

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Luiz André da Silva Carvalho

**Arquitetura Frontend e Interface de um
Sistema de Acompanhamento de Egressos para
a UFU**

Uberlândia, Brasil

2025

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Luiz André da Silva Carvalho

**Arquitetura Frontend e Interface de um Sistema de
Acompanhamento de Egressos para a UFU**

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: Maria Adriana Vidigal de Lima

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2025

Luiz André da Silva Carvalho

Arquitetura Frontend e Interface de um Sistema de Acompanhamento de Egressos para a UFU

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Trabalho aprovado. Uberlândia, Brasil, 11 de abril de 2025:

Maria Adriana Vidigal de Lima
Orientador

Maurício Cunha Escarpinati

Rafael Dias Araújo

Uberlândia, Brasil
2025

Dedico este trabalho à minha família, pelo apoio incondicional em todos os momentos, e aos meus amigos, que sempre estiveram ao meu lado nesta jornada. Um agradecimento especial à minha companheira, Thainá Pereira Silva, por sua paciência, incentivo e apoio essencial ao longo dessa caminhada.

Agradecimentos

Agradeço à minha orientadora, Prof^a. Dra. Maria Adriana Vidigal de Lima, por sua orientação, paciência e contribuições fundamentais para a realização deste trabalho.

Aos técnicos do Centro de Tecnologia da Informação e Comunicação, pelo suporte prestado e pela disponibilidade em auxiliar nos aspectos técnicos necessários para o desenvolvimento desta pesquisa.

Resumo

O presente trabalho dá continuidade ao desenvolvimento de um sistema de acompanhamento de egressos para a Universidade Federal de Uberlândia (UFU). Nesta etapa, o foco está na criação de uma interface responsiva, acessível e modular, utilizando o framework Angular em conformidade com as diretrizes do *Design System* do Governo Brasileiro (Ds-Gov). O sistema foi concebido para integrar *frontend*, *backend* e banco de dados, sendo que o *frontend* foi concluído, o *backend* encontra-se em andamento, e a modelagem do banco de dados passou por ajustes. Com base em padrões ágeis e utilizando o framework Scrum, o *frontend* incorpora funcionalidades como gerenciamento de informações acadêmicas e profissionais, coleta de depoimentos e exibição de *dashboards* públicos. Algumas funcionalidades, como a integração com o SSO (*Single Sign-On* ou Autenticação Única) da UFU e a extração automatizada de dados da plataforma Lattes foram postergadas para etapas futuras do projeto. Os resultados obtidos nesta etapa demonstram que a aplicação facilita o acompanhamento de ex-alunos, cumpre exigências regulatórias e fortalece o vínculo com egressos, apresentando potencial de expansão para outras instituições de ensino superior.

Palavras-chave: Acompanhamento de Egressos, Sistema de Egressos, Ex-alunos, Angular, Design System, Sistema Acadêmico.

Lista de ilustrações

Figura 1 – Linha do tempo do projeto	29
Figura 2 – Tela de Login	31
Figura 3 – Página Home	31
Figura 4 – Tela de Informações Acadêmicas	32
Figura 5 – Tela de Informações Profissionais	33
Figura 6 – Tela de Depoimentos	33
Figura 7 – Modal de Consentimento LGPD	34
Figura 8 – Tela de Publicações	35
Figura 9 – Tela de Informações Pessoais	35
Figura 10 – Tela de Listagem de Egressos	36
Figura 11 – Tela de Detalhes do Egresso	37
Figura 12 – Etapa de <i>discovery</i> : documentação inicial no Notion.	47
Figura 13 – Detalhamento das funções e layout no Notion.	48
Figura 14 – Protótipo de média fidelidade no Figma.	49
Figura 15 – Protótipo de alta fidelidade no Figma.	49
Figura 16 – Tela final da funcionalidade de depoimentos após o <i>review</i>	50

Lista de tabelas

Lista de abreviaturas e siglas

CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CSS	Cascading Style Sheets
CTIC	Centro de Tecnologia da Informação e Comunicação
DsGov	Design System do Governo Brasileiro
DOM	Document Object Model
eMAG	Modelo de Acessibilidade em Governo Eletrônico
ePING	Padrões de Interoperabilidade de Governo Eletrônico
HTML	HyperText Markup Language
IES	Instituição de Ensino Superior
JSON	JavaScript Object Notation
LGPD	Lei Geral de Proteção de Dados
MEC	Ministério da Educação
MVC	Model-View-Controller
MVP	Minimum Viable Product ou Produto Mínimo Viável
PWA	Progressive Web Application
SPA	Single Page Application
SSO	Single Sign-On
UFU	Universidade Federal de Uberlândia
XML	eXtensible Markup Language

Sumário

1	INTRODUÇÃO	11
1.1	Objetivos	12
1.2	Ambiente de Desenvolvimento e Metodologia Ágil	12
1.3	Organização da Monografia	13
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Modelo Cliente e Servidor	15
2.1.1	Cliente	15
2.1.2	Servidor	16
2.2	Tecnologias e ferramentas	16
2.2.1	Frontend	16
2.2.2	Backend	17
2.2.3	Banco de dados	17
2.3	Metodologia Ágil	18
2.4	Adoção de Padrões Governamentais	19
2.4.1	Metodologia e Artefatos	20
2.4.2	Vantagens da Adoção dos Padrões	20
3	TRABALHOS RELACIONADOS	22
3.1	Sistema de acompanhamento de egressos UFU: levantamento e análise de requisito	22
3.2	As universidades e o relacionamento com seus ex-alunos: uma análise de portais online de egressos	23
3.3	Ações de acompanhamento de egressos: um estudo das universidades públicas do sul do Brasil	23
4	DESENVOLVIMENTO	25
4.1	Arquitetura do Sistema	25
4.1.1	Arquitetura de Frontend	25
4.1.1.1	Estrutura de Pastas	25
4.1.1.2	Gerenciamento de Rotas	26
4.1.1.3	Perfis de Acesso	26
4.1.1.4	Principais Convenções	27
4.1.1.5	Recursos Adicionais	27
4.1.2	Arquitetura de Backend	27
4.1.3	Arquitetura de Banco de Dados	28

4.2	Etapas do desenvolvimento	28
4.2.1	Linha do tempo	28
4.2.2	Etapas de Desenvolvimento do Frontend	29
4.2.2.1	Criação do Projeto e Estruturação Inicial	29
4.2.2.2	Desenvolvimento de Componentes	29
4.2.2.3	Desenvolvimento de Páginas	30
4.3	Adequações ao Padrão e-MAG	38
4.4	Acesso ao Código-Fonte	39
5	CONCLUSÃO	40
	REFERÊNCIAS	42
	APÊNDICES	45
	APÊNDICE A – BIBLIOTECA DE COMPONENTES BASEADOS NO DESIGN SYSTEM DO GOVERNO BRASI- LEIRO (GOVBR DS)	46
	APÊNDICE B – PROCESSO ITERATIVO DO DESENVOLVIMENTO DA FUNCIONALIDADE DE DEPOIMENTOS	47
	ANEXOS	51
	ANEXO A – DOCUMENTO DE FORMALIZAÇÃO DA DEMANDA	52

1 Introdução

A importância do acompanhamento de egressos pelas Instituições de Ensino Superior (IES) tem sido cada vez mais reconhecida em todo o mundo, à medida que as demandas do mercado de trabalho e da sociedade evoluem rapidamente. No contexto brasileiro, embora o investimento na criação e no desenvolvimento do ensino superior tenha sido mais tardio em comparação com outros países do Ocidente, as IES estão progressivamente reconhecendo a necessidade de implementar políticas eficazes de acompanhamento de ex-alunos. Essas políticas não apenas refletem a interação dinâmica entre a universidade e seu entorno social, mas também desempenham um papel crucial na avaliação constante da qualidade do ensino e na adaptação dos currículos às demandas em evolução do mercado. Esse processo proporciona à universidade uma avaliação mais próxima da realidade e das exigências do mercado de trabalho, possibilitando a implementação de melhorias contínuas em seus cursos (GUIMARÃES, 2012). Nesse contexto, a implementação de um sistema computacional integrado para facilitar a comunicação com os ex-alunos é uma prática amplamente adotada por diversas instituições de ensino superior no Brasil e no mundo. Essa abordagem se revela como uma ferramenta essencial para que as instituições acompanhem de perto a jornada de seus graduados, compreendam suas demandas e desafios no mercado de trabalho e na sociedade em geral.

A relevância deste trabalho está profundamente enraizada na necessidade da Universidade Federal de Uberlândia (UFU) de estabelecer e manter laços duradouros com seus graduados mesmo após deixarem a universidade. Acompanhar o desenvolvimento profissional e acadêmico dos ex-alunos não apenas contribui para elevar o padrão de qualidade do ensino fornecido, mas também fortalece os vínculos entre a instituição e seus antigos estudantes, o que, por sua vez, enriquece a comunidade acadêmica da UFU, permitindo que a instituição se mantenha atualizada sobre as necessidades do mercado de trabalho e as demandas da sociedade.

O acompanhamento de egressos desempenha um papel importante em avaliações realizadas pelo Ministério da Educação (MEC) e Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) em relação ao desempenho acadêmico da universidade. Seguindo as diretrizes estabelecidas na Portaria Nº 92, datada de 31 de Janeiro de 2014, a UFU e outras instituições de ensino superior têm a obrigação de demonstrar a eficácia de seus programas de acompanhamento de ex-alunos. Portanto, a concepção e implementação de uma ferramenta dedicada a atender a essas necessidades não é apenas desejável, mas também essencial para garantir a conformidade com os padrões regulatórios e a excelência acadêmica.

A importância do acompanhamento de ex-alunos vai além do ambiente acadêmico, refletindo diretamente na questão das desigualdades sociais. Como destacado por Vargas (2011), a assistência estudantil, incluindo o acompanhamento de egressos, é um investimento a médio e longo prazo em mão de obra qualificada e em justiça social. Ao proporcionar um ambiente onde o estudante poderá receber oportunidades de emprego e se manter conectado ao ecossistema acadêmico, as universidades não apenas promovem o acesso à educação de qualidade, mas também contribuem significativamente para a redução das disparidades sociais. Portanto, o desenvolvimento de estratégias eficazes de acompanhamento de ex-alunos não apenas fortalece os laços entre a instituição e seus graduados, mas também desempenha um papel fundamental na construção de uma sociedade mais justa e igualitária.

Após analisar estudos acadêmicos como os de Cabral, Silva e Pacheco (2016) e Guimarães (2012), fica evidente que as IES reconhecem unanimemente a importância de manter um vínculo contínuo com seus ex-alunos. Essa prática é reconhecida como crucial, considerando que os graduados representam um ativo valioso, e a qualidade da formação reflete diretamente na reputação das instituições de ensino responsáveis por ela. O estudo conduzido por Mendes (2023), será o ponto de partida essencial para a elaboração do sistema de software proposto neste trabalho. A análise detalhada dos requisitos apresentada juntamente com os casos de uso forneceu uma base sólida para o desenvolvimento deste projeto, garantindo a coerência e a eficácia do sistema de acompanhamento de ex-alunos a ser desenvolvido.

1.1 Objetivos

Os objetivos principais deste trabalho englobam a validação do levantamento de requisitos, a elaboração dos artefatos necessários para o projeto, como o design de interface e a modelagem de banco de dados, e a implementação do *frontend* do sistema de acompanhamento de ex-alunos, com base nas diretrizes do *Design System* do Governo Brasileiro (DsGov). Já o *backend* e outras funcionalidades avançadas foram planejados, mas não implementados nesta etapa do projeto. Como objetivos secundários, destaca-se a análise dos processos de negócio relacionados ao acompanhamento de ex-alunos e a compreensão de seu papel na evolução contínua da universidade, visando garantir que o sistema atenda às necessidades institucionais e dos egressos.

1.2 Ambiente de Desenvolvimento e Metodologia Ágil

O desenvolvimento do sistema foi guiado por uma combinação estratégica de ferramentas tecnológicas e a metodologia ágil Scrum. Essa abordagem promoveu um processo

iterativo, incremental e colaborativo, garantindo flexibilidade, adaptação às mudanças e entregas contínuas de valor ao longo do processo.

