiente de execução JavaScript que permite sua utilização fora dos navegadores, possibilitando:

- Desenvolvimento de *backend*
- Criação de ferramentas CLI
- Automação de processos

2.5 NestJS

Framework para desenvolvimento de *backend* em JavaScript/TypeScript que oferece:

- Estruturação baseada em arquitetura MSC
- Facilidades para criação de APIs REST
- Conjunto abrangente de ferramentas para desenvolvedores

3 Trabalhos Relacionados

Neste tópico serão abordados os projetos ou soluções, científicas ou não, que contemplam problemas semelhantes ou complementares ao do trabalho em tela. Boa parte das produções citadas abaixo, foram encontradas através do Google Acadêmico¹ e do Repositório Institucional UFU². Outras, foram localizadas nas referências de outros trabalhos relacionados e se mostraram relevantes.

Uma das produções encontradas é o “Projeto e Implementação de um Aplicativo iOS para Apoio à Jornada Acadêmica dos Estudantes no PPGCO” ([PAULA, 2024](#)). O referido software é complementar ao discutido no presente trabalho, por se tratar de uma versão mobile iOS, com o diferencial de ter sido construído para o uso do discente.

O “Controle universitário: um aplicativo de controle dos estudos e do andamento do graduando ao longo do curso superior” foi referenciado no estudo acima mencionado. O projeto se propõe a desenvolver uma aplicação para o gerenciamento das próprias atividades, permitindo ao usuário cadastrar e gerenciar seus trabalhos, seminários e provas, vinculando-os às suas respectivas disciplinas. O sistema também notifica o usuário de seus prazos, facilitando o controle e o desenvolvimento pessoal do estudante ([PAULA, 2024](#)).

O “Sistema de controle de prazos para os alunos de pós-graduação” foi encontrado através do repositório da UFU ([SCANDIFFIO, 2017](#)). A plataforma se assemelha ao *software* proposto por esse trabalho em dois aspectos: o primeiro é que ambos se propõem a resolver problemas do PPGCO; o segundo é o fato de que uma das funcionalidades do sistema de proposto é o controle dos prazos.

O objetivo do trabalho de [Scandiffio \(2017\)](#) desenvolveu uma aplicação web onde secretários pudesse alertar alunos e orientadores dos prazos de vencimento de suas atividades, usando alertas automáticos e pré-configurados. De maneira semelhante, o SAPE-PPGCO permitirá ao coordenador definir os marcos temporais referentes às entregas do projeto e editá-los, por sua vez, os estudantes serão notificados desses prazos via e-mail.

No repositório da UFU também foi descoberto o projeto “Desenvolvimento do painel de inscrição para o Programa de Pós-graduação em Ciência da Computação” ([ALVES, 2023](#)), que é complementar a este, por se tratar de um sistema de utilização dos estudantes e da coordenação do PPGCO.

O contexto de utilização desse aplicativo web é o mesmo do sistema proposto nesse trabalho, entretanto ele resolve um problema distinto. O trabalho possui como principal objetivo o desenvolvimento de uma plataforma para inscrição e submissão da documen-

¹ <https://scholar.google.com/>

² <https://repositorio.ufu.br/>

tação necessária prevista nos editais de ingresso no PPGCO. Esse sistema visa organizar e automatizar o processo de inscrição do Programa de Pós-Graduação em Computação da UFU ([ALVES, 2023](#)).

Outro trabalho encontrado foi o "Sistema para gerenciamento de projetos de *Software*" ([MENDES; PEREIRA; SOUZA, 2010](#)), que descreve o desenvolvimento do Sistema para Gerenciamento de Projetos de *Software* (SGPS) destaca como seu principal objetivo auxiliar na gestão de projetos de *software*, abrangendo aspectos como prazos, comunicação entre os envolvidos, definição de objetivos e controle das tarefas.

O SAPE-PPGCO apresenta semelhanças com o SGPS apenas no que diz respeito à sua natureza de gerenciador de projetos, uma vez que os contextos de utilização de cada sistema são distintos.

O “Sistema para Gerenciamento de Projetos de *Software*” (SGPS) ([MENDES; PEREIRA; SOUZA, 2010](#)) é outro estudo correlato, uma vez que descreve o SGPS como uma solução focada no gerenciamento integral de projetos de *software*, incluindo controle de prazos, comunicação, definição de objetivos e monitoramento de tarefas. Embora o SAPE-PPGCO também seja um gerenciador de projetos, sua aplicação difere significativamente em contexto e escopo quando comparado ao SGPS.

Além dos trabalhos citados, existem diversas ferramentas de gerenciamento de projetos disponíveis no mercado, como *Trello*³, *ClickUp*⁴, *Monday.com*⁵ e *Jira*⁶. Essas plataformas inspiraram a concepção do Sistema de Acompanhamento do Percurso dos Alunos no PPGCO (SAPE-PPGCO), especialmente em funcionalidades como:

- Gestão colaborativa com múltiplos integrantes
- Monitoramento de prazos e metas
- Divisão de projetos em tarefas menores com priorização
- Definição de submetas para conclusão de atividades

Dentre essas soluções, o Jira merece destaque por seu enfoque específico no desenvolvimento de software utilizando metodologias ágeis, como *Scrum*.

Outra categoria relevante inclui ferramentas como Cronograma, *Microsoft Project* e *Project Libre*, que utilizam abordagens baseadas em tabelas para gerenciar:

- Duração de atividades

³ <https://trello.com>

⁴ <https://clickup.com>

⁵ <https://monday.com/lang/pt>

⁶ <https://www.atlassian.com/software/jira>

- Custos envolvidos
- Baseline de projetos
- Alocação de recursos

Existem também gerenciadores como *Cronograma*, *Microsoft Project* e *Project Libre* que gerenciam os projetos por meio de tabelas, com campos de duração, custo e *baselines*. Esses *softwares* tratam as pessoas envolvidas como recursos e focam na alocação eficiente desses recursos. Tais funcionalidades diferem da proposta do SAPE-PPGCO, visto que apenas um aluno desenvolverá o projeto, enquanto os demais papéis envolvidos terão uma participação significativamente menos ativa.

No entanto, a definição de duração e *baselines* pode ser relevante para garantir o cumprimento do planejamento e manter um histórico das alterações realizadas. Os sistemas mencionados incorporam os conceitos do PMBOK, que consiste em um guia e conjunto de boas práticas em gerenciamento de projetos, organizado pelo *Project Management Institute (PMI)* ([BRAULIO-GONZALO et al., 2021](#)).

4 Desenvolvimento

Este capítulo discorre sobre o desenvolvimento do projeto SAPE-PPGCO. Para o desenvolvimento da aplicação, o projeto foi dividido em etapas: a primeira etapa foi o levantamento de requisitos; a segunda foi o planejamento e prototipação; a terceira foi a implementação da solução.

Para atingir os objetivos propostos, fez-se necessário planejar o projeto e seguir determinados passos listados a baixo:

- Coleta e análise de requisitos funcionais e não funcionais, embasados nos regulamentos internos do Programa de Pós-Graduação da FACOM.
- Modelagem do banco de dados considerando as entidades presumidas pela coleta de requisitos (Diagrama *EER - Enhanced Entity-Relationship*) ([ELMASRI; NAVATHE, 2019](#)).
- Modelagem dos requisitos em diagramas da UML tanto para o *backend*, quanto para o *frontend*.
 - Construção do diagrama de Caso de Uso, para especificar os requisitos funcionais de maneira a conceber um contexto de utilização, bem como seus atores.
 - Construção de um modelo conceitual do domínio do problema, para visualização do problema de maneira abstrata e alto nível.
 - Construção de diagrama de estados, para especificar os estados da aplicação conforme a utilização.
 - Construção de diagrama de classes, para definir as classes do problema já considerando arquitetura, dependências e os requisitos não funcionais
- Construção de um protótipo iterativo da aplicação, para reunir as ideias obtidas da etapa anterior com uma ideia mais concreta de como será a aplicação. Nessa etapa, serão explorados conceitos de Interação Humano-Computador, para garantir uma melhor experiência de usuário.
- Validar o modelo proposto, em termos de protótipo e dos diagramas feitos. Caso necessário, refazer as etapas anteriores a fim de refinar o projeto e de melhorar o processo de engenharia de *software*.
- Construir banco de dados de acordo com o que foi modelado nas etapas iniciais do processo de engenharia de *software*.

- Implementar *REST API*, responsável por conter a regra de negócio, manipular e armazenar os dados da aplicação e de usuários.
- Implementar aplicação web *frontend* com base no que foi desenvolvido na etapa de prototipação, incorporando a integração com a *REST API* desenvolvida.
- Implementar estrutura de *Jobs*, para notificar os envolvidos sobre os prazos na aplicação e via e-mail.

4.1 Requisitos

Para a coleta de requisitos, foram realizadas reuniões com o Orientador e com os demais envolvidos. Nessa etapa, foi definido o que o sistema deveria ser capaz de fazer, em quais ambientes deveria funcionar e quem seriam os usuários. Por conta dos prazos e de recursos limitados, este trabalho ficou responsável pelo desenvolvimento do sistema administrativo na versão WEB, também com a API e o banco de dados utilizados nas demais aplicações.

4.1.1 Requisitos Funcionais

O SAPE-PPGCO deve ser capaz de:

- Realizar o auto-cadastro de estudantes;
- Editar informações de Perfil (e-mail, lattes, endereço);
- Listar, Pré-Cadastrar, Editar e Deletar Estudantes;
- Listar, Pré-Cadastrar, Editar e Deletar Professores;
- Listar, Cadastrar, Editar e Deletar disciplinas;
- Listar, Cadastrar, Editar e Deletar marcos temporais;
- Listar, Cadastrar, Editar e Deletar marcos temporais padrão;
- Listar, Cadastrar, Editar e Deletar publicações;
- Listar, Cadastrar, Editar e Deletar notificações;
- Permitir o usuário manipular suas informações pessoais editáveis;
- Visualizar estatísticas.