O Framework Angular ([GOOGLE, 2024a](#)) foi empregado para o desenvolvimento do *frontend*, oferecendo recursos avançados e uma estrutura robusta para criar interfaces dinâmicas e responsivas. A Linguagem Java ([ORACLE, 2024a](#)) foi escolhida para a posterior implementação do *backend* por ser conhecida pela confiabilidade e flexibilidade no desenvolvimento de sistemas escaláveis e seguros. O sistema de gerenciamento de banco de dados selecionado foi o PostgreSQL ([GROUP, 2024](#)) de forma a prover um ambiente confiável e eficiente para armazenar e recuperar dados de forma otimizada. Para o estudo e o projeto da interface gráfica do sistema, a ferramenta Figma ([FIGMA, 2024](#)) foi designada para facilitar a criação de interfaces de usuário de forma colaborativa, interativa e eficiente.

Quanto aos padrões e regulamentos, o desenvolvimento do sistema foi orientado pelas diretrizes estabelecidas pelos Padrões de Interoperabilidade de Governo Eletrônico - ePING ([BRASILEIRO, 2018](#)), o Padrão Digital de Governo - Design System ([BRASILEIRO, 2022b](#)) e o eMAG - Modelo de Acessibilidade em Governo Eletrônico ([BRASILEIRO, 2014](#)). Todas essas escolhas foram discutidas e validadas em conjunto com os técnicos do Centro de Tecnologia da Informação e Comunicação ([CTIC, 2014](#)) da UFU, garantindo assim a conformidade com os padrões e diretrizes institucionais.

1.3 Organização da Monografia

Esta monografia está organizada em cinco capítulos principais, conforme descrito a seguir:

- **Capítulo 1 - Introdução:** Apresenta o contexto e a relevância do trabalho, os objetivos gerais e específicos, além das ferramentas utilizadas no desenvolvimento do sistema.
- **Capítulo 2 - Fundamentação Teórica:** Discute os conceitos e tecnologias relevantes para o desenvolvimento do sistema, incluindo o modelo cliente-servidor, as ferramentas escolhidas e os padrões aplicados.
- **Capítulo 3 - Trabalhos Relacionados:** Aborda estudos e sistemas similares, destacando os avanços e limitações encontrados em pesquisas anteriores e sua relação com o presente trabalho.
- **Capítulo 4 - Desenvolvimento:** Detalha as etapas de desenvolvimento do sistema, com foco na arquitetura do *frontend*, no planejamento do *backend* e na modelagem do banco de dados, além de descrever as funcionalidades implementadas.

- **Capítulo 5 - Conclusão:** Apresenta uma síntese dos resultados alcançados, desafios enfrentados, contribuições do trabalho e sugestões para melhorias e estudos futuros.

Adicionalmente, são incluídos apêndices e anexos que complementam as informações apresentadas no corpo do texto, fornecendo detalhes técnicos e referências adicionais.

2 Fundamentação Teórica

Este capítulo proporciona a base teórica necessária para compreender as tecnologias empregadas no desenvolvimento do sistema. Ele abrange os conceitos essenciais relacionados às ferramentas e linguagens utilizadas, delineando seu papel na arquitetura e funcionamento do sistema proposto de acompanhamento dos egressos da UFU.

2.1 Modelo Cliente e Servidor

O modelo cliente-servidor é uma arquitetura de software amplamente utilizada na construção de sistemas distribuídos, sendo baseada na divisão de responsabilidades entre o cliente, que solicita serviços, e o servidor, que os fornece. Segundo [Tanenbaum e Steen \(2007\)](#), “o modelo cliente-servidor é um modelo de computação que distribui o trabalho entre dois tipos de entidades: um fornecedor de serviços e um consumidor de serviços”. Como muitos sistemas cliente-servidor têm o objetivo de simplificar o acesso dos usuários aos bancos de dados, é comum dividir essa arquitetura em três camadas distintas:

1. Interface do Usuário: Esta camada é responsável por tudo o que é necessário para interagir diretamente com o usuário, como gerenciamento de exibição. Aqui, os usuários interagem com o sistema, enviando solicitações e recebendo respostas.
2. Processamento: No nível de processamento, estão localizadas as aplicações responsáveis por processar as solicitações do cliente e executar a lógica de negócios. Aqui, as operações solicitadas pelos usuários são processadas e os resultados são preparados para serem enviados de volta ao cliente.
3. Dados: Esta camada gerencia os dados reais que estão sendo manipulados pelo sistema. Nela, os dados são armazenados, recuperados, atualizados e removidos conforme necessário para atender às solicitações dos usuários.

2.1.1 Cliente

O cliente, ou *frontend*, é a parte do sistema com a qual os usuários interagem diretamente. Ele é responsável por apresentar informações ao usuário e capturar entradas do usuário. De acordo com [Pressman \(2014\)](#), “a interface do usuário é a parte do sistema com a qual o usuário interage. Isso inclui todas as telas, páginas, formulários e outros artefatos visíveis”.

2.1.2 Servidor

O servidor, ou *backend*, é responsável por processar as solicitações do cliente, executar a lógica de negócios e interagir com o banco de dados. Segundo [Tanenbaum e Steen \(2007\)](#), “o servidor é uma entidade que oferece serviços a outras entidades. Os servidores normalmente têm mais poder de processamento, armazenamento e memória do que os clientes”.

2.2 Tecnologias e ferramentas

A implementação do sistema de acompanhamento de egressos da UFU envolve um conjunto de ferramentas e metodologias que promovem escalabilidade, acessibilidade e segurança. O desenvolvimento segue uma arquitetura modular projetada para atender às necessidades do sistema de forma eficiente.

2.2.1 Frontend

Angular ([GOOGLE, 2024a](#)) é um framework de desenvolvimento de aplicativos web criado e mantido pela Google, baseado em JavaScript ([MOZILLA, 2024b](#)) e construído com TypeScript ([MICROSOFT, 2024](#)). Utiliza um modelo de programação baseado em componentes para facilitar a criação de aplicativos web modulares e escaláveis. Oferece recursos como vinculação de dados bidirecional (*Two-Way Data Binding*), injeção de dependência e suporte a testes automatizados, sendo uma plataforma completa para construir aplicações web complexas. O Angular não utiliza virtualização do Modelo de Objetos do Documento (DOM), mas sim mecanismos próprios de detecção de alterações na interface, como o *Two-Way Data Binding* ([GOOGLE, 2024f](#)).

Esse framework adota o conceito de programação declarativa para interfaces e componentes, enquanto a programação imperativa é utilizada para regras de negócio. Angular segue o padrão *Model, View e Controller* (MVC) no lado cliente, trabalhando com *Hypertext Markup Language* (HTML), *Cascading Style Sheet* (CSS) e JavaScript. De acordo com [Guedes \(2017\)](#), Angular é um framework para *Single Page Application* (SPA) no desenvolvimento *frontend*, destacando-se pela modularização. A arquitetura modular do Angular aliada ao conceito de SPA proporciona uma experiência de usuário mais fluida, evitando a necessidade de recarregar a página durante a interação. Esse modelo permite o carregamento inicial otimizado dos arquivos essenciais de JavaScript e CSS, melhorando o desempenho e a responsividade da aplicação. A implementação do carregamento lento dos módulos ([GOOGLE, 2024b](#)) permite carregar partes do aplicativo apenas quando necessário, contribuindo significativamente para a escalabilidade do projeto ao otimizar o desempenho e a eficiência no uso de recursos. Adicionalmente, o Angular é composto por

diversas bibliotecas, sendo a *core* a mais crucial delas, contendo o código base essencial para o funcionamento do framework.

2.2.2 Backend

Java ([IBM, 2024a](#)) é uma linguagem de programação amplamente empregada no desenvolvimento de aplicativos empresariais devido à sua portabilidade, confiabilidade e robustez. Com uma sintaxe limpa e suporte à orientação a objetos, Java proporciona recursos avançados para a construção de aplicações escaláveis e de alto desempenho. Sua ampla adoção na indústria de software e a abundância de bibliotecas disponíveis tornam Java uma escolha popular para projetos de grande envergadura.

Complementando o ecossistema Java, o Spring ([IBM, 2024b](#)) é um conjunto de ferramentas voltado para o desenvolvimento de aplicativos empresariais, oferecendo um modelo abrangente para simplificar o processo de codificação. Recursos como injeção de dependência, controle transacional e segurança tornam o Spring uma solução ideal para criar aplicativos modulares e robustos. O Spring Boot, uma extensão do Spring, facilita ainda mais o desenvolvimento ao adotar uma abordagem baseada em convenções para a configuração de aplicativos Java, tornando a combinação Spring e Spring Boot uma escolha poderosa para criar sistemas escaláveis e confiáveis.

Além disso, o Swagger ([IBM, 2022](#)) é utilizado para a documentação e o consumo de serviços baseados em arquitetura RESTful ([IBM, 2024c](#)), que estruturam a comunicação entre cliente e servidor com base na separação de responsabilidades, o uso de métodos HTTP padronizados e a troca de dados em JSON ou XML. Com o *OpenAPI Specification* ([SOFTWARE, 2021](#)), o Swagger oferece ferramentas para projetar, documentar e consumir APIs de forma padronizada. Seu principal objetivo é simplificar a comunicação e integração com serviços web RESTful, fornecendo uma documentação precisa e uniforme que facilita a compreensão e a interação eficaz entre diferentes sistemas.

2.2.3 Banco de dados

O PostgreSQL ([GROUP, 2024](#)), ou simplesmente Postgres, é um sistema de gerenciamento de banco de dados relacional de código aberto amplamente utilizado em diversas aplicações, desde projetos web até sistemas empresariais. Conhecido por sua confiabilidade e flexibilidade, ele oferece ferramentas poderosas para armazenamento e recuperação de dados de forma eficiente. Com suporte a consultas complexas, transações seguras e extensões personalizáveis, o PostgreSQL é uma escolha popular entre desenvolvedores e empresas que buscam um banco de dados robusto e adaptável às suas necessidades.

Durante o processo de modelagem do banco de dados, o uso do DBeaver ([DBEAVER, 2024](#)) foi indispensável. Essa ferramenta, também de código aberto, proporcionou

uma interface amigável que facilitou tanto o design quanto a visualização do esquema do banco. Além disso, ela permitiu realizar testes, ajustar detalhes e validar o modelo com mais segurança e agilidade. O DBeaver não só simplificou o trabalho técnico, como também ajudou a manter a organização do projeto, garantindo que as decisões de modelagem fossem bem fundamentadas e alinhadas às necessidades do sistema.

2.3 Metodologia Ágil

Diante da complexidade e das constantes mudanças inerentes ao desenvolvimento de sistemas, optou-se por adotar a metodologia ágil Scrum ([SCRUM.ORG](https://www.scrum.org), 2024). Essa abordagem foi escolhida por sua capacidade de promover flexibilidade, colaboração e entregas frequentes, elementos essenciais para um projeto alinhado às expectativas das partes interessadas.

O Scrum organiza o trabalho em ciclos curtos, chamados de *sprints*, que geralmente duram de duas a quatro semanas. Cada *sprint* segue etapas bem definidas, que garantem o progresso contínuo e estruturado do projeto:

- **Planejamento da Sprint:** A equipe define os objetivos e tarefas prioritárias a serem realizadas durante o ciclo, com base no *backlog* do projeto.
- **Execução:** As tarefas planejadas são desenvolvidas, incentivando a colaboração e a resolução de problemas de forma iterativa.
- **Reuniões Diárias (*Daily Scrum*):** Curtos encontros diários são realizados para compartilhar o progresso, identificar desafios e ajustar os planos, quando necessário.
- **Revisão da Sprint:** Ao final do ciclo, os resultados são apresentados às partes interessadas, permitindo o recebimento de feedback valioso.
- **Retrospectiva:** A equipe reflete sobre o que funcionou bem e o que pode ser melhorado, aplicando essas lições para aprimorar as próximas *sprints*.

A aplicação do Scrum trouxe diversas vantagens para o projeto, como:

- **Flexibilidade:** A possibilidade de adaptar-se rapidamente às mudanças de requisitos ou prioridades ao longo do desenvolvimento.
- **Colaboração:** Uma comunicação contínua e próxima entre a equipe e os *stakeholders*, garantindo alinhamento e clareza.
- **Entrega Iterativa:** A entrega de funcionalidades em pequenos incrementos agregou valor ao sistema de forma constante.

- **Foco na Qualidade:** A revisão frequente dos resultados permitiu a identificação precoce de problemas e contribuiu para a melhoria contínua.

Com o Scrum, o projeto ganhou não apenas organização, mas também agilidade, permitindo ajustes frequentes e entregas significativas a cada etapa. Essa abordagem foi fundamental para atender às expectativas dos *Stakeholders* e para construir uma base sólida que possibilitará futuras expansões e melhorias.

2.4 Adoção de Padrões Governamentais

Ao longo das últimas décadas, o desenvolvimento do governo eletrônico no Brasil foi marcado por distintos focos e estratégias. De 2000 a 2024, foram definidos marcos importantes na modernização da administração pública, com a implementação de políticas e estruturas voltadas para a digitalização e a melhoria dos serviços públicos. O processo começou com a introdução de iniciativas para garantir a conectividade e a disponibilização de informações ao público, como o portal gov.br, que centraliza serviços digitais do governo federal. Ao longo do tempo, o país avançou para a adoção de sistemas mais integrados, baseados em princípios e padrões de interoperabilidade, transparência e eficiência (SANTOS; FIGUEIREDO; GOMES, 2023).

No desenvolvimento da aplicação de acompanhamento de egressos, a adoção de padrões governamentais desempenhou um papel essencial na garantia de qualidade, acessibilidade e conformidade com diretrizes nacionais. Esses padrões foram aplicados para criar uma solução tecnológica robusta, segura e inclusiva, que atende tanto às necessidades institucionais quanto às exigências legais. Três principais padrões foram adotados no projeto: o Modelo de Acessibilidade em Governo Eletrônico (eMAG), o Design System do Governo Brasileiro (DsGov) e os Padrões de Interoperabilidade de Governo Eletrônico (ePING). Cada um deles trouxe vantagens específicas que contribuíram significativamente para o sucesso do sistema.

O **eMAG**¹ tem como objetivo principal assegurar a acessibilidade de aplicações eletrônicas, permitindo que pessoas com diferentes habilidades, incluindo aquelas com deficiência, utilizem os serviços disponibilizados. Para atender às diretrizes do eMAG, foram implementadas práticas como o uso de contraste adequado nas interfaces, a navegação por teclado e a inclusão de descrições textuais em imagens. Essas medidas não apenas garantem a inclusão social e digital, mas também cumprem requisitos legais importantes, ampliando o alcance do sistema a um público diversificado.

O **DsGov**² foi utilizado como referência para padronizar a interface e melhorar a experiência do usuário. Este design system oferece um conjunto de componentes pron-

¹ <https://emag.governoeletronico.gov.br/>

² <https://www.gov.br/ds/home>

tos, como botões, formulários e tabelas, que foram integrados ao projeto para garantir consistência visual e funcionalidade intuitiva. Por exemplo, o *date-time-picker*, alinhado ao DsGov, facilitou a seleção de datas, enquanto componentes como *alerts* e *paginators* proporcionaram uma navegação mais eficiente. A padronização visual também trouxe benefícios como a redução de esforços no desenvolvimento e a confiança dos usuários ao interagir com uma interface consistente e familiar.

Por fim, o **ePING**³ foi essencial para orientar a integração entre o frontend e o backend do sistema, garantindo interoperabilidade por meio do uso de formatos padronizados, como o JSON. Este padrão promoveu uma comunicação eficiente entre os componentes do sistema, facilitando a troca de informações e reduzindo custos de manutenção. Além disso, o alinhamento com o ePING assegurou que o sistema pudesse se integrar facilmente a outras aplicações governamentais, garantindo escalabilidade e flexibilidade para expansões futuras.

A adoção combinada desses padrões resultou em uma aplicação alinhada às melhores práticas governamentais, promovendo acessibilidade, usabilidade e interoperabilidade. Além de atender às necessidades específicas da Universidade Federal de Uberlândia, o sistema desenvolvido também representa uma solução modelo que pode ser adaptada por outras instituições de ensino superior. Assim, o trabalho contribui para a consolidação de um ecossistema tecnológico mais eficiente, inclusivo e sustentável.

2.4.1 Metodologia e Artefatos

A aplicação dos padrões foi organizada em etapas metodológicas claras:

- **Análise e Validação de Requisitos:** Realizou-se um levantamento detalhado dos requisitos, seguido por sua validação, para garantir que fossem viáveis e alinhados às necessidades institucionais.
- **Criação de Protótipos:** Foram desenvolvidos protótipos de alta fidelidade baseados no DsGov. Por exemplo, a página inicial do sistema foi projetada com uma estrutura modular, respeitando as diretrizes visuais e funcionais do padrão.
- **Modelagem de Arquitetura:** Seguindo orientações do **CTIC (2014)** e respeitando definições do ePING (**BRASILEIRO, 2018**), definiu-se uma arquitetura flexível e escalável, alinhada às boas práticas de desenvolvimento.

2.4.2 Vantagens da Adoção dos Padrões

A aplicação dos padrões governamentais trouxe benefícios significativos:

³ <https://eping.governoeletronico.gov.br/>

- **Conformidade Legal:** Aderência a normas de acessibilidade e interoperabilidade exigidas por leis e regulamentos brasileiros.
- **Usabilidade e Inclusão:** Interfaces mais acessíveis e intuitivas, promovendo uma experiência satisfatória para todos os usuários.
- **Eficiência e Manutenção:** A padronização facilitou o desenvolvimento, integração e futura manutenção do sistema.
- **Credibilidade Institucional:** O alinhamento com padrões reconhecidos reforça a confiança dos *stakeholders* e usuários finais na aplicação.

Com essa abordagem, o sistema desenvolvido não apenas atende às expectativas da Universidade Federal de Uberlândia (UFU), mas também estabelece uma base sólida para futuras expansões e adaptações por outras instituições.

3 Trabalhos Relacionados

Este capítulo apresenta uma análise dos trabalhos relacionados que fundamentam o desenvolvimento do sistema de acompanhamento de egressos da UFU. A primeira seção descreve o levantamento e a análise de requisitos realizados por [Mendes \(2023\)](#), que representam a base para o projeto de um sistema de acompanhamento de egressos na UFU. O estudo de [Cabral, Silva e Pacheco \(2016\)](#) analisa portais de egressos das principais universidades brasileiras, revelando deficiências como a falta de integração com redes sociais e funcionalidades voltadas para a promoção de oportunidades profissionais. Finalmente, a pesquisa de [Pacheco \(2017\)](#) oferece uma visão sobre as ações de acompanhamento em universidades públicas do sul do Brasil.

3.1 Sistema de acompanhamento de egressos UFU: levantamento e análise de requisito

O trabalho de [Mendes \(2023\)](#) constitui um estudo dos requisitos para a construção de um sistema de acompanhamento de egressos para a Universidade Federal de Uberlândia (UFU). A partir de reuniões com representantes da instituição, análise de documentos e estudo de sistemas similares, foi realizada uma modelagem de processos, usando a notação BPMN (*Business Process Model and Notation*), para detalhar fluxos operacionais como cadastro de egressos, respostas à questionários e visualização de um painel de indicadores (*dashboard*).

Os modelos criados permitiram uma visualização clara das interações entre os processos e permitiram identificar atores, cenários de uso e fluxo de atividades para o software, resultando na elaboração detalhada do documento de requisitos funcionais e não funcionais, além da definição da estrutura do banco de dados. Foram definidos 15 requisitos funcionais, abrangendo funcionalidades como autenticação, coleta e atualização de dados acadêmicos, envio de comunicados, criação de questionários e geração de relatórios. Os requisitos não funcionais incluíram aspectos como segurança, interoperabilidade com sistemas existentes da UFU e conformidade com a LGPD (Lei Geral de Proteção de Dados).

Os resultados se relacionam diretamente com o presente trabalho, que se concentra na continuidade do processo, visando a construção dos artefatos restantes da fase de projeto, como protótipos de interfaces e modelagem do banco de dados, além do desenvolvimento de uma versão inicial do sistema. Desta forma, ao construir sobre as bases estabelecidas pelo trabalho anterior, este projeto busca concretizar as especificações de-

finalizadas e transformá-las em uma versão funcional da aplicação para que a UFU possa, a partir dela, implantar um sistema de acompanhamento de seus egressos.

3.2 As universidades e o relacionamento com seus ex-alunos: uma análise de portais online de egressos

O estudo conduzido por [Cabral, Silva e Pacheco \(2016\)](#) verificou a existência e analisou os portais de egressos das dez melhores Instituições de Ensino Superior públicas brasileiras, conforme o Ranking Universitário Folha de 2015 ([FOLHA, 2015](#)). Os resultados revelaram falta de atenção adequada por parte das instituições aos portais de egressos, a ausência de centralização desses portais e a carência de uma página dedicada às oportunidades profissionais para os ex-alunos.

A análise também revelou a falta de recursos atrativos, como integração com redes sociais e aplicativos úteis. Este estudo é relevante para o presente trabalho, pois fornece visões valiosas sobre as lacunas e deficiências encontradas nos portais de egressos das instituições de ensino superior. Enquanto o artigo focou na análise desses portais, o projeto em questão visa desenvolver um sistema de acompanhamento de egressos que possa preencher tais lacunas. O sistema será direcionado para atender às necessidades e expectativas dos ex-alunos, promovendo um relacionamento eficaz entre as instituições e seus egressos.

3.3 Ações de acompanhamento de egressos: um estudo das universidades públicas do sul do Brasil

O trabalho em [Pacheco \(2017\)](#) investigou as ações de acompanhamento de egressos em universidades públicas do sul do Brasil. Utilizando um estudo multicaso descritivo, o método adotado examinou vinte instituições, evidenciando que a maioria ainda precisa evoluir em suas estratégias de acompanhamento. A análise revelou que, embora as universidades usem predominantemente métodos indutivos para coletar informações sobre os ex-alunos, poucas realizam pesquisas constantes, e os portais do egresso ainda não são plenamente explorados.

Esses resultados têm relevância direta para o desenvolvimento do sistema de acompanhamento do egresso proposto, pois enumeram diversas funcionalidades encontradas nos portais do egresso das universidades estudadas que podem ser exploradas e aplicadas no presente trabalho. A conclusão assinala que a implementação de serviços e benefícios para aproximar os egressos da instituição e a padronização das informações coletadas são diretrizes importantes para o sucesso do sistema, alinhando-se com a proposta de manter o vínculo entre a instituição de ensino superior e seus ex-alunos, bem como aprimorar a

qualidade da formação oferecida.

4 Desenvolvimento

No capítulo dedicado ao desenvolvimento deste trabalho, são abordados os aspectos técnicos e as decisões arquiteturais que sustentam a construção do sistema proposto. A partir de uma visão geral da arquitetura do sistema, serão discutidos os componentes principais, incluindo o frontend, o backend e o banco de dados, detalhando a estruturação, as interações entre os módulos e as tecnologias utilizadas. Este capítulo visa oferecer uma compreensão aprofundada das etapas de desenvolvimento, desde o planejamento inicial até a implementação das funcionalidades específicas, proporcionando uma base sólida para a análise do processo de construção do sistema.

4.1 Arquitetura do Sistema

Esta seção detalha a arquitetura do sistema desenvolvido para o acompanhamento de egressos, englobando as camadas de *frontend*, *backend* e banco de dados. Embora este projeto tenha tido ênfase na camada de *frontend*, desenvolvendo uma interface intuitiva e acessível para os usuários, a arquitetura completa do sistema foi planejada para integrar os demais componentes de forma escalável e eficiente.