4.1.2 Requisitos Não-Funcionais

Os requisitos não funcionais levantados foram:

- A API deverá ser executada em ambiente Linux.
- A aplicação frontend WEB deverá executar nos navegadores mais populares (Google Chrome, Mozilla Firefox, Brave, Opera), independente do sistema operacional.
- A aplicação WEB deverá ser responsiva e adaptar-se aos diferentes tamanhos de tela (Com foco no ambiente desktop).
- Os componentes utilizados devem ser padronizados.
- Ocultar e tratar falhas.

4.1.3 Casos de Uso

A Figura 1 mostra o Diagrama de Casos de Uso do sistema proposto. Observa-se que há apenas um ator: Coordenador/secretário do PPG. Cada caso de uso é detalhado nas subseções a seguir.

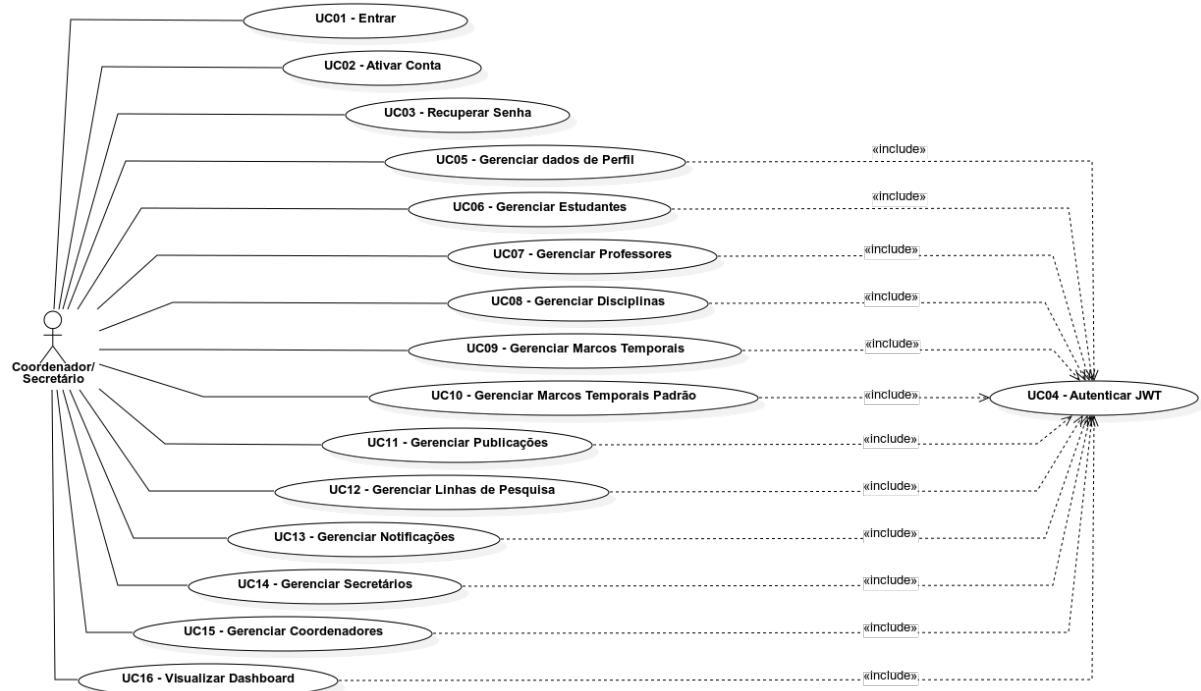


Figura 1 – Diagrama de Caso de Uso

4.1.3.1 UC01 - Entrar

- Ator: Coordenador/Secretário

- Descrição: O coordenador/secretário devem poder fazer login no sistema.

Fluxo Principal

1. O usuário acessa a página de login
2. Inserir *e-mail*
3. Inserir senha
4. Solicitar autenticação
 - 4.1 O sistema valida o *e-mail* inserido
 - 4.2 O sistema valida o senha inserida
5. O sistema gera um *token* de acesso (JWT)
6. O sistema armazena o *token* de acesso na máquina do usuário
7. O usuário é redirecionado à página principal do sistema.

Fluxo de Exceção

4.1.1 O e-mail inserido não corresponde a nenhum e-mail cadastrado no sistema.

4.1.1.1 Uma mensagem de login não autorizado é exibida

4.2.1 A senha não corresponde com a senha cadastrada para aquele usuário no sistema.

4.2.1.1 Uma mensagem de login não autorizado é exibida

4.1.3.2 UC02 - Ativar Conta

- Ator: Coordenador/Secretário
- Descrição: O coordenador/secretário recebem um e-mail de ativação e poder fazer a ativação da conta no sistema.

Fluxo Principal

1. O usuário acessa o link de ativação recebido por e-mail
2. O sistema valida a conta
3. O sistema marca a conta como ativada
4. O sistema atualiza a data de ativação da conta

4.1.3.3 UC03 - Recuperar Senha

- Ator: Coordenador/Secretário
- Descrição: O coordenador/secretário devem poder recuperar a conta, caso esqueçam a senha no sistema.

Fluxo Principal

1. O usuário acessa a página de recuperação de conta
2. Inserir o e-mail vinculado a conta
 - 2.1 O sistema verifica se o e-mail inserido está cadastrado no sistema
3. O sistema gera um link de redefinição de senha
4. O sistema salva o link de redefinição de senha vinculando-o ao usuário solicitante
5. O sistema envia um e-mail contendo o link de redefinição de senha
6. O usuário acessa o link enviado
7. O usuário insere a nova senha
8. O usuário confirma a nova senha
9. O sistema salva a senha atualizada
10. O sistema desativa o link de alteração de senha enviado por *e-mail*

4.1.3.4 UC04 - Autenticar

- Ator: Coordenador/Secretário
- Descrição: O coordenador/secretário devem poder se manter autenticado no sistema por um período de tempo, sem precisar fazer login a cada requisição.

Fluxo Principal

1. O usuário solicita alguma operação restrita no sistema
2. O sistema recupera o *token* de acesso armazenado na máquina do usuário, após o login
3. O sistema valida o *token*

4. O sistema libera o acesso ao módulo em questão
5. O sistema gera um novo *token* de acesso

Fluxo de Exceção

3.1 O token de acesso expirou

- 3.1.1 O sistema exibe uma mensagem que error
- 3.1.2 O sistema redireciona o usuário para a tela de login

3.1 O token de acesso é inválido

- 3.1.1 O sistema exibe uma mensagem que error
- 3.1.2 O sistema redireciona o usuário para a tela de login

4.1.3.5 UC05 - Gerenciar dados de Perfil

- Ator: Coordenador/Secretário
- Descrição: O coordenador/secretário devem poder editar seus dados cadastrais no sistema.

Fluxo Principal

1. O usuário acessa a página de perfil
2. Atualizar os dados desejados
3. O sistema salva os dados atualizados do usuário de acordo com o formulário

4.1.3.6 UC06 - Gerenciar Estudantes

- Ator: Coordenador/Secretário
- Descrição: O coordenador/secretário devem poder listar, cadastrar, editar e deletar estudantes cadastrados no sistema.
- Pré-requisitos: UC06 - Ativar Conta, UC10 - Autenticar

Fluxo Principal

(Inserção)

1. O coordenador/secretário acessa a página de estudantes
2. O sistema exibe a lista de estudantes
3. Abrir o formulário de cadastro de estudantes
4. Inserir os dados do novo registro
5. Salvar os dados inseridos no formulário
6. Retornar a tela de listagem de estudantes

(Edição)

1. O coordenador/secretário acessa a página de estudantes
2. O sistema exibe a lista de estudantes
3. Pesquisar o estudante a editar
4. Abrir o formulário de edição do estudante selecionado
5. Atualizar os dados do estudante
6. Salvar os dados atualizados do formulário
7. Retornar a tela de listagem de estudantes

(Remoção)

1. O coordenador/secretário acessa a página de estudantes
2. O sistema exibe a lista de estudantes
3. Pesquisar o estudante a deletar
4. O sistema solicita a confirmação da ação
5. Confirma a exclusão
6. Exclui o estudante e seus registros relacionados em outras tabelas
7. O sistema atualiza a lista de estudantes

4.1.3.7 UC07 - Gerenciar Professores

- Ator: Coordenador/Secretário
- Descrição: O coordenador/secretário devem poder listar, cadastrar, editar e deletar professores cadastrados no sistema.
- Pré-requisitos: UC06 - Ativar Conta, UC10 - Autenticar

Fluxo Principal

(Inserção)

1. O coordenador/secretário acessa a página de professores
2. O sistema exibe a lista de professores
3. Abrir o formulário de cadastro de professores
4. Inserir os dados do novo registro
5. Salvar os dados inseridos no formulário
6. Retornar a tela de listagem de professores

(Edição)

1. O coordenador/secretário acessa a página de professores
2. O sistema exibe a lista de professores
3. Pesquisar o professor a editar
4. Abrir o formulário de edição do professor selecionado
5. Atualizar os dados do professor
6. Salvar os dados atualizados do formulário
7. Retornar a tela de listagem de professores

(Remoção)

1. O coordenador/secretário acessa a página de professores
2. O sistema exibe a lista de professores
3. Pesquisar o professor a deletar

4. O sistema solicita a confirmação da ação
5. Confirma a exclusão
6. Exclui o professor e seus registros relacionados em outras tabelas
7. O sistema atualiza a lista de professores

4.1.3.8 UC08 - Gerenciar Disciplinas

- Ator: Coordenador/Secretário
- Descrição: O coordenador/secretário devem poder listar, cadastrar, editar e deletar disciplinas cadastrados no sistema.
- Pré-requisitos: UC06 - Ativar Conta, UC10 - Autenticar

Fluxo Principal

(Inserção)

1. O coordenador/secretário acessa a página de disciplinas
2. O sistema exibe a lista de disciplinas
3. Abrir o formulário de cadastro de disciplinas
4. Inserir os dados do novo registro
5. Salvar os dados inseridos no formulário
6. Retornar a tela de listagem de disciplinas

(Edição)