4.1.1 Arquitetura de Frontend

Este projeto foi desenvolvido utilizando o **Angular 18**, adotando uma abordagem modular para otimizar tanto a escalabilidade quanto a manutenção do código. Em projetos Angular, a organização modular implica na criação de pastas específicas para diferentes áreas do sistema, como **core**, **features**, **shared**, entre outras. Essa estrutura permite que cada parte do código tenha responsabilidades bem definidas, resultando em uma organização mais lógica, coesa e de fácil gerenciamento.

4.1.1.1 Estrutura de Pastas

A divisão da estrutura de pastas busca garantir organização, escalabilidade e facilidade de manutenção do sistema. Essa separação por responsabilidade torna o desenvolvimento mais modular e facilita o entendimento do projeto.

A estrutura principal é composta por três pastas:

- **/core** – Contém funcionalidades essenciais e globais, como:
 - **auth** – Serviços de autenticação e autorização.

- **interceptors** – Interceptadores HTTP para tratamento de requisições e respostas.
- **strategy** – Estratégias personalizadas, como métodos de autenticação.
- **update** – Gerenciamento de atualizações do sistema.
- **/features** – Agrupa funcionalidades específicas por perfil de usuário:
 - **coordinator** – Funcionalidades para coordenadores, como gestão de egressos e anúncios.
 - **graduates** – Funcionalidades para egressos, como informações acadêmicas, profissionais e depoimentos.
 - **dashboard** – Telas compartilhadas entre diferentes perfis.
 - **home, login, not-found, profile** – Telas comuns a todos os usuários.
- **/shared** – Contém recursos reutilizáveis em todo o sistema:
 - **animations, components, directives, enums, mocks, models, pipes, services, types, utils.**

Essa organização promove uma separação clara entre a infraestrutura, as funcionalidades específicas e os recursos compartilhados, otimizando a manutenção e a escalabilidade da aplicação.

4.1.1.2 Gerenciamento de Rotas

O gerenciamento de rotas é feito no arquivo **app.routes.ts**, que agrupa e direciona as rotas para as diferentes funcionalidades do projeto. Arquivos de rotas específicos para coordenadores e graduados são organizados dentro de suas respectivas pastas.

4.1.1.3 Perfis de Acesso

O sistema possui diferentes níveis de acesso, cada um com funcionalidades e telas específicas:

- **Coordenador/Diretor** - Tem acesso a funcionalidades administrativas, como o gerenciamento de anúncios e o controle de informações dos graduados.
- **Graduado/Egresso** - Pode acessar suas próprias informações acadêmicas e profissionais, além de poder enviar depoimentos.
- **Público Geral** - Tem acesso restrito, podendo apenas visualizar *dashboards* públicos.

4.1.1.4 Principais Convenções

- **Agrupamento por Perfil** - As funcionalidades são agrupadas por perfil, com algumas funcionalidades disponíveis apenas para coordenadores/diretores e outras para alunos (egressos).
- **Standalone Components** - Componentes reutilizáveis (como componentes, pipes e services) são centralizados na pasta `/shared` e utilizam do recurso de *standalone components* para promover a reutilização e evitar duplicação de código.

4.1.1.5 Recursos Adicionais

Foram implementados também os seguintes recursos para melhorar a experiência e a operação do sistema:

- **PWA e Service Worker** - A aplicação foi configurada como um Progressive Web App (PWA) ([MOZILLA, 2024c](#)), com um *service worker* que facilita o reconhecimento e atualização de novas versões através da API SwUpdate ([GOOGLE, 2024e](#)) do Angular.
- **Hospedagem na Azure e CI/CD** - Inicialmente, o projeto foi hospedado em um plano gratuito da Azure e integrado ao GitHub Actions para um fluxo de iteração eficiente.

4.1.2 Arquitetura de Backend

A arquitetura de backend do sistema será implementada utilizando a linguagem de programação **Java**, que oferece uma base robusta e escalável para o suporte das funcionalidades da aplicação de acompanhamento de egressos. Embora a implementação ainda não tenha sido iniciada, a escolha por **Java** reflete a necessidade de um ambiente confiável e capaz de lidar com grandes volumes de dados e requisições de forma eficiente.

A comunicação entre o *backend* e o *frontend* será realizada por meio de contratos de integração baseados em **JSON** ([DEVMEDIA, 2011](#)), proporcionando uma interface de dados padronizada e estruturada. Essa abordagem facilita a interoperabilidade com o *frontend* desenvolvido em Angular, garantindo que a troca de informações entre as camadas do sistema ocorra de forma eficiente e sem ambiguidade.

Embora, neste momento, não tenham sido definidas diretrizes arquiteturais específicas, como o uso de *Clean Architecture* ou outros padrões de design, essas práticas serão consideradas para a evolução do sistema. O objetivo é garantir que a arquitetura seja escalável, modular e fácil de manter, promovendo a evolução contínua do sistema sem comprometer sua performance ou qualidade. Tais diretrizes arquiteturais serão detalhadas

em etapas posteriores do desenvolvimento, com vistas a atender às necessidades futuras de manutenção e expansão do sistema.

4.1.3 Arquitetura de Banco de Dados

O banco de dados do sistema foi revisado a partir de um modelo existente desenvolvido no projeto *Sistema de Acompanhamento de Egressos UFU: Levantamento e Análise de Requisitos*, conforme descrito no trabalho de [Mendes \(2023\)](#). Após uma análise detalhada, foram identificados e levantados alguns ajustes necessários para melhor atender às necessidades atuais da aplicação. Com base nessa revisão, chegou-se a uma nova modelagem do banco de dados, mais alinhada aos requisitos do sistema de acompanhamento de egressos.

Para a criação e o ajuste do modelo, foi utilizado o software **MySQL Workbench** ([ORACLE, 2024b](#)), que fornece um ambiente integrado para criar e gerenciar esquemas de banco de dados, escrever e executar consultas SQL, realizar engenharia reversa e modelagem de dados, e ainda oferece funcionalidades para administração de usuários, backup e otimização de desempenho.

4.2 Etapas do desenvolvimento

4.2.1 Linha do tempo

A Figura 1 ilustra as etapas principais do desenvolvimento do sistema de acompanhamento de egressos. Cada fase foi dividida em sprints, iniciando pelo planejamento, que envolveu o estudo de trabalhos anteriores e a revisão da modelagem. Em seguida, sprints foram realizados para a elaboração do protótipo, com a criação de esboços, o desenvolvimento do MVP (Produto Mínimo Viável) e testes de usabilidade. A fase de desenvolvimento concentrou-se na implementação inicial do sistema, culminando na apresentação final para todos os *stakeholders*.

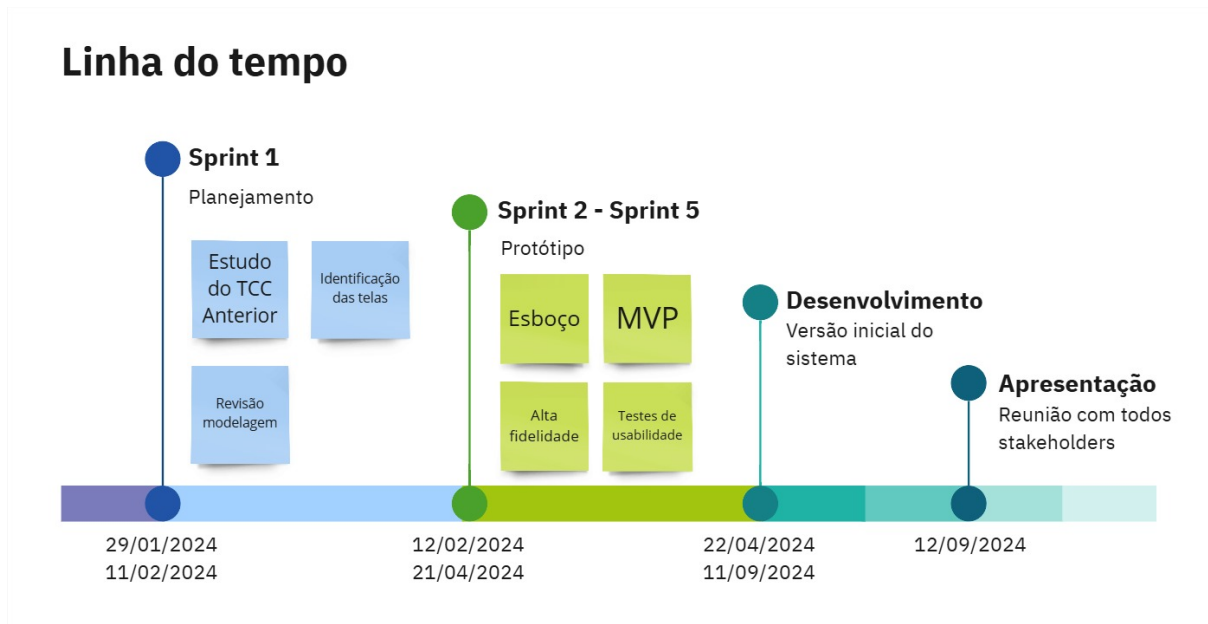


Figura 1 – Linha do tempo.

4.2.2 Etapas de Desenvolvimento do Frontend

Para o desenvolvimento do frontend do **Sistema de Egressos UFU**, foram seguidos os padrões de design estabelecidos pelo **Design System do Governo Brasileiro (DsGov)** ([BRASILEIRO, 2022b](#)), visando manter uma interface padronizada e intuitiva para os usuários. O uso do DsGov também vem ao encontro do esforço de padronização de todas as páginas e serviços oferecidos pelo CTIC/UFU, garantindo uma identidade visual unificada e uma experiência de navegação consistente para a comunidade acadêmica.

4.2.2.1 Criação do Projeto e Estruturação Inicial

A primeira etapa envolveu a criação do projeto no Angular e a estruturação inicial da arquitetura, utilizando as pastas principais **core**, **features** e **shared**. Essa organização modular foi planejada para facilitar a manutenção e a escalabilidade do sistema.

4.2.2.2 Desenvolvimento de Componentes

Os componentes essenciais para a interface foram desenvolvidos seguindo as diretrizes do DsGov, mas com adaptações para se integrarem melhor ao Angular. Embora o Design System do Governo forneça componentes prontos através da biblioteca `@govbr-ds/core`, optou-se por criar um padrão de fachada ([REFACTORING.GURU, 2024](#)), implementando APIs próprias que utilizam **signals** ([GOOGLE, 2024d](#)) e **Reactive Forms** ([GOOGLE, 2024c](#)) para um controle mais avançado.

Os principais componentes desenvolvidos incluem:

- **Estrutura:** Menu, Header, Footer.
- **Elementos de Formulário:** Botão, Input, Textarea, Select e Multi-Select, Date-Time Picker, Checkbox.
- **Componentes de Interface:** Alert, Paginator, Table, Loading, Collapse, Tabs.

Cada componente foi adaptado para atender às necessidades específicas do projeto, oferecendo uma integração mais nativa com o Angular e mantendo a flexibilidade necessária para futuras modificações.

4.2.2.3 Desenvolvimento de Páginas

As páginas do **Sistema de Egressos UFU** foram construídas com base em componentes reutilizáveis previamente desenvolvidos, como header, menu de navegação, footer, e elementos de formulário. Esse enfoque em componentes reutilizáveis garantiu uma consistência visual e simplificou a manutenção do sistema. Todas as telas foram projetadas para serem **responsivas**, garantindo uma experiência de uso fluida e adaptada a diferentes dispositivos e tamanhos de tela.

Para otimizar o desempenho, foi implementado **lazy loading** (GOOGLE, 2024b) nas rotas, permitindo que as páginas sejam carregadas apenas quando necessário, reduzindo o tempo de carregamento inicial do sistema. Além disso, durante o desenvolvimento, foram utilizados dados fictícios para popular os dados nas páginas, facilitando o teste de layout e funcionalidade antes da integração com os dados reais. O [Apêndice B](#) apresenta um exemplo do processo iterativo utilizado no desenvolvimento de funcionalidades, ilustrando como a abordagem de iteração e refinamento foi aplicada em diferentes contextos durante o desenvolvimento do sistema.

A tela de Login, mostrada na Figura 2, permite uma autenticação simulada para perfis de coordenador e egresso. Essa simulação é necessária porque, na versão final do sistema, será utilizada a autenticação oficial da UFU por meio do **Single Sign-On (SSO)**. Durante o desenvolvimento, foi criada essa versão provisória da autenticação, chamada de "mock"— ou seja, uma simulação que imita o funcionamento real sem, de fato, se conectar a sistemas externos. Com isso, é possível testar o acesso de usuários e armazenar temporariamente seus dados no *storage* (MOZILLA, 2024a) local após o login. Entre os atributos do usuário, o campo **role** indica o perfil de acesso, permitindo que a interface e as funcionalidades disponíveis sejam personalizadas para coordenadores e egressos.



Entrar com conta UFU

Usuário

Senha

Entrar [Esqueceu sua senha?](#)

[Entrar com gov.br](#)

Figura 2 – Tela de Login

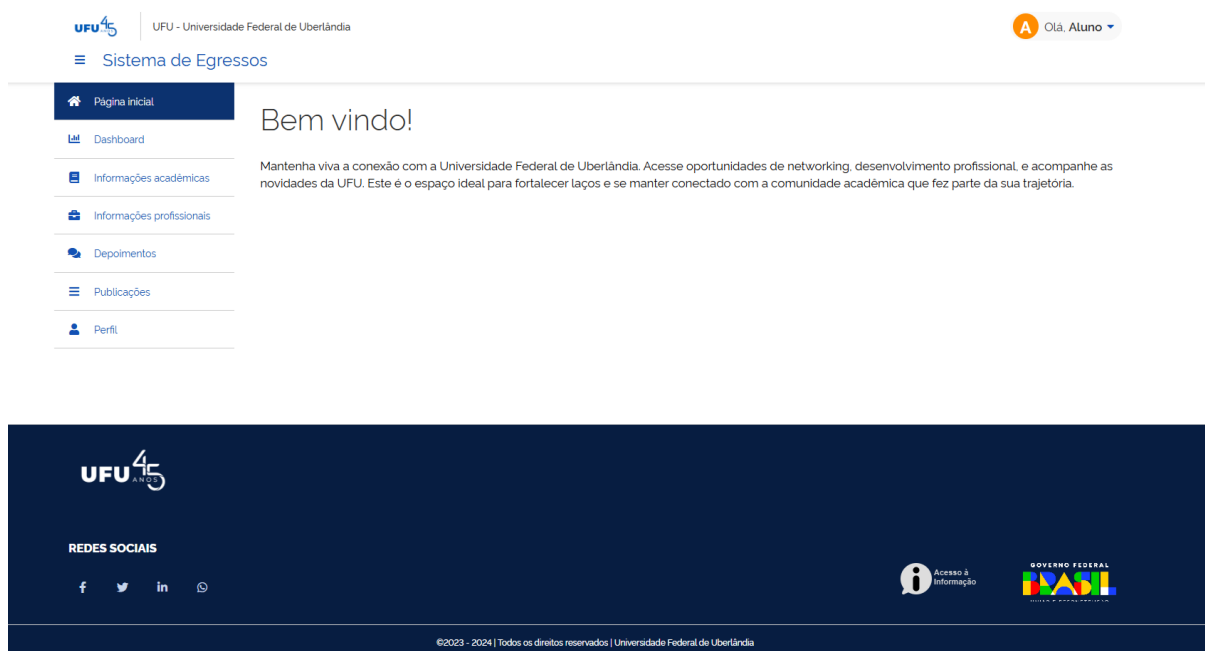


Figura 3 – Página Home

A tela de início (Home) do **Sistema de Egressos UFU**, ilustrada na Figura 3, é uma página simples e acessível a todos os perfis de usuário. Ela apresenta um título de boas-vindas e uma mensagem que se adapta conforme o perfil de acesso (coordenador, ex-aluno ou público geral). A estrutura da página segue um layout padrão com **header**, **menu** de navegação e **footer**, garantindo uma navegação fluida e intuitiva. A tela é responsiva, oferecendo uma interface limpa e eficiente para todos os usuários.

A tela de Informações Acadêmicas, mostrada na Figura 4, exibe os dados acadêmicos do usuário, permitindo o gerenciamento de múltiplas formações. Para cada curso, são apresentados detalhes como nome da instituição, tipo de instituição, nome e nível do

curso, país, cidade, data de ingresso, data de conclusão, estado e matrícula. A interface é organizada de forma a facilitar a visualização dessas informações agrupadas, com seções expansíveis que permitem ao usuário abrir ou fechar detalhes de cada formação acadêmica. A funcionalidade de adicionar um novo curso é disponibilizada através do botão **Novo Curso**, enquanto o botão **Editar Curso** permite modificar as informações de um curso específico.

A interface 'Informações Acadêmicas' apresenta uma barra de navegação superior com um ícone de casa e o texto 'Informações Acadêmicas'. Abaixo, o título 'Informações Acadêmicas' é exibido ao lado de um botão azul 'Novo Curso'. O curso selecionado, 'Engenharia de Computação - UEC', é mostrado com um ícone de seta para cima e um botão 'Editar Curso' com um ícone de lápis. O formulário contém os seguintes campos:

Nome da Instituição	Tipo de Instituição	Nome do Curso	Nível do Curso
Universidade Estadual de Campinas	Instituição Pública	Engenharia de Computação	Bacharelado
País	Data de Ingresso	Data de Conclusão	Estado
Brasil	01/02/2020	01/02/2024	São Paulo
Cidade	Matrícula		
Campinas	--		

Abaixo do formulário, há uma seção para 'Sistemas da Informação - UFU' com uma seta para baixo.

Figura 4 – Tela de Informações Acadêmicas

A Figura 5 ilustra a tela de Informações Profissionais, na qual são exibidas as experiências profissionais do usuário, permitindo a visualização e o gerenciamento de diversas posições ocupadas. Cada experiência é detalhada com informações como nome da empresa, categoria, tipo de emprego, localidade, cargo, nível de cargo, média salarial, função, data de início e data de término.

A interface possibilita ainda o relacionamento de uma experiência profissional a uma formação acadêmica específica, facilitando o acompanhamento do histórico de carreira do usuário. Há a opção de adicionar uma nova entrada por meio do botão **Nova Experiência**, ou de atualizar uma experiência existente utilizando o botão **Editar Experiência**.

Informações Profissionais

Nova Experiência

Innovatech - Analista de Sistemas ^

Editar Experiência

Nome da Empresa	Categoria	Tipo de Emprego	Localidade
Innovatech	Pública	Contrato	Híbrido
Cargo	Nível de Cargo	Data de Início	Data de Término
Analista de Sistemas	Júnior	31/01/2021	--
Média Salarial	Função	Informações Acadêmicas Relacionadas	Identificador
R\$ 4.000,00	Desenvolvimento de software	Bacharelado em Engenharia de Computação	3

Tech Solutions - Engenheiro de Software v

LUIT - Desenvolvedor FrontEnd v

Figura 5 – Tela de Informações Profissionais

A Figura 6 ilustra a tela de Depoimentos, onde o gerenciamento de relatos relacionados à experiência acadêmica ou profissional do usuário é realizado. Cada depoimento contém informações como o **texto do depoimento**, o **curso** associado, a configuração de **privacidade do depoimento** (público, privado ou anônimo) e a **data do cadastro**.

Depoimentos

Novo Depoimento

Depoimento 1 ^

Editar Depoimento

Texto do Depoimento

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce pretium sit amet sem sit amet viverra.

Curso	Privacidade do Depoimento	Identificador	Data do Cadastro do Depoimento
Bacharelado em Sistemas de Informação	Público	1	30/04/2023

Depoimento 2 v

Figura 6 – Tela de Depoimentos

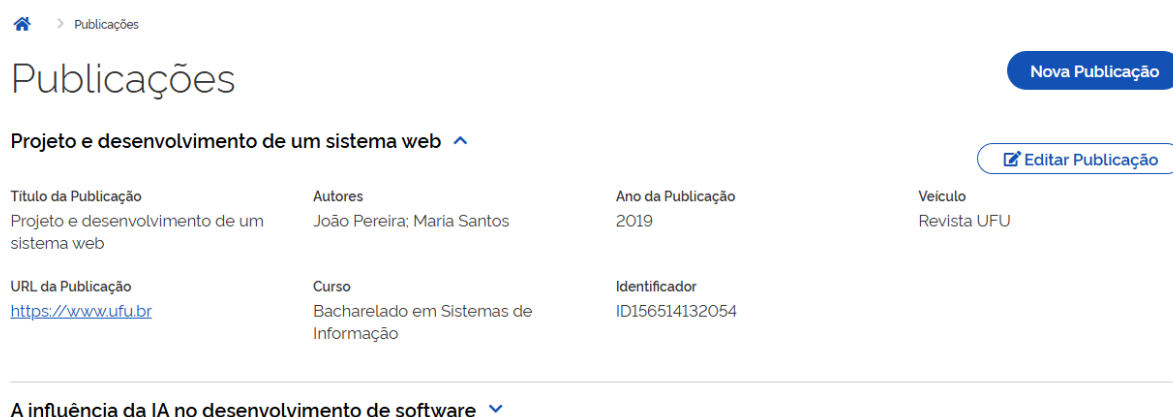
Os botões **Novo Depoimento** e **Editar Depoimento** possibilitam, respectivamente, o registro de um novo depoimento e a edição de depoimentos já cadastrados. Durante o processo de criação de um novo depoimento, é necessário o consentimento do usuário em relação aos termos da **LGPD (Lei Geral de Proteção de Dados)**. Esse consentimento é solicitado por meio de um **popup** de confirmação, cuja representação é apresentada na Figura 7, garantindo a conformidade com as diretrizes de proteção de dados.



Figura 7 – Modal de Consentimento LGPD exibido ao cadastrar um novo depoimento

A Figura 8 ilustra a tela de Publicações, onde é realizado o gerenciamento das publicações acadêmicas ou profissionais do usuário. Cada publicação é exibida com informações detalhadas, incluindo o **Título da Publicação**, **Autores**, **Ano da Publicação**, **Veículo** (revista ou conferência onde foi publicada), **URL da Publicação**, **Curso** associado e um **Identificador** único.

Os botões **Nova Publicação** e **Editar Publicação** possibilitam, respectivamente, o registro de uma nova publicação e a edição das já cadastradas. A estrutura da página utiliza componentes reutilizáveis, proporcionando uma interface consistente com as demais telas do sistema.



Publicações

[Nova Publicação](#)

Projeto e desenvolvimento de um sistema web ^

[Editar Publicação](#)

Título da Publicação	Autores	Ano da Publicação	Veículo
Projeto e desenvolvimento de um sistema web	João Pereira; Maria Santos	2019	Revista UFU

URL da Publicação	Curso	Identificador
https://www.ufu.br	Bacharelado em Sistemas de Informação	ID156514132054

A influência da IA no desenvolvimento de software v

Figura 8 – Tela de Publicações

A Figura 9 ilustra a tela de Informações Pessoais, na qual são exibidos os dados pessoais do usuário, permitindo a visualização e edição dessas informações. Entre os dados apresentados estão: **Nome**, **Nome Social**, **CPF**, **E-mail Principal**, **E-mail Secundário**, **Telefone Principal**, **Telefone Secundário**, **Link do ORCID**, **Link do LinkedIn** e **Link do Lattes**.

A edição dos dados pode ser realizada por meio do botão **Editar Informações**, com exceção de campos como **Nome** e **CPF**, que são considerados dados fixos provenientes da base da faculdade e, portanto, não podem ser alterados diretamente pelo usuário.




Informações Pessoais

[Editar Informações](#)

Nome José da Silva Pereira	Nome Social José da Silva	CPF 123.456.789-00	E-mail josesilvapereira@email.com
E-mail Secundário josesilvapereira01@email.com	Telefone (34) 99999-9999	Telefone Secundário (34) 99999-8888	Link do Lattes http://lattes.cnpq.br/12345678900
Link do ORCID http://orcid.com/josedasilva	Link do LinkedIn http://linkedin.com/in/josedasilva		

Figura 9 – Tela de Informações Pessoais

A interface foi projetada para facilitar o acesso rápido às informações e inclui links para perfis externos, como ORCID, LinkedIn e Lattes, que podem ser importantes no contexto acadêmico e profissional.