1. O coordenador/secretário acessa a página de disciplinas
2. O sistema exibe a lista de disciplinas
3. Pesquisar a disciplina a editar
4. Abrir o formulário de edição da disciplina selecionada
5. Atualizar os dados da disciplina
6. Salvar os dados atualizados do formulário
7. Retornar a tela de listagem de disciplinas

(Remoção)

1. O coordenador/secretário acessa a página de disciplinas
2. O sistema exibe a lista de disciplinas
3. Pesquisar a disciplina a deletar
4. O sistema solicita a confirmação da ação
5. Confirma a exclusão
6. Exclui a disciplina e seus registros relacionados em outras tabelas
7. O sistema atualiza a lista de disciplinas

4.1.3.9 UC09 - Gerenciar Marcos Temporais

- Ator: Coordenador/Secretário
- Descrição: O coordenador/secretário devem poder listar, cadastrar, editar e deletar marcos temporais cadastrados no sistema.
- Pré-requisitos: UC06 - Ativar Conta, UC10 - Autenticar

Fluxo Principal

(Inserção)

1. O coordenador/secretário acessa a página de marcos temporais
2. O sistema exibe a lista de marcos temporais
3. Abrir o formulário de cadastro de marcos temporais
4. Inserir os dados do novo registro
5. Salvar os dados inseridos no formulário
6. Retornar a tela de listagem de marcos temporais

(Edição)

1. O coordenador/secretário acessa a página de marcos temporais
2. O sistema exibe a lista de marcos temporais
3. Pesquisar o marco temporal a editar

4. Abrir o formulário de edição do marco temporal selecionado
5. Atualizar os dados do marco temporal
6. Salvar os dados atualizados do formulário
7. Retornar a tela de listagem de marcos temporais

(Remoção)

1. O coordenador/secretário acessa a página de marcos temporais
2. O sistema exibe a lista de marcos temporais
3. Pesquisar o marco temporal a deletar
4. O sistema solicita a confirmação da ação
5. Confirma a exclusão
6. Exclui o marco temporal e seus registros relacionados em outras tabelas
7. O sistema atualiza a lista de marcos temporais

4.1.3.10 UC10 - Gerenciar Marcos Temporais Padrão

- Ator: Coordenador/Secretário
- Descrição: O coordenador/secretário devem poder listar, cadastrar, editar e deletar marcos temporais padrão cadastrados no sistema.
- Pré-requisitos: UC06 - Ativar Conta, UC10 - Autenticar

Fluxo Principal

(Inserção)

1. O coordenador/secretário acessa a página de marcos temporais padrão
2. O sistema exibe a lista de marcos temporais padrão
3. Abrir o formulário de cadastro de marcos temporais padrão
4. Inserir os dados do novo registro
5. Salvar os dados inseridos no formulário
6. Retornar a tela de listagem de marcos temporais padrão

(Edição)

1. O coordenador/secretário acessa a página de marcos temporais padrão
2. O sistema exibe a lista de marcos temporais padrão
3. Pesquisar o marco temporal padrão a editar
4. Abrir o formulário de edição do marco temporal padrão selecionado
5. Atualizar os dados do marco temporal padrão
6. Salvar os dados atualizados do formulário
7. Retornar a tela de listagem de marcos temporais padrão

(Remoção)

1. O coordenador/secretário acessa a página de marcos temporais padrão
2. O sistema exibe a lista de marcos temporais padrão
3. Pesquisar o marco temporal padrão a deletar
4. O sistema solicita a confirmação da ação
5. Confirma a exclusão
6. Exclui o marco temporal padrão
7. O sistema atualiza a lista de marcos temporais padrão

4.1.3.11 UC11 - Gerenciar Publicações

- Ator: Coordenador/Secretário
- Descrição: O coordenador/secretário devem poder listar, cadastrar, editar e deletar publicações cadastrados no sistema.
- Pré-requisitos: UC06 - Ativar Conta, UC10 - Autenticar

Fluxo Principal

(Inserção)

1. O coordenador/secretário acessa a página de publicações
2. O sistema exibe a lista de publicações

3. Abrir o formulário de cadastro de publicações
4. Inserir os dados do novo registro
5. Salvar os dados inseridos no formulário
6. Retornar a tela de listagem de publicações

(Edição)

1. O coordenador/secretário acessa a página de publicações
2. O sistema exibe a lista de publicações
3. Pesquisar a publicação a editar
4. Abrir o formulário de edição da publicação selecionada
5. Atualizar os dados da publicação
6. Salvar os dados atualizados do formulário
7. Retornar a tela de listagem de publicações

(Remoção)

1. O coordenador/secretário acessa a página de publicações
2. O sistema exibe a lista de publicações
3. Pesquisar a publicação a deletar
4. O sistema solicita a confirmação da ação
5. Confirma a exclusão
6. Exclui a publicação
7. O sistema atualiza a lista de publicações

4.1.3.12 UC12 - Gerenciar Linhas de Pesquisa

- Ator: Coordenador/Secretário
- Descrição: O coordenador/secretário devem poder listar, cadastrar, editar e deletar linhas de pesquisa cadastrados no sistema.
- Pré-requisitos: UC06 - Ativar Conta, UC10 - Autenticar

Fluxo Principal

(Inserção)

1. O coordenador/secretário acessa a página de linhas de pesquisa
2. O sistema exibe a lista de linhas de pesquisa
3. Abrir o formulário de cadastro de linhas de pesquisa
4. Inserir os dados do novo registro
5. Salvar os dados inseridos no formulário
6. Retornar a tela de listagem de linhas de pesquisa

(Edição)

1. O coordenador/secretário acessa a página de linhas de pesquisa
2. O sistema exibe a lista de linhas de pesquisa
3. Pesquisar a linha de pesquisa a editar
4. Abrir o formulário de edição da linha de pesquisa selecionada
5. Atualizar os dados da linha de pesquisa
6. Salvar os dados atualizados do formulário
7. Retornar a tela de listagem de linhas de pesquisa

(Remoção)

1. O coordenador/secretário acessa a página de linhas de pesquisa
2. O sistema exibe a lista de linhas de pesquisa
3. Pesquisar a linha de pesquisa a deletar
4. O sistema solicita a confirmação da ação
5. Confirma a exclusão
6. Exclui a linha de pesquisa
7. O sistema atualiza a lista de linhas de pesquisa

4.1.3.13 UC13 - Gerenciar Notificações

- Ator: Coordenador/Secretário
- Descrição: O coordenador/secretário devem poder listar, cadastrar, editar e deletar notificações cadastrados no sistema.
- Pré-requisitos: UC06 - Ativar Conta, UC10 - Autenticar

Fluxo Principal

(Inserção)

1. O coordenador/secretário acessa a página de notificações
2. O sistema exibe a lista de notificações
3. Abrir o formulário de cadastro de notificações
4. Inserir os dados do novo registro
5. Salvar os dados inseridos no formulário
6. Retornar a tela de listagem de notificações

(Edição)

1. O coordenador/secretário acessa a página de notificações
2. O sistema exibe a lista de notificações
3. Pesquisar a notificação a editar
4. Abrir o formulário de edição da notificação selecionada
5. Atualizar os dados da notificação
6. Salvar os dados atualizados do formulário
7. Retornar a tela de listagem de notificações

(Remoção)

1. O coordenador/secretário acessa a página de notificações
2. O sistema exibe a lista de notificações
3. Pesquisar a notificação a deletar

4. O sistema solicita a confirmação da ação
5. Confirma a exclusão
6. Exclui a notificação
7. O sistema atualiza a lista de notificações

4.1.3.14 UC14 - Gerenciar Secretários

- Ator: Coordenador/Secretário
- Descrição: O coordenador/secretário devem poder listar, cadastrar, editar e deletar secretários cadastrados no sistema.
- Pré-requisitos: UC06 - Ativar Conta, UC10 - Autenticar

Fluxo Principal

(Inserção)

1. O coordenador/secretário acessa a página de secretários
2. O sistema exibe a lista de secretários
3. Abrir o formulário de cadastro de secretários
4. Inserir os dados do novo registro
5. Salvar os dados inseridos no formulário
6. Retornar a tela de listagem de secretários

(Edição)

1. O coordenador/secretário acessa a página de secretários
2. O sistema exibe a lista de secretários
3. Pesquisar o secretário a editar
4. Abrir o formulário de edição do secretário selecionado
5. Atualizar os dados do secretário
6. Salvar os dados atualizados do formulário
7. Retornar a tela de listagem de secretários

(Remoção)

1. O coordenador/secretário acessa a página de secretários
2. O sistema exibe a lista de secretários
3. Pesquisar o secretário a deletar
4. O sistema solicita a confirmação da ação
5. Confirma a exclusão
6. Exclui o secretário e seus registros relacionados em outras tabelas
7. O sistema atualiza a lista de secretários

4.1.3.15 UC15 - Gerenciar Coordenadores

- Ator: Coordenador/Secretário
- Descrição: O coordenador/secretário devem poder listar, cadastrar, editar e deletar coordenadores cadastrados no sistema.
- Pré-requisitos: UC06 - Ativar Conta, UC10 - Autenticar

Fluxo Principal

(Inserção)

1. O coordenador/secretário acessa a página de coordenadores
2. O sistema exibe a lista de coordenadores
3. Abrir o formulário de cadastro de coordenadores
4. Inserir os dados do novo registro
5. Salvar os dados inseridos no formulário
6. Retornar a tela de listagem de coordenadores

(Edição)

1. O coordenador/secretário acessa a página de coordenadores
2. O sistema exibe a lista de coordenadores
3. Pesquisar o coordenador a editar

4. Abrir o formulário de edição do coordenador selecionado

5. Atualizar os dados do coordenador

6. Salvar os dados atualizados do formulário

7. Retornar a tela de listagem de coordenadores

(Remoção)

1. O coordenador/secretário acessa a página de coordenadores

2. O sistema exibe a lista de coordenadores

3. Pesquisar o coordenador a deletar

4. O sistema solicita a confirmação da ação

5. Confirma a exclusão

6. Exclui o coordenador e seus registros relacionados em outras tabelas

7. O sistema atualiza a lista de coordenadores

4.1.3.16 UC16 - Visualizar Dashboard

- Ator: Coordenador/Secretário
- Descrição: O coordenador/secretário devem poder ver estatísticas de uso do sistema e ser informados a respeito do cumprimento dos prazos por parte dos estudantes

Fluxo Principal

1. Acessar a página de dashboard

2. Os dados são recuperados

3. Os dados recuperados são processados pelo sistema

4. Os dados tratados são exibidos na tela

4.2 Prototipação

A prototipação do projeto foi feita através da plataforma [Figma](#), que é um editor de prototipagem de projetos de design. Pode ser utilizado através de um navegador e possui uma versão gratuita. No Figma, foi possível construir a ideia inicial do aplicativo, usando componentes disponíveis gratuitamente.