 > Egressos

Listagem de Egressos

Filtros ^

Nome do Egresso <input type="text"/>	CPF <input type="text"/>	Campus <input type="text" value="Selecione o item"/>	Nome ou Código do Curso <input type="text"/>
Titulação <input type="text" value="Selecione o item"/>	Data de Ingresso <input type="text" value="00/00/0000"/>	Data de Conclusão <input type="text" value="00/00/0000"/>	<input type="button" value="Limpar"/> <input type="button" value="Pesquisar"/>

Egressos						
Nome do Egresso	Nome do Curso	Nível do Curso	Campus	Data Ingresso	Data Conclusão	Ação
Ana Maria Silva	Ciência da Computação	Bacharelado	Santa Mônica	10/02/2018	15/12/2022	
Carlos Eduardo Ferreira	Engenharia Elétrica	Graduação	Umuarama	05/03/2017	30/11/2021	
Fernanda Souza	Administração	MBA	Monte Carmelo	20/08/2019	25/06/2023	
João Pedro Oliveira	Direito	Licenciatura	Patos de Minas	15/01/2016	10/12/2020	
Mariana Lima	Psicologia	Especialização	Pontal	01/07/2015	20/11/2019	
Lucas Almeida	Medicina	Mestrado	Santa Mônica	20/02/2014	30/01/2020	
Bruna Rocha	Design Gráfico	Curso Tecnólogo	Umuarama	15/03/2017	10/12/2021	
Rafael Sousa	Arquitetura e Urbanismo	Doutorado	Monte Carmelo	25/08/2016	05/02/2021	
Larissa Martins	Jornalismo	Graduação	Pontal	10/01/2018	01/12/2022	
Felipe Costa	Economia	Bacharelado	Santa Mônica	20/07/2019	30/05/2023	

Exibir 10 1-10 de 20 itens Página 1

Figura 10 – Tela de Listagem de Egressos

A Figura 10 ilustra a tela de Listagem de Egressos, na qual coordenadores e diretores podem visualizar e gerenciar as informações dos egressos de seus respectivos cursos. A página apresenta um conjunto de filtros para facilitar a busca por ex-alunos, incluindo **Nome do Egresso**, **CPF**, **Campus**, **Nome ou Código do Curso**, **Titulação**, **Data de Ingresso** e **Data de Conclusão**. Após o preenchimento dos campos desejados, é possível acionar o botão **Pesquisar** para exibir os resultados que correspondem aos critérios de busca.

No cabeçalho (*header*) da tabela, um menu disponibiliza a opção de **Exportar** os dados dos egressos exibidos. Essa funcionalidade permite a extração das informações para utilização externa, como a elaboração de relatórios ou a realização de análises adicionais, facilitando o compartilhamento e o armazenamento dos dados fora do sistema.

A tabela de egressos exibe colunas com as seguintes informações: - **Nome do Egresso** - **Nome do Curso** - **Nível do Curso** - **Campus** - **Data de Ingresso** - **Data de Conclusão** - **Ação**: um ícone que permite visualizar os detalhes de cada egresso.

No rodapé da tabela, há uma opção de seleção de quantidade de itens por página e navegação entre as páginas de resultados, proporcionando uma visualização organizada e escalável, especialmente útil para listas extensas de egressos.

Essa tela facilita a consulta e análise de dados dos egressos e é uma ferramenta essencial para manter o contato e monitorar o progresso dos ex-alunos, permitindo que o coordenador acompanhe as trajetórias profissionais e acadêmicas dos estudantes após a conclusão de seus cursos.



Figura 11 – Tela de Detalhes do Egresso

A Figura 11 ilustra a tela de Detalhes do Egresso, na qual coordenadores e diretores podem visualizar todas as informações detalhadas cadastradas por um egresso. Os dados são organizados em um componente de abas, o que facilita a navegação e a visualização dos diferentes tipos de informações sem sobrecarregar a interface.

As abas disponíveis na tela incluem:

- **Informações Pessoais**: exibe dados de identificação do egresso, incluindo informações básicas e de contato.
- **Informações Acadêmicas**: apresenta o histórico acadêmico do egresso, como curso, nível, campus, e datas de ingresso e conclusão.
- **Depoimentos**: lista os depoimentos cadastrados pelo egresso, que podem incluir reflexões ou experiências durante o curso.
- **Publicações**: mostra as publicações acadêmicas ou profissionais associadas ao egresso.

Cada aba carrega o conteúdo específico relacionado ao tópico, tornando a visualização das informações mais segmentada e intuitiva. Essa abordagem em abas possibilita uma navegação rápida e eficiente entre diferentes conjuntos de dados, ajudando o coordenador a obter uma visão completa das atividades e contribuições do egresso durante e após a graduação.

4.3 Adequações ao Padrão e-MAG

O sistema foi desenvolvido contemplando o **Modelo de Acessibilidade em Governo Eletrônico (e-MAG)**, que orienta boas práticas de acessibilidade digital no setor público brasileiro. Para atender a essas diretrizes, foram implementadas as seguintes ações:

- **Uso de componentes acessíveis:** A biblioteca de componentes `@govbr-ds/core` foi utilizada, garantindo que botões, formulários e elementos de navegação seguissem padrões de acessibilidade, como foco visível e contraste adequado.
- **Suporte a navegação por teclado:** Todos os componentes interativos foram testados para garantir que fossem totalmente navegáveis utilizando apenas o teclado, respeitando a sequência lógica de navegação (tabindex).
- **Textos alternativos em imagens:** Sempre que imagens foram utilizadas (incluindo ícones e ilustrações), atributos `alt` ou descrições equivalentes foram aplicados para leitores de tela.
- **Labels associados a inputs:** Campos de formulários foram implementados com rótulos (`label`) corretamente vinculados aos seus respectivos elementos de entrada, facilitando a identificação por tecnologias assistivas.
- **Mensagens e validações compreensíveis:** Mensagens de erro e validação foram elaboradas de forma clara e objetiva, permitindo que todos os usuários entendam facilmente o que precisa ser corrigido.
- **Estrutura semântica de HTML:** Elementos semânticos (`section`, `nav`, `header`, `footer`, `main`, etc.) foram utilizados para estruturar as páginas, promovendo uma interpretação adequada por ferramentas de acessibilidade.
- **Alto contraste e legibilidade:** A paleta de cores respeita as exigências mínimas de contraste do e-MAG, favorecendo a leitura e a percepção visual dos elementos.

Essas implementações visam tornar o sistema inclusivo, permitindo seu acesso e uso adequado por pessoas com deficiência, em conformidade com as políticas públicas de acessibilidade.

4.4 Acesso ao Código-Fonte

O código-fonte do sistema desenvolvido está disponível publicamente no repositório do GitHub, podendo ser acessado pelo seguinte link:

[<https://github.com/oluizcarvalho/egress-system>](https://github.com/oluizcarvalho/egress-system)

Este repositório contém todo o material referente ao projeto, incluindo a implementação do frontend e os componentes criados.

5 Conclusão

Este trabalho propôs uma etapa no desenvolvimento de um sistema de acompanhamento de egressos para a Universidade Federal de Uberlândia (UFU), com o objetivo de fortalecer a relação da instituição com seus ex-alunos e facilitar a gestão de informações relacionadas às suas trajetórias acadêmicas e profissionais. O projeto teve ênfase no desenvolvimento do *frontend*, utilizando o framework Angular, seguindo diretrizes do *Design System* do Governo Brasileiro (DsGov) e padrões de acessibilidade, resultando em uma interface modular, responsiva e escalável.

Embora o *backend* não faça parte do escopo deste trabalho, definiu-se a utilização da tecnologia Java para sua implementação. Da mesma forma, a modelagem do banco de dados foi elaborada em sua primeira versão e pode ser ainda aprimorada. Apesar dessas limitações, a estrutura arquitetural projetada permite a integração harmoniosa dessas camadas em trabalhos futuros, garantindo a escalabilidade e robustez do sistema.

Algumas funcionalidades que estavam previstas inicialmente para este trabalho foram direcionadas para desenvolvimentos futuros, devido à complexidade e ao escopo deste projeto. Dentre elas, destacam-se:

- Integração com o sistema de autenticação *Single Sign-On* (SSO) da UFU;
- Extração de dados da plataforma Lattes para alimentar a tela de publicações;
- Moderação de depoimentos submetidos pelos egressos;
- Implementação de um sistema de satisfação de usuários, integrado aos serviços do *Gov.br*.

A adoção de metodologias ágeis, como o Scrum, foi essencial para a organização do trabalho, permitindo a entrega de um *frontend* funcional com base em componentes reutilizáveis e aderentes às diretrizes institucionais. As funcionalidades desenvolvidas incluem telas de informações acadêmicas e profissionais, gerenciamento de depoimentos e publicações, além de um *dashboard* acessível ao público geral. Além de um *dashboard* acessível ao público geral contendo dados públicos sobre o panorama de egressos, incluindo informações quantitativas e estatísticas relevantes.

Como trabalhos futuros, destacam-se a implementação do *backend*, a integração com serviços externos e a construção do banco de dados. Adicionalmente, a evolução do sistema poderá incorporar funcionalidades avançadas, como análise de dados e relatórios para auxiliar na tomada de decisão da universidade.

Este projeto representa um passo importante para a UFU no aprimoramento do relacionamento com seus egressos. A aplicação desenvolvida reforça a relevância do acompanhamento contínuo dos ex-alunos, contribuindo para a avaliação institucional, a qualidade do ensino e a interação com a sociedade. Além disso, a implantação do sistema possui potencial para ser expandida e também adaptada por outras instituições de ensino superior, sendo uma ferramenta estratégica no contexto acadêmico.

Referências

- BRASILEIRO, G. **eMAG - Modelo de Acessibilidade em Governo Eletrônico**. 2014. <<https://emag.governoeletronico.gov.br/#s2.1>>. [Online; accessed 06-Fevereiro-2024]. Citado na página 13.
- _____. **Padrões de Interoperabilidade de Governo Eletrônico - ePING**. 2018. <<https://eping.governoeletronico.gov.br>>. [Online; accessed 06-Fevereiro-2024]. Citado 2 vezes nas páginas 13 e 20.
- _____. **@govbr-ds/core**. 2022. <<https://www.npmjs.com/package/@govbr-ds/core>>. [Online; accessed 26-Abril-2025]. Citado na página 46.
- _____. **Padrão Digital de Governo**. 2022. <<https://www.gov.br/ds/home>>. [Online; accessed 06-Fevereiro-2024]. Citado 2 vezes nas páginas 13 e 29.
- CABRAL, T. L. d. O.; SILVA, F. C. d.; PACHECO, A. S. V. As universidades e o relacionamento com seus ex-alunos: Uma análise de portais online de egressos. **Revista Gestão Universitária na América Latina - GUAL**, Universidade Federal de Santa Catarina, v. 9, n. 3, p. 157–173, 2016. Citado 3 vezes nas páginas 12, 22 e 23.
- CTIC. **Centro de Tecnologia da Informação e Comunicação**. 2014. <<https://ctic.ufu.br>>. [Online; accessed 06-Fevereiro-2024]. Citado 2 vezes nas páginas 13 e 20.
- DBEAVER. **DBeaver Community**. 2024. <<https://dbeaver.io/>>. [Online; accessed 09-Novembro-2024]. Citado na página 17.
- DEVMEDIA. **O que é JSON**. 2011. <<https://www.devmedia.com.br/o-que-e-json/23166>>. [Online; accessed 09-Novembro-2024]. Citado na página 27.
- FIGMA. **FIGMA DESIGN OVERVIEW**. 2024. <<https://www.figma.com/design-overview/>>. [Online; accessed 06-Fevereiro-2024]. Citado na página 13.
- FOLHA. **Ranking de universidades do Brasil**. 2015. <<https://ruf.folha.uol.com.br/2015/>>. [Online; accessed 06-Abril-2024]. Citado na página 23.
- GOOGLE. **Angular**. 2024. <<https://angular.io>>. [Online; accessed 06-Fevereiro-2024]. Citado 2 vezes nas páginas 13 e 16.
- _____. **Lazy-loading feature modules**. 2024. <<https://angular.io/guide/lazy-loading-ngmodules>>. [Online; accessed 10-Março-2024]. Citado 2 vezes nas páginas 16 e 30.
- _____. **Reactive Forms**. 2024. <<https://angular.dev/guide/forms/reactive-forms/>>. [Online; accessed 09-Novembro-2024]. Citado na página 29.
- _____. **Signals**. 2024. <<https://angular.dev/guide/signals>>. [Online; accessed 09-Novembro-2024]. Citado na página 29.
- _____. **SwUpdate**. 2024. <<https://angular.dev/api/service-worker/SwUpdate>>. [Online; accessed 09-Novembro-2024]. Citado na página 27.