4.3 Ferramentas e Tecnologias

4.3.1 Banco de Dados

A modelagem e o desenvolvimento do banco de dados foi realizada utilizando o sistema da Oracle MySQL Workbench¹. Nele, um Diagrama Entidade-Relacionamento (DER) foi construído feito e seu respectivo código SQL foi gerado.

A Figura 19 representa o conjunto de entidades do sistema e seus relacionamentos.

4.3.2 API

A API foi construída usando a ferramenta Node.Js junto com a linguagem de programação TypeScript, para as seguintes funcionalidades

- Validação e sanitização dos formulários foi utilizada a biblioteca [Zod](#), essa biblioteca trata os dados que chegam na API, com o intuito de evitar ataques de SQLInjection, JavaScriptInjection entre outros. Além de garantir uma formatação padrão para os dados a serem armazenados no banco de dados.
- Comunicação com o banco de dados e o mapeamento dos dados recebidos em objetos JavaScript foi utilizada a biblioteca [SequelizeORM](#).
- Ganhos de produtividade no desenvolvimento da API, foi utilizado o *framework* Nest.Js, que possui um ecossistema voltado para a construção de aplicações *backend* com foco em API's Restful.

O projeto da API utiliza o sistema de versionamento GIT e tem seu código hospedado no Github no link <https://github.com/pedro-bigz/api-ppgco-v2>

4.3.3 WEB

A aplicação WEB foi desenvolvida utilizando a biblioteca ReactJs, com o intuito de ganhar produtividade. Junto com o React, outras bibliotecas foram utilizadas, como:

- [Zod](#), é utilizada no *frontend*, mas com o intuito de validar e formatar os campos dos formulários e exibir mensagens de erro apropriadas para os usuários.
- [Axios](#), para fazer a comunicação com a API.
- [NextUI](#), para padronizar os componentes e ganhar produtividade. Essa biblioteca segue as diretrizes e especificações definidas pela [Material Design](#) do Google.

¹ <https://www.mysql.com/products/workbench/>

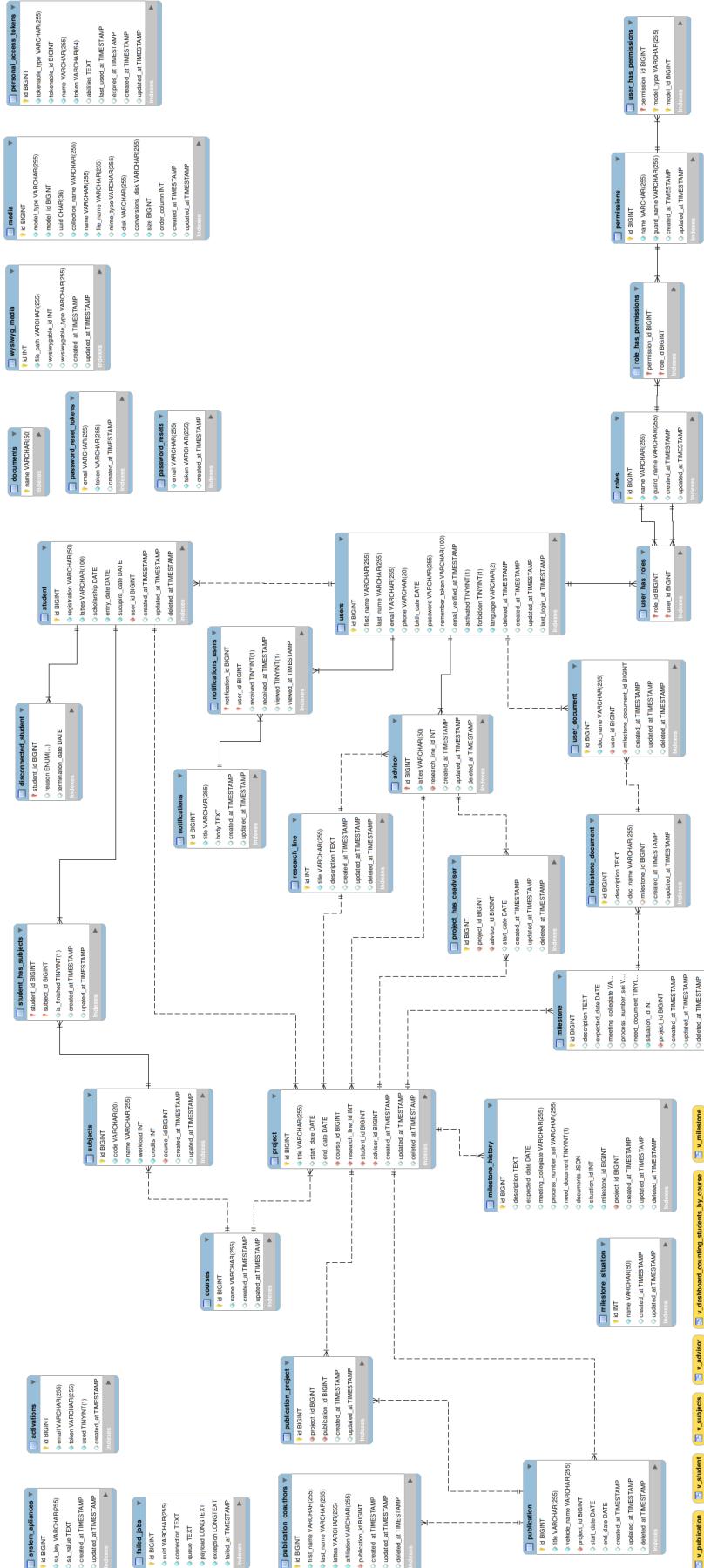


Figura 2 – Diagrama de Entidade e Relacionamento

Visando proteger os arquivos dos usuários, foi desenvolvido um servidor à parte, responsável por armazenar e controlar o acesso a esses arquivos. No entanto, essa solução foi implementada devido à indisponibilidade de outros recursos que seriam importantes para garantir o controle de acesso. Essa ferramenta pode ser substituída facilmente por algum *Bucket* disponível, podendo ser Google, AWS ou outros.

O projeto da aplicação web utiliza o sistema de versionamento GIT e tem seu código hospedado no Github no link <https://github.com/pedro-bigz/web-ppgco>. O projeto referente ao Bucket utilizado também utiliza o GIT e o GitHub, para acessar o repositório acesse <https://github.com/pedro-bigz/ppgco-bucket>

4.4 Organização de Pastas e Arquitetura (*API*)

A *API* segue a estrutura construída pelo *Framework NestJS*, onde é dividida em módulos. Cada módulo possui um contexto definido, como Usuários, Estudantes, Professores ou Marcos Temporais. Dentro de cada módulo existem arquivos, como *models*, *controllers*, *middlewares*, *providers*, *services* e os arquivos de testes unitários. A biblioteca utiliza estratégias de inversão de dependências e permite a criação de aplicações escaláveis, testáveis e com baixo acoplamento.

A Figura 1 apresenta os módulos da *API*. São eles:

- O modulo **Common** contém classes reutilizáveis, uteis para centralizar código e evitar repetições.
- O modulo **Utils** contém funções auxiliares.
- O modulo **Config** contém arquivos de configuração de *API's* e do banco de dados.
- O modulo **Errors** implementa as regras de tratamento de alguns erros não previstos pelo *framework NestJS*.
- O Module **Crud-Generator** é um modulo gerador de código, implementado para facilitar o desenvolvimento de novos módulos.
- Os demais módulos implementam a manipulação de dados do banco de dados e fazem a comunicação com as aplicações clientes.

4.5 Organização de Pastas e Arquitetura (WEB)

A biblioteca *React* utilizada no *frontend* não possui uma estrutura pré-definida, permitindo ao desenvolvedor, organizar a sua maneira. Contudo existem recomendações

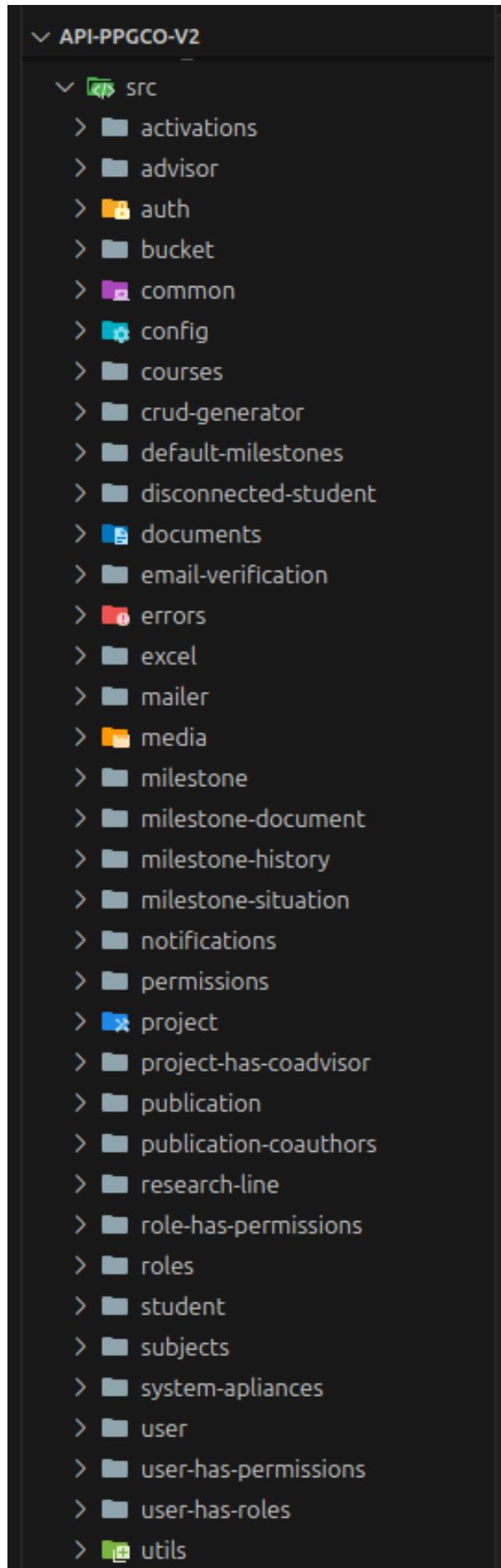


Figura 3 – Organização de Pastas API

de estruturas de pasta para projetos *ReactJS*. A estrutura definida está retratada na Figura 2.

A estrutura fica dividida entre as pastas:

- **Assets:** contém imagens, ícones e as fontes do projeto.
- **Components:** contém todos os componentes reutilizáveis com propósito genérico, como campos de formulário, tabelas, modais entre outros.
- **Business:** contém componentes reutilizáveis, com regras específicas do *PPGCO*.
- **Core:** contém arquivos de configuração de *API*, contextos, *hooks* e tipos comuns da aplicação.
- **Layout:** contém os componentes que definem a disposição dos elementos principais da aplicação, como a *Sidebar*, a *Topbar* e a estrutura onde as páginas serão exibidas.
- **Services:** são funções que implementam a comunicação com algum determinado serviço ou *API*.
- **Utils:** são as funções auxiliares da aplicação.
- **Views:** implementam cada uma das telas da aplicação.

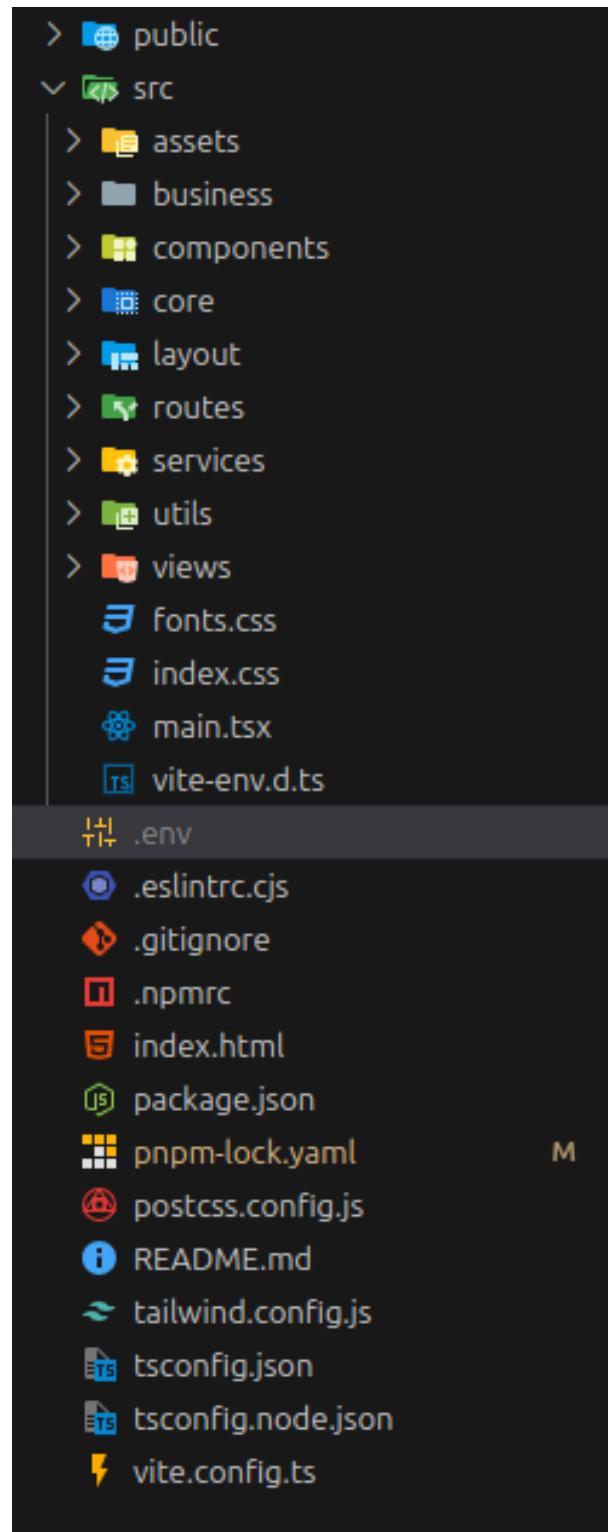


Figura 4 – Organização de Pastas WEB

4.6 Diagramas

4.6.1 Fluxogramas

4.6.1.1 Fluxo de cadastro de estudantes

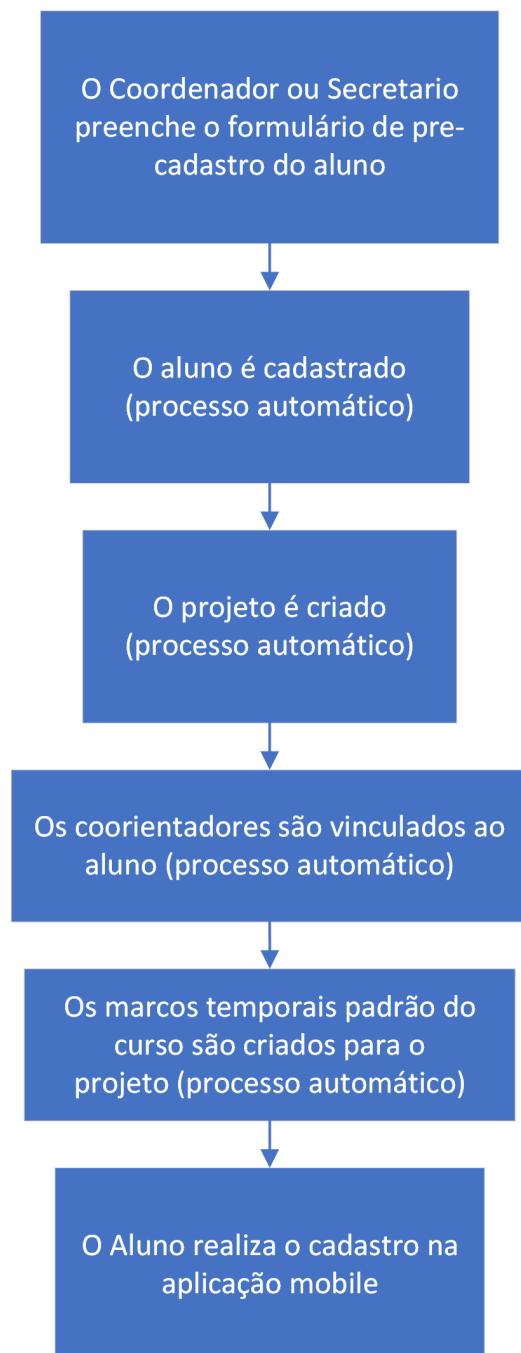


Figura 5 – Fluxo de cadastro de Estudantes

No cadastro de estudantes, o formulário solicita ao coordenador ou secretário os seguintes dados:

- Nome
- Sobrenome
- Data de nascimento
- Celular
- Lattes
- Curso
- Nome temporário do projeto
- Orientador
- Lista de Coorientadores

Ao enviar o formulário, o sistema cria os registros no banco de dados do aluno, do projeto, dos coorientadores e dos marcos temporais padrão de acordo com o curso informado. Concluída essa etapa, o aluno pode finalizar seu cadastro no aplicativo móvel, onde definirá o nome do projeto e a senha de acesso à plataforma.

4.6.1.2 Fluxo de cadastro marcos temporais

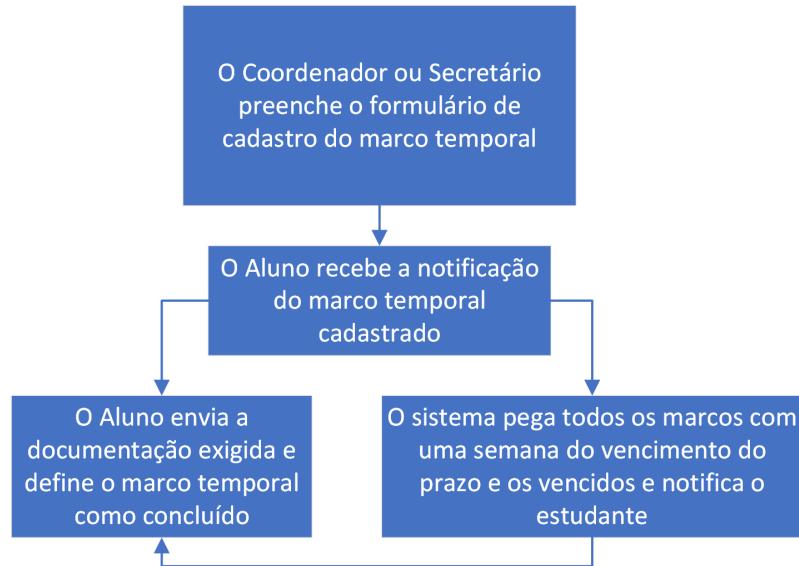


Figura 6 – Fluxo de cadastro de Marcos Temporais

Ao cadastrar um marco temporal, os estudantes são notificados via e-mail e via plataforma. Quando os marcos estiverem próximos do vencimento do prazo (1 semana) ou estiverem vencendo, o estudante receberá novas notificações a respeito.