_____. **Two-way binding**. 2024. <<https://angular.io/guide/two-way-binding>>. [Online; accessed 10-Março-2024]. Citado na página 16.

GROUP, G. D. **PostgreSQL**. 2024. <<https://www.postgresql.org/>>. [Online; accessed 06-Fevereiro-2024]. Citado 2 vezes nas páginas 13 e 17.

GUEDES, T. **Crie aplicações com Angular: O novo framework do Google**. CASA DO CODIGO., 2017. ISBN 9788555192715. Disponível em: <<https://books.google.com.br/books?id=GN7QDgAAQBAJ>>. Citado na página 16.

GUIMARÃES, M. A. M. **O acompanhamento de egressos como ferramenta de inserção no mercado de trabalho**. 2012. <<https://app.uff.br/riuff/handle/1/20507>>. [Online; accessed 24-Fevereiro-2024]. Citado 2 vezes nas páginas 11 e 12.

IBM. **Swagger**. 2022. <<https://www.ibm.com/docs/pt-br/integration-bus/10.1?topic=apis-swagger>>. [Online; accessed 10-Março-2024]. Citado na página 17.

_____. **O que é Java?** 2024. <<https://www.ibm.com/br-pt/topics/java>>. [Online; accessed 10-Março-2024]. Citado na página 17.

_____. **O que é Java Spring Boot?** 2024. <<https://www.ibm.com/br-pt/topics/java-spring-boot>>. [Online; accessed 10-Março-2024]. Citado na página 17.

_____. **O que é uma API de REST?** 2024. <<https://www.ibm.com/br-pt/topics/rest-apis>>. [Online; accessed 10-Março-2024]. Citado na página 17.

MENDES, F. C. **Sistema de acompanhamento de egressos UFU: levantamento e análise de requisitos**. 77 p. Monografia (Graduação em Sistemas de Informação) — Universidade Federal de Uberlândia, Uberlândia, 2023. Disponível em: <<https://repositorio.ufu.br/handle/123456789/39896>>. Citado 3 vezes nas páginas 12, 22 e 28.

MICROSOFT. **Typescript**. 2024. <<https://www.typescriptlang.org/>>. [Online; accessed 10-Março-2024]. Citado na página 16.

MOZILLA. **API de Armazenamento na Web**. 2024. <https://developer.mozilla.org/pt-BR/docs/Web/API/Web_Storage_API>. [Online; accessed 09-Novembro-2024]. Citado na página 30.

_____. **JavaScript**. 2024. <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. [Online; accessed 10-Março-2024]. Citado na página 16.

_____. **Progressive Web Apps**. 2024. <https://developer.mozilla.org/pt-BR/docs/Web/Progressive_web_apps>. [Online; accessed 09-Novembro-2024]. Citado na página 27.

ORACLE. **Java**. 2024. <<https://www.java.com/pt-BR>>. [Online; accessed 06-Fevereiro-2024]. Citado na página 13.

_____. **MySQL Workbench**. 2024. <<https://www.mysql.com/products/workbench/>>. [Online; accessed 09-Novembro-2024]. Citado na página 28.

- PACHECO, L. S. e A. Ações de acompanhamento de egressos: um estudo das universidades públicas do sul do Brasil. **Revista Brasileira de Ensino Superior**, v. 3, n. 2, p. 94–113, 2017. ISSN 2447-3944. Disponível em: <<https://seer.atitus.edu.br/index.php/REBES/article/view/2023>>. Citado 2 vezes nas páginas 22 e 23.
- PRESSMAN, R. S. **Software engineering: a practitioner's approach**. [S.l.]: McGraw-Hill Education, 2014. ISBN 9780078022128. Citado na página 15.
- REFACTORING.GURU. **Facade**. 2024. <<https://refactoring.guru/pt-br/design-patterns/facade>>. [Online; accessed 09-Novembro-2024]. Citado na página 29.
- SANTOS, M. R.; FIGUEIREDO, R. M. d. C.; GOMES, M. M. F. Evolution of perspectives on government digitalization in Brazil from 2000 to 2023. **REVES - Revista Relações Sociais**, v. 6, n. 4, p. 17777, Dezembro 2023. Disponível em: <<https://periodicos.ufv.br/reves/article/view/17777>>. Citado na página 19.
- SCRUM.ORG. **What is Scrum?** 2024. <<https://www.scrum.org/resources/what-is-scrum>>. [Online; accessed 25-Abril-2024]. Citado na página 18.
- SOFTWARE, S. **OpenAPI Specification**. 2021. <<https://swagger.io/specification/>>. [Online; accessed 17-Março-2024]. Citado na página 17.
- TANENBAUM, A. S.; STEEN, M. V. **Distributed systems: principles and paradigms**. [S.l.]: Pearson Prentice Hall, 2007. ISBN 9780132392273. Citado 2 vezes nas páginas 15 e 16.
- VARGAS, M. L. F. Ensino superior, assistência estudantil e mercado de trabalho: um estudo com egressos da UFMG. **Avaliação: Revista da Avaliação da Educação Superior (Campinas)**, Publicação da Rede de Avaliação Institucional da Educação Superior (RAIES), da Universidade Estadual de Campinas (UNICAMP) e da Universidade de Sorocaba (UNISO)., v. 16, n. 1, p. 149–163, Mar 2011. ISSN 1414-4077. Disponível em: <<https://doi.org/10.1590/S1414-40772011000100008>>. Citado na página 12.

Apêndices

APÊNDICE A – Biblioteca de Componentes Baseados no Design System do Governo Brasileiro (GovBR DS)

Com o objetivo de garantir padronização visual, acessibilidade e conformidade com as diretrizes governamentais, foi desenvolvida uma biblioteca de componentes compartilháveis utilizando a base da biblioteca `@govbr-ds/core`, disponibilizada oficialmente pelo Governo Federal Brasileiro (2022a).

A implementação foi realizada em Angular 19, resultando na criação da biblioteca **govbr-ds-angular**, que fornece componentes, diretivas e pipes que seguem fielmente o Design System do Governo Brasileiro (DsGov). Essa abordagem facilita a criação de aplicações compatíveis com a identidade visual e os padrões de usabilidade definidos para sistemas públicos.

Acesso à Biblioteca

- **Demonstração Online:** <<https://oluizcarvalho.github.io/govbr-ds-angular/>>
- **Repositório no GitHub:** <<https://github.com/oluizcarvalho/govbr-ds-angular>>
- **Publicação no NPM:** <<https://www.npmjs.com/package/govbr-ds-angular>>

Detalhes Técnicos

Durante o desenvolvimento da biblioteca, foi adotado o **Padrão de Fachada (Facade Pattern)**, permitindo o encapsulamento da complexidade dos componentes originais e a criação de uma camada de abstração simplificada para os consumidores da biblioteca. Essa estratégia facilita a manutenção, melhora a organização do código e permite a atualização dos estilos do `@govbr-ds/core` de forma isolada, sem impactar diretamente as aplicações que utilizam a biblioteca.

A utilização do padrão de fachada também contribuiu para garantir que as APIs dos componentes Angular fossem projetadas de maneira mais alinhada ao ecossistema do Angular, utilizando recursos modernos como **Signals** e **Reactive Forms** para otimizar o desempenho e a experiência de desenvolvimento.

APÊNDICE B – Processo Iterativo do Desenvolvimento da Funcionalidade de Depoimentos

Este apêndice detalha as etapas envolvidas no desenvolvimento da funcionalidade de cadastro de depoimentos no sistema de acompanhamento de egressos. A funcionalidade evoluiu ao longo de um processo iterativo, desde a concepção inicial até a implementação final, com ajustes baseados em feedback dos *stakeholders*.

1. Discovery

A etapa inicial de *discovery* consistiu na documentação das primeiras ideias para a funcionalidade. O objetivo era identificar como os usuários poderiam cadastrar depoimentos e como essas informações seriam apresentadas no sistema. Essa fase ajudou a traçar um panorama inicial e guiar as etapas subsequentes. A Figura 12 apresenta um registro dessa etapa documentado no Notion.

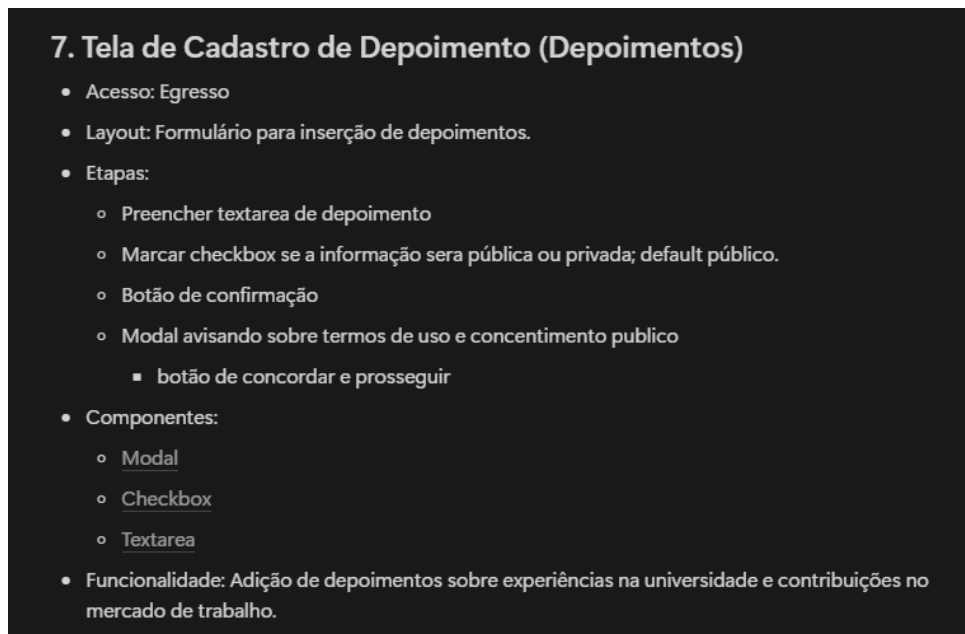


Figura 12 – Etapa de *discovery*: documentação inicial no Notion.

2. Refinamento

Na etapa de refinamento, a funcionalidade foi detalhada em termos de requisitos, interações e elementos visuais. Campos obrigatórios, botões e comportamentos esperados foram definidos para orientar o desenvolvimento e a prototipação. A Figura 13 exibe o documento no Notion que consolidou esses detalhes.