4.6.1.3 Fluxo de Edição marcos temporais

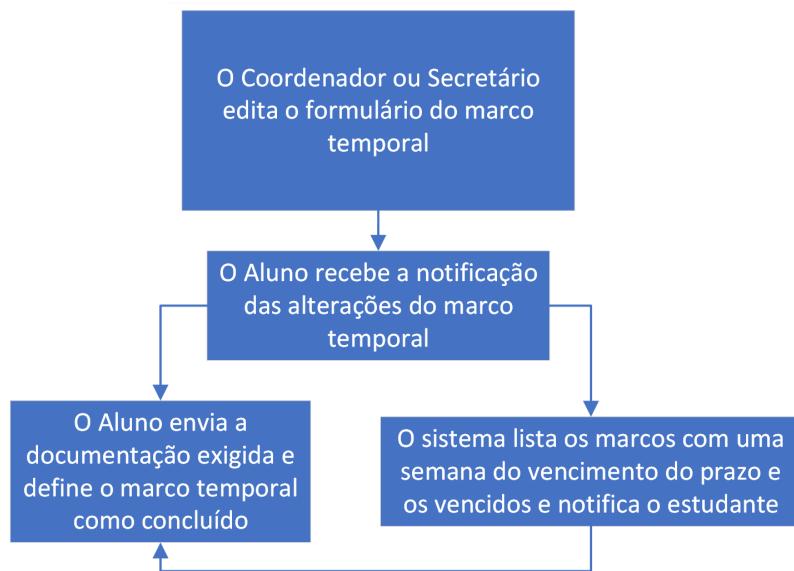


Figura 7 – Fluxo de edição de Marcos Temporais

Ao editar um marco temporal, os estudantes são notificados via e-mail e via plataforma, tal qual ocorre no fluxo de cadastro. As notificações a respeito da proximidade do vencimento dos prazos também ocorrem como no fluxo de cadastro.

4.6.2 Telas do Sistema

O sistema SAPE-PPGCO adota uma abordagem modular, utilizando um conjunto de componentes reutilizáveis compartilhados pela maioria das interfaces. Para telas de acesso restrito, a estrutura básica é definida por um componente Layout principal, composto por três elementos fundamentais:

1. **TopBar** - Barra superior do sistema
2. **Sidebar** - Menu lateral de navegação
3. **Outlet** - Área dinâmica para renderização do conteúdo específico de cada página

4.6.2.1 Categorias de Telas

O sistema possui duas categorias principais de interfaces:

4.6.2.1.1 Telas de Listagem

Apresentam dados em formato tabular e incluem os seguintes componentes padrão:

- Cabeçalho
- Corpo
- Título
- Tabela
- Filtros
- Paginação
- Botão de link para a tela de cadastro

4.6.2.1.2 Telas de Formulário

Responsáveis pela coleta de dados e são compostas por:

- O provedor do formulário
- Campos para inserção de dados
- Campos para seleção de dados
- Campos para envio de arquivos
- Botão de envio do formulário

4.6.2.1.3 Regras de Validação e Padrões Visuais

O sistema implementa as seguintes convenções para a garantia de qualidade dos dados:

- **Indicação de campos obrigatórios:** Asterisco vermelho (*) adjacente ao rótulo do campo
- **Validação integrada:** Verificação automática dos dados ao submeter o formulário

4.6.2.2 Gerenciamento de Estudantes

4.6.2.2.1 Listagem

A listagem de discentes é composta por informações provenientes de relacionamentos entre estudantes, projetos, linhas de pesquisa, orientadores e dados cadastrais. Os campos exibidos e suas respectivas origens são:

Dados Pessoais (Tabela de Usuários):

- Nome
- Sobrenome
- E-mail
- Data de Nascimento

Dados Acadêmicos (Tabela de Estudantes):

- Nº de Matrícula
- Lattes
- Data de Ingresso
- Data Sucupira

Dados de Vinculação (Relacionamentos):

- Título (tabela de projetos)
- Linha de Pesquisa (tabela de projetos + tabela de linha de pesquisa)
- Orientador (tabela de projetos + tabela de professores)
- Curso (tabela de projetos + tabela de cursos)

4.6.2.2.2 Cadastro/Edição

Descrição do Formulário de Cadastro

O formulário de cadastro é dividido em seções, conforme detalhado abaixo:

Dados do Estudante

Contém as informações pessoais e acadêmicas do aluno:

- **Dados Pessoais:** Nome, sobrenome, e-mail e data de nascimento.
- **Dados de Matrícula:** Número de matrícula, data de ingresso e outras informações acadêmicas relevantes.

Estudantes
(8 resultados)

ID	Nº Matrícula	Nome Completo	E-mail	Data de Ingresso	Data Sucupira	Data Início da Bolsa	Cadastrado em	Ações
6	1111BBB444	Pedro Joaquim	pj@gmail.com	03 mar 2024	03 mar 2024		2024-06-03T05:34:35.000Z	<button>Ações</button>
14	1111BBB555	Pedro Henrique Nobrega	ph.nobrega@gmail.com	08 ago 2023	08 ago 2023		2024-06-03T06:17:52.000Z	<button>Ações</button>
	Estudantes	Gina Goodman	gina.goodman@gmail.com	10 out 2023	10 nov 2023	10 out 2023	2024-06-03T06:20:50.000Z	<button>Ações</button>
17	1111BBB777	Raymond Holt	raymond.holt@gmail.com	15 nov 2023	15 nov 2023	15 nov 2023	2024-06-05T06:32:52.000Z	<button>Ações</button>
18	1111BBB888	Maria Chiquinha	maria.chiquinha@gmail.com	05 mar 2016	05 mar 2023		2024-06-05T06:57:55.000Z	<button>Ações</button>
19	1111BBB999	Gina Weasley	gina.weasley@gmail.com	16 jan 2021	16 dez 2021		2024-06-05T07:00:39.000Z	<button>Ações</button>
20	112BSI200	Deodoro Matinhos	deodoro.matinhos@gmail.com	21 mar 2021	21 mai 2022		2024-06-05T07:05:56.000Z	<button>Ações</button>
21	1111BBB666	João Araújo	joao.araujo@gmail.com	15 dez 2023	15 dez 2023		2024-08-26T16:46:11.000Z	<button>Ações</button>

Exibindo 8 items de 8

Figura 8 – Listagem de Estudantes

Observações:

- A senha não é definida neste momento, pois o discente deverá finalizar o cadastro via aplicativo móvel, onde definirá sua senha de acesso.
- Uma senha temporária é gerada automaticamente e enviada para o e-mail cadastrado.

Dados do Projeto

Contém informações sobre o projeto vinculado ao estudante:

- **Título do Projeto** (valor padrão: "ND", para casos em que o título ainda não foi definido).
- **Linha de Pesquisa**
- **Orientador**
- **Período do Projeto:** Data de início e data de término.

Figura 9 – Formulário de Estudantes

4.6.2.3 Gerenciamento de Professores

4.6.2.3.1 Listagem

A tela apresenta os dados cadastrais dos professores, combinando informações do perfil de usuário com dados acadêmicos específicos. Embora o acesso ao sistema pelos professores ainda não esteja implementado, os seguintes campos são exibidos:

Dados Pessoais (Tabela de Usuários):

- Nome
- Sobrenome
- E-mail
- Data de Nascimento

Dados Acadêmicos (Tabela de Professores):

- Link Lattes (tabela `professores`)
- Linha de Pesquisa (tabela `linha de pesquisa`)

Os dados apresentados na imagem abaixo são fictícios, apenas utilizam o nome de pessoas reais.

ID	Nome Completo	E-mail	Data de Nascimento	Linha de Pesquisa	Lattes	Ações
177	Autran Macedo	autran@ufu.br	31 dez 1980	Engenharia de Software	http://lattes.cnpq.br/4481587640098813	Ações
170	Bruno Augusto Nassif Travencolo	travencolo@ufu.br	25 mai 1984	Processamento Digital de Imagens	http://lattes.cnpq.br/2590427857264952	Ações
176	Caio Augusto Rodrigues dos Santos	caio@santos@ufu.br	31 dez 1980	Programação de Computadores	http://lattes.cnpq.br/6061460578417908	Ações
178	Carlos Cesar Mansur Tuma	carlos.tuma@ufu.br	31 dez 1980	Inteligência Artificial	http://lattes.cnpq.br/0916152883066962	Ações
179	Christiane Regina Soares Brasil	christiane@ufu.br	31 dez 1980	Programação Bioinspirada	http://lattes.cnpq.br/5064007473299439	Ações
180	Daniel Antônio Furtado	furtado@ufu.br	31 dez 1980	Visão Computacional	http://lattes.cnpq.br/0277782739196529	Ações
181	Daniel Duarte Abdala	abdala@ufu.br	31 dez 1980	Processamento Digital de Imagens	http://lattes.cnpq.br/0083781586520627	Ações
213	João Marquêz	joao.marquez@gmail.com	19 dez 1992	Banco de Dados	asdf	Ações
169	Rivalino Matias Junior	rivalino@ufu.br	30 dez 1978	Engenharia de Confiabilidade de Software	http://lattes.cnpq.br/3034950214458518	Ações

Exibindo 9 items de 9

Figura 10 – Listagem de Professores

4.6.2.3.2 Cadastro/Edição

O formulário de cadastro de professores pede os dados pessoais e os dados acadêmicos do professor.

Dados Pessoais

- Informações individuais e de contato do docente

Dados Acadêmicos

- Currículo Lattes e linha de pesquisa

The screenshot shows a web-based application interface for managing professors at UFU. On the left, there's a sidebar with various navigation options: Dashboard, Disciplinas, Marcos Temporais, Publicações, Professores (which is highlighted in blue), Estudantes, Linhas de Pesquisa, and Configurações. The main content area is titled 'Formulário de cadastro de Professores'. It contains several input fields: 'Nome' (Name), 'Sobrenome' (Surname), 'E-mail' (Email), 'Lattes' (Lattes number), 'Linha de Pesquisa' (Research Line), 'Data de Nascimento' (Birth Date) with a calendar icon, 'Telefone' (Phone), and a 'Cadastrar' (Register) button.

Figura 11 – Formulário de Professores

4.6.2.4 Gerenciamento de Publicações

4.6.2.4.1 Listagem

A tela de listagem exibe as seguintes informações sobre cada publicação cadastrada:

Dados da Publicação

- Título
- Nome do veículo de Publicação
- O nome do estudante que enviou a publicação
- O nome do projeto ao qual a publicação está vinculada

Relacionamentos

- O estudante que enviou a publicação
- O projeto ao qual a publicação está vinculada
- Coautores da publicação

Publicações
(12 resultados)

ID	Título	Veículo de Publicação	Estudante	Projeto	Data de inicio	Data de Fim	Cadastrado em	Ações
5	Publication 1	ufu	Pedro Henrique Nobrega	Projeto 001	12 dez 2020	12 dez 2021	2024-09-20T07:15:32.000Z	<button>Ações</button>
6	Publication 1	ufu	Gina Goodman	Project 0002	12 dez 2020	12 dez 2021	2024-09-20T07:15:32.000Z	<button>Ações</button>
7	Publication 1	ufu	Maria Chiquinha	Project 0004	12 dez 2020	12 dez 2021	2024-09-20T07:15:32.000Z	<button>Ações</button>
8	Publication 1	ufu	Gina Weasley	Project 0005	12 dez 2020	12 dez 2021	2024-09-20T07:15:32.000Z	<button>Ações</button>
9	Publication 1	ufu	Pedro Henrique Nobrega	Projeto 001	12 dez 2020	12 dez 2021	2024-09-20T07:16:00.000Z	<button>Ações</button>
10	Publication 1	ufu	Gina Goodman	Project 0002	12 dez 2020	12 dez 2021	2024-09-20T07:16:00.000Z	<button>Ações</button>
11	Publication 1	ufu	Maria Chiquinha	Project 0004	12 dez 2020	12 dez 2021	2024-09-20T07:16:00.000Z	<button>Ações</button>
12	Publication 1	ufu	Gina Weasley	Project 0005	12 dez 2020	12 dez 2021	2024-09-20T07:16:00.000Z	<button>Ações</button>
29	Publication 1	ufu	Pedro Henrique Nobrega	Projeto 001	12 dez 2020	12 dez 2021	2024-09-20T07:29:52.000Z	<button>Ações</button>
30	Publication 1	ufu	Gina Goodman	Project 0002	12 dez 2020	12 dez 2021	2024-09-20T07:29:52.000Z	<button>Ações</button>

Exibindo 10 items de 12

Figura 12 – Listagem de Publicações

4.6.2.4.2 Cadastro/Edição

Esta tela permite o registro de novas publicações acadêmicas no sistema SAPE-PPGCO, coletando as informações necessárias para vincular a produção científica aos projetos e autores correspondentes.

Informações Básicas da Publicação

- Título
- Veículo de publicação
- Data de publicação

Coautores e Vinculação

- Coautores (nome, sobrenome, Lattes e afiliação)
- Projetos vinculados (campo de seleção com projetos ativos)

The screenshot shows the 'UFU' system interface with a sidebar containing links like Dashboard, Disciplinas, Marcos Temporais, Publicações (selected), Professores, Estudantes, Linhas de Pesquisa, and Configurações. The main content area is titled 'Formulário de cadastro de Publicações'. It contains fields for 'Projeto', 'Título', 'Véculo de Publicação', date range ('Data de Início' and 'Data de Fim'), 'Coautores' (Nome and SobreNome), 'Lattes', and 'Afiliação'. A blue '+ Cadastrar' button is located at the bottom right.

Figura 13 – Formulário de Publicações

4.6.2.5 Gerenciamento de Linhas de Pesquisa

4.6.2.5.1 Listagem

O módulo de linhas de pesquisa tem uma estrutura mais simples, por não possuir relacionamentos com outras entidades. Sua função é padronizar a gestão das linhas de pesquisa em todo o sistema.

The screenshot shows the 'UFU' system interface with a sidebar containing links like Dashboard, Disciplinas, Marcos Temporais, Publicações, Professores, Estudantes, Linhas de Pesquisa (selected), and Configurações. The main content area is titled 'Linhas de Pesquisa' and shows '(14 resultados)'. It features a table with columns: ID, Título, Descrição, Data de criação, Data de atualização, and Ações. The table lists 14 items from 1 to 10. At the bottom, it says 'Exibindo 10 items de 14' and has pagination controls (10, 1, 2, >).

ID	Título	Descrição	Data de criação	Data de atualização	Ações
1	Banco de Dados	Descrição - Banco de Dados	2024-05-23T06:13:15.000Z	2024-05-26T08:23:15.000Z	Ações
2	Programação de Computadores	Descrição - Programação de Computadores	2024-05-23T06:39:54.000Z	2024-05-26T08:23:01.000Z	Ações
3	Inteligência Artificial	Descrição - Inteligência Artificial	2024-05-26T01:57:21.000Z	2024-05-26T08:22:49.000Z	Ações
4	Redes de Computadores	Descrição - Redes de Computadores	2024-05-26T08:23:39.000Z	2024-05-26T08:23:39.000Z	Ações
5	Engenharia de Software	Descrição - Engenharia de Software	2024-05-26T08:23:57.000Z	2024-05-26T08:23:57.000Z	Ações
6	Redes Neurais	Descrição - Redes Neurais	2024-05-26T08:24:27.000Z	2024-05-26T08:24:27.000Z	Ações
7	Computação Natural	Descrição - Computação Natural	2024-05-26T08:24:51.000Z	2024-05-26T08:24:51.000Z	Ações
8	Bioinformática	Descrição - Bioinformática	2024-05-26T08:25:29.000Z	2024-05-26T08:25:29.000Z	Ações
9	Computação Evolutiva	Descrição - Computação Evolutiva	2024-05-26T08:25:46.000Z	2024-05-26T08:25:46.000Z	Ações
10	Processamento Digital de Imagens	Descrição - Processamento Digital de Imagens	2024-05-26T08:26:15.000Z	2024-05-26T22:20:59.000Z	Ações

Figura 14 – Listagem de Linhas de Pesquisa

4.6.2.5.2 Cadastro/Edição

Para cadastrar uma nova linha de pesquisa, o usuário deve inserir o título/nome da linha no formulário designado. Adicionalmente, há a opção de incluir uma descrição para complementar as informações.

The screenshot shows a web application interface for managing research lines at UFU. The left sidebar has a dark blue header with the UFU logo and several menu items: Dashboard, Disciplinas, Marcos Temporais, Publicações, Professores, Estudantes, Linhas de Pesquisa (which is highlighted in blue), and Configurações. The main content area has a light gray background. At the top, there's a navigation bar with 'Inicio' and a search bar labeled 'Pesquisar no histórico'. Below that is a section titled 'Formulário de cadastro de Linhas de Pesquisa' containing two input fields: 'Título' and 'Descrição', both with placeholder text. A blue 'Cadastrar' button is located to the right of the 'Descrição' field. In the top right corner, there's a user profile icon with the name 'Pedro Paulo' and the email 'pedro.paulo.ufu@gmail.com'.

Figura 15 – Formulário de Linhas de Pesquisa

4.6.2.6 Gerenciamento de Marcos Temporais

4.6.2.6.1 Listagem

A tela de listagem de marcos temporais exibe as informações de cada marco, incluindo os dados do estudante e do projeto associados, além do status atual. Os marcos são priorizados de acordo com seu estado, seguindo a seguinte ordem de exibição:

- Em Atraso (prioritários)
- Não Iniciado
- Aguardando Validação
- Documentação Pendente
- Em Andamento
- Concluído

ID	Descrição	Estudante - Projeto	Prazo	Reunião do Colegiado	Nº processo SEI	Exige Documentação	Status	Ações
67	AC Milestone 0001	Gina Weasley - Project 0005	05 jun 2024	456456	12344	0	Em Atraso	<button>Ações</button>
66	AC Milestone 0001	Maria Chiquinha - Project 0004	05 jun 2024	456456	12344	0	Em Atraso	<button>Ações</button>
65	AC Milestone 0001	Gina Goodman - Project 0002	05 jun 2024	456456	12344	0	Em Atraso	<button>Ações</button>
14		Gina Weasley - Project 0005	07 jun 2023	234234	23423423	0	Em Atraso	<button>Ações</button>
20		Deodoro Matinhos - Project 0006	07 jun 2023	234234	23423423	0	Em Atraso	<button>Ações</button>
17		Raymond Holt - Título 0001	07 jun 2023	234234	23423423	0	Em Atraso	<button>Ações</button>
6	default_mestrado_test_0002	Deodoro Matinhos - Project 0006	22 dez 2024	default_meeting_collegiate_0002		1	Não Iniciado	<button>Ações</button>
70	Teste 0003	João Araújo - ND	01 set 2024	561980023	12300085858	1	Não Iniciado	<button>Ações</button>
64	Milestone Teste 0007	Gina Goodman - Project 0002	04 jun 2023	12344	12344	1	Não Iniciado	<button>Ações</button>
5	default_mestrado_test_0001	Deodoro Matinhos - Project 0006	22 dez 2024	default_meeting_collegiate_0001		1	Não Iniciado	<button>Ações</button>

Figura 16 – Listagem de Marcos Temporais

4.6.2.6.2 Cadastro/Edição

A tela de cadastro de marcos temporais permite o registro de novas etapas ou entregas associadas a um projeto e um estudante, facilitando o acompanhamento do progresso e o cumprimento de prazos.