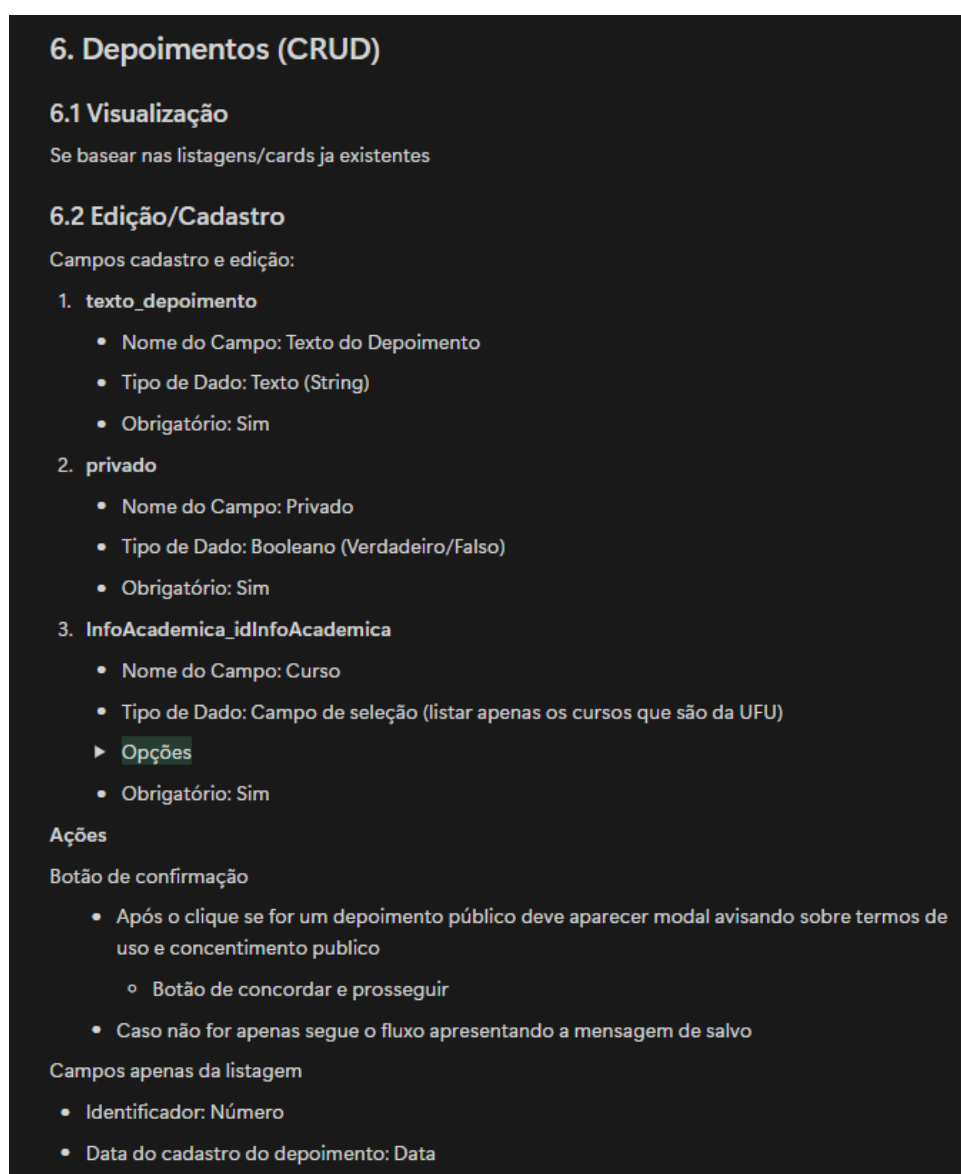


Figura 13 – Detalhamento das funções e layout no Notion.

3. Protótipo de Média Fidelidade

Com as definições claras, foi criado um protótipo de média fidelidade utilizando o Figma. Nesse protótipo, os componentes do DsGov foram aplicados, mas sem foco em acabamentos ou detalhes visuais. Ele serviu como uma base inicial para validações e

ajustes. A Figura 14 ilustra essa etapa.

O protótipo de média fidelidade para a tela 'Cadastrar Depoimento' apresenta uma barra de navegação superior com o ícone de casa e o texto '> Cadastrar Depoimento'. O título da tela é 'Cadastrar Depoimento'. O formulário principal contém um campo de texto grande rotulado 'Texto do Depoimento (Obrigatório)' com um placeholder 'Placeholder'. À direita, há um menu suspenso rotulado 'Curso (Obrigatório)' com o placeholder 'Placeholder', e um toggle switch rotulado 'Privacidade do Depoimento (Obrigatório)' com a opção 'Privado' selecionada. Um botão azul 'Salvar' está no canto inferior direito.

Figura 14 – Protótipo de média fidelidade no Figma.

4. Protótipo de Alta Fidelidade

Com base nos feedbacks recebidos e nos ajustes necessários, o protótipo foi refinado para alta fidelidade. Nesta versão, todos os detalhes visuais e de interação foram finalizados, garantindo que o protótipo servisse como guia para o desenvolvimento da funcionalidade. A Figura 15 apresenta o resultado final dessa etapa.

O protótipo de alta fidelidade para a tela 'Cadastrar Depoimento' apresenta uma barra de navegação superior com o ícone de casa e o texto '> Depoimentos > Cadastrar Depoimento'. O título da tela é 'Cadastrar Depoimento'. O formulário principal contém um campo de texto rotulado 'Texto do Depoimento' com um texto de exemplo Lorem Ipsum. À direita, há um menu suspenso rotulado 'Informações Acadêmicas Relacionadas' com o placeholder 'Selecione um curso', e um toggle switch rotulado 'Privacidade do Depoimento' com a opção 'Privado' selecionada. Um botão azul 'Salvar' está no canto inferior direito.

Figura 15 – Protótipo de alta fidelidade no Figma.

5. Tela Concluída

Após o desenvolvimento, a funcionalidade foi apresentada em uma sessão de *review* com os *stakeholders*. Durante essa reunião, foi solicitado substituir o componente de *toggle* por um *radio button*, resultando na versão final apresentada na Figura 16.

The screenshot shows a web interface for recording a statement. At the top, there is a breadcrumb trail: a home icon, followed by 'Depoimentos', and then 'Novo Depoimento'. Below this is the title 'Cadastrar Informações'. The form is divided into two main sections. The left section, titled 'Texto do Depoimento', contains a text area with placeholder Lorem Ipsum text. Below the text area, it indicates '481 caracteres digitados'. The right section, titled 'Curso Relacionado', features a dropdown menu with the placeholder text 'Selecione o item'. Below the dropdown, there is a 'Rótulo' section with three radio buttons: 'Público', 'Privado', and 'Anônimo'. The 'Anônimo' option is selected. At the bottom right of the form is a blue 'Salvar' button.

Figura 16 – Tela final da funcionalidade de depoimentos após o *review*.

Conclusão

O desenvolvimento da funcionalidade de depoimentos exemplifica como um processo iterativo e colaborativo pode resultar em soluções alinhadas às necessidades dos usuários e aos padrões governamentais. Desde a concepção inicial até a entrega final, as etapas de *discovery*, refinamento, prototipação e validação permitiram melhorias contínuas e garantiram a qualidade do produto.

Anexos

ANEXO A – Documento de Formalização da Demanda

Nesta seção, apresenta-se o Documento de Formalização da Demanda, elaborado e encaminhado pelo professor Dr. Jean Carlos Domingos, que serviu como material fundamental para compreender as expectativas da UFU em relação ao sistema de acompanhamento de egressos.



DOCUMENTO DE FORMALIZAÇÃO DA DEMANDA

Documento formulado com base no art. 10º da IN nº1 de 4 de abril de 2019

Em acordo com o art. 10 da Instrução Normativa SGD/ME nº 1, de 4 de abril de 2019, a fase de Planejamento se iniciará com o recebimento do Documento de Formalização da Demanda (DFD) a ser analisado pela Comissão do Plano Diretor de Tecnologia da Informação e Comunicação (COMPDTIC). Este DFD será avaliado pelo(s) setor(es) envolvido(s) na eventual execução da solicitação para fins de elaboração do novo PDTIC/UFU.

1. IDENTIFICAÇÃO DA ÁREA REQUISITANTE

Unidade ou Órgão Requirante:	Pró-Reitoria de Pesquisa e Pós Graduação - PROPP			
Autoridade da área administrativa:	Carlos Henrique de Carvalho	Cargo/Função:	Pró-Reitor de Pesquisa e Pós graduação	Matrícula
Responsável pela Demanda:	Carlos Henrique de Carvalho	Cargo/Função:	Pró-Reitor de Pesquisa e Pós graduação	Matrícula
E-mail do responsável:	chc@ufu.br			Telefone

2. NECESSIDADE DA UNIDADE/ÓRGÃO

Nome da Meta/Ação: <i>Solução para Gestão de Egressos da UFU (Alumni) - que permita visões por UA/Curso/Programa</i>												
Processos de negócios envolvidos na Meta/Ação: <i>Gestão de egressos da UFU (Alumni)</i> <i>Avaliação dos PPGs (Programas de Pós-graduação) pela CAPES</i>												
Motivo da solicitação: <i>Conforme os indicadores e metas do PIDE 2022-2027, descritos em: http://www.proplad.ufu.br/pide/pide-2022-2027#tab-0-1</i> <i>Faz-se necessário, em Pesquisa e Pós-graduação, no indicador PP02:</i> <i>Indicador: Conceito CAPES médio dos programas de pós-graduação stricto sensu</i> <i>Meta: Elevar o Conceito CAPES médio dos programas de pós-graduação stricto sensu</i> <i>É importante destacar que no documento de área da CAPES e a ficha de avaliação de área da CAPES há pontuação pelo acompanhamento dos egressos nos PPGs. Por isso é muito institucional, para isso, pois, ajudará amplamente todos os PPGs a melhorar o conceito CAPES, e atingir o indicador PP02 do PIDE 2022-2027.</i> <i>Sugere-se, inicialmente, entregar uma Fase I para teste em 1 até 3 PPGs. Após isto e a validação, então escalar para todos PPGs.</i>												
Prazo(s): <i>Até Dez/2022: Fase I para teste em 1 até 3 PPGs.</i> <i>Até Jul/2023: Fase II implantada em todos PPGs da UFU</i> <i>Até Dez/2023: Fase III implantada nos cursos de graduação da UFU</i>												
Problema a ser resolvido: <i>O problema a ser resolvido é a gestão de egressos da UFU, conforme os indicadores da CAPES descritos nos documentos de área e fichas de avaliação de área.</i> <i>Isto será de extrema importância para a PROPP entregar os resultados planejados Nos indicadores e metas do PIDE 2022-2027, descritos em:</i> <i>http://www.proplad.ufu.br/pide/pide-2022-2027#tab-0-1</i> <i>Importante: Esta ferramenta deverá permitir à PROPP gestionar a evolução dos indicadores da CAPES descritos nos documentos de área e fichas de avaliação de área para todos P</i>												
Setor(es) envolvido(s) na execução da Meta/Ação: <i>CTIC, PROPP e todos PPGs</i>												
Finalidade:	<input checked="" type="checkbox"/>	Administração	<input checked="" type="checkbox"/>	Ensino	<input checked="" type="checkbox"/>	Pesquisa	<input type="checkbox"/>	Extensão	<input type="checkbox"/>	Cultura	<input type="checkbox"/>	Outro: Pós Graduação, Tecnologia e Inova

3. DA ORIGEM DOS RECURSOS

Sua/seu unidade/órgão já possui Analista de Tecnologia da Informação (ATI) e/ou Técnico de Tecnologia da Informação (TTI) para execução desta Meta/Ação? Se sim, quantos? <i>Não.</i>									
Sua/seu unidade/órgão pretende converter vaga(s) da Carreira de Cargos Técnicos-Administrativos em Educação (TAE) em vaga(s) de ATI e/ou TTI para execução das Metas/Ações, quantos ATIs e TTIs? <i>Não.</i>									
Caso a Meta/Ação demande envolvimento do CTIC, os ATIs e TTIs de sua/seu unidade/órgão, necessários para execução da Meta/Ação, poderão estar em exercício no CTIC com sua unidade/órgão? <i>Não se aplica.</i>									
Informe a estimativa de aporte financeiro necessário para execução da Meta/Ação: <i>É necessário que a área de tecnologia (CTIC) faça a estimativa orçamentária. A PROPP não possui conhecimento técnico suficiente para estimativa do custo.</i>									

Fonte dos recursos:		Orçamentário UFU		Extraorçamentário UFU	X	Orçamentário da Unidade Gestora	X	Extraorçamentário da Unidade Gestora		Não se aplica		Outro: <especifique>
---------------------	--	------------------	--	-----------------------	---	---------------------------------	---	--------------------------------------	--	---------------	--	----------------------

4. ALINHAMENTO AO PDTIC