Campos do formulário

- Título
- Descrição
- Data de entrega
- Número do Processo SEI
- Identificador da Reunião do Colegiado
- Status inicial (valor padrão: não iniciado)
- Projeto vinculado

The screenshot shows a web-based application for managing temporal milestones. On the left, there's a sidebar with navigation links: Dashboard, Disciplinas, Marcos Temporais (which is highlighted in blue), Publicações, Professores, Estudantes, Linhas de Pesquisa, and Configurações. The main area is titled 'Formulário de cadastro de Milestone'. It contains several input fields: 'Projeto' (dropdown), 'Prazo' (text input with placeholder 'dd/mm/aaaa'), 'Descrição' (text area), 'Status' (dropdown), 'Reunião do Colegiado' (text input), 'Número do Processo SEI' (text input), and a checkbox labeled 'Necessita documentação'. At the bottom right of the form is a blue 'Cadastrar' button.

Figura 17 – Formulário de Marcos Temporais

4.6.2.7 Gerenciamento de Disciplinas

4.6.2.7.1 Listagem

A tela de disciplinas apresenta todas as disciplinas cadastradas no sistema, exibindo suas informações principais e os cursos aos quais estão vinculadas. Cada disciplina é composta pelos seguintes dados:

Dados da Disciplina

- Código
- Nome
- Curso (Mestrado, Doutorado e Pós-Doutorado)
- Carga Horária
- Créditos

4.6.2.7.2 Cadastro/Edição

O formulário de cadastro de disciplina permite o registro de novas disciplinas no sistema, associando-as aos cursos correspondentes e definindo suas características acadêmicas.

ID	Código	Nome	Carga Horária	Créditos	Curso	Ações
1	facom001	Disciplina Teste 001	60	100	Mestrado	<button>Ações</button>
2	facom002	Disciplina Teste 002	60	100	Mestrado	<button>Ações</button>
3	facom003	Disciplina Teste 003	60	100	Mestrado	<button>Ações</button>
4	facom004	Disciplina Teste 004	60	100	Mestrado	<button>Ações</button>
5	facom005	Disciplina Teste 005	60	100	Mestrado	<button>Ações</button>
6	facom006	Disciplina Teste 006	60	100	Mestrado	<button>Ações</button>
7	facom007	Disciplina Teste 007	60	100	Mestrado	<button>Ações</button>
8	facom008	Disciplina Teste 008	60	100	Mestrado	<button>Ações</button>
9	facom009	Disciplina Teste 009	60	100	Mestrado	<button>Ações</button>
10	facom010	Disciplina Teste 010	60	100	Mestrado	<button>Ações</button>

Figura 18 – Listagem de Disciplinas

Campos do formulário

- Código (exemplo: GSI001)
- Nome
- Curso (Mestrado, Doutorado e Pós-Doutorado)
- Carga Horária
- Créditos

The screenshot shows a web application interface for the Universidade Federal de Uberlândia (UFU). The left sidebar contains navigation links: Dashboard, Disciplinas (highlighted in blue), Marcos Temporais, Publicações, Professores, Estudantes, Linhas de Pesquisa, and Configurações. The main content area has a header with 'Início' and a search bar. A user profile is shown on the right. The central part of the screen displays a form titled 'Formulário de cadastro de Disciplinas' (Discipline Registration Form) with fields for Código, Nome, Curso (dropdown), Carga Horária, and Créditos, and a 'Cadastrar' button.

Figura 19 – Formulário de Disciplinas

5 Considerações Finais

O desenvolvimento deste sistema integrado para acompanhamento do percurso dos estudantes de pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia (UFU) surge como uma solução eficiente para os desafios enfrentados no processo atual, que é manual, descentralizado e propenso a falhas. Ao unificar as funcionalidades de gestão de prazos, vinculação de publicações científicas e acompanhamento por parte de professores, coordenação e secretaria, o sistema proposto otimiza a comunicação e a organização, reduzindo a carga burocrática e os riscos de inadimplência acadêmica.

Além disso, a automatização de notificações e a centralização das informações permitem que os estudantes tenham maior autonomia e clareza sobre suas obrigações, enquanto os orientadores e a administração ganham ferramentas eficazes para monitorar o progresso dos discentes. A eliminação da dependência de sistemas desconexos também contribui para a eficiência institucional, garantindo maior confiabilidade e transparência no processo.

Como trabalhos futuros, sugere-se a integração do sistema com plataformas acadêmicas externas (como o Lattes e repositórios de publicações) e a implementação de funcionalidades adicionais, como relatórios personalizados e análises preditivas para auxiliar na tomada de decisões. Por fim, acredita-se que esta solução não apenas beneficia a comunidade acadêmica da UFU, mas também serve como modelo para outras instituições que enfrentam desafios semelhantes na gestão de pós-graduação.

Referências

AIRBNB. **Javascript: JavaScript style guide.** N.D. Citado na página [16](#).

ALVES, W. P. M. Desenvolvimento do painel de inscrição para o programa de pós-graduação em ciência da computação. Universidade Federal de Uberlândia, 2023. Citado 2 vezes nas páginas [18](#) e [19](#).

BRAULIO-GONZALO, M. et al. Gestión de proyectos con projectlibre tm. Publicacions de la Universitat Jaume I, 2021. Citado na página [20](#).

CAPES, C. de Aperfeiçoamento de Pessoal de N. S. Portaria, **Portaria CAPES Nº 158, de 17 de agosto de 2023.** 2023. Institui e regulamenta a governança da informação relacionada à pós-graduação stricto sensu. Disponível em: <<https://cad.capes.gov.br/ato-administrativo-detalhar?idAtoAdmElastic=12742>>. Citado na página [10](#).

CONTRIBUTORS, M. **JavaScript.** 2022. <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Accessed: 2023-11-24. Citado na página [16](#).

_____. **Visão geral das tecnologias JavaScript.** 2023. <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/JavaScript_technologies_overview>. Accessed: 2023-11-24. Citado na página [16](#).

CORADIN, F. **O Futuro dos Bancos de Dados: Tendências e Inovações.** 2023. <<https://horusinfo.com.br/o-futuro-dos-bancos-de-dados-tendencias-e-inovacoes/>>. Accessed: 2023-11-20. Citado na página [16](#).

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados.** 7. ed. [S.l.]: Pearson, 2019. Citado na página [21](#).

FERREIRA, P. **JavaScript História: Uma breve história sobre a origem e evolução do JavaScript.** 2023. <<https://awari.com.br/javascript-historia-uma-breve-historia-sobre-a-origem-e-evolucao-do-javascript/>>. Accessed: 2023-11-24. Citado na página [16](#).

GONÇALVES, M. M. 2021. <<https://medium.com/@marcelomg21/arquitetura-de-software-estilos-e-padrões-de-design-50d62d684ef2>>. Accessed: 2023-12-2. Citado na página [14](#).

HAT, R. **API REST.** 2023. <<https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>>. Accessed: 2023-12-2. Citado na página [15](#).

LTDA, U. T. I. . U. do B. T. N.D. <<https://www.uber.com/en-BR/blog/uber-tech-stack-part-two/>>. Accessed: 2023-11-24. Citado na página [16](#).

MENDES, C. A. D.; PEREIRA, W. C.; SOUZA, R. R. de. Sistema para gerenciamento de projetos de software. **Centro Universitário do Sul de Minas Gerais**, 2010. Citado na página [19](#).

- MOHR, M. 2022. <<https://medium.com/@mariannamohr/arquitetura-msc-o-que-%C3%A9}-e-por-que-usar-42ad4cf19583>>. Accessed: 2023-12-2. Citado na página 15.
- NET, C. **Cliente-Servidor, uma estrutura lógica para a computação centralizada.** [S.l.]: Controle Net, 2023. <<https://www.controle.net/faq/cliente-servidor-uma-estrutura-para-a-computacao-centralizada>>. Accessed: 2023-12-2. Citado na página 14.
- NOLETO, C. N.D. <<https://blog.betrybe.com/javascript/>>. Accessed: 2023-11-24. Citado na página 16.
- ORACLE. **MySQL :: MySQL Customers by Country : United States.** N.D. <<https://www.mysql.com/customers/country/?id=1259>>. Accessed: 2023-11-20. Citado na página 16.
- PACIEVITCH, Y. **MySQL.** N.D. <<https://www.infoescola.com/informatica/mysql>>. Accessed: 2023-11-20. Citado na página 16.
- PAULA, D. G. d. Projeto e implementação de um aplicativo ios para apoio à jornada acadêmica dos estudantes no ppgco. Universidade Federal de Uberlândia, 2024. Citado 2 vezes nas páginas 11 e 18.
- PEREIRA, E. C. d. O. Sistema de alerta de limites e prazos acadêmicos para discentes de graduação da facom. Universidade Federal de Uberlândia, 2023. Citado na página 16.
- ROCK CONTENT. **Conheça a história da Internet, sua finalidade e qual o cenário atual.** [S.l.]: Rock Content, 2020. <<https://rockcontent.com/br/blog/historia-da-internet>>. Accessed: 2023-11-24. Citado na página 13.
- SCANDIFFIO, M. G. Sistema de controle de prazos para alunos de pós-graduação. Universidade Federal de Uberlândia, 2017. Citado 3 vezes nas páginas 15, 16 e 18.
- SCHMITZ, M. **Monitoramento de aplicativos do WebSocket.** 2021. <<https://www.dotcom-monitor.com/blog/pt-br/monitoramento-de-aplicativos-do-websocket>>. Accessed: 2023-11-24. Citado na página 13.
- SERVICES, I. A. W. N.D. <<https://aws.amazon.com/pt/what-is/iot>>. Accessed: 2023-11-24. Citado na página 13.
- _____. **O que é arquitetura orientada a serviços?** N.D. <<https://aws.amazon.com/pt/what-is/service-oriented-architecture>>. Accessed: 2023-11-24. Citado na página 14.
- SOUZA, I. de. **Entenda o que é HTTP e o quão importante esse protocolo é para o seu site.** [S.l.]: Rock Content, 2019. <<https://rockcontent.com/br/blog/http>>. Accessed: 2023-11-24. Citado na página 13.
- SUPERO. **SOA vs. microsserviços: como determinar qual é o ideal?** N.D. <<https://www.supero.com.br/blog/soa-vs-microsservicos-como-determinar-qual-e-a-arquitetura-ideal>>. Accessed: 2023-11-24. Citado na página 15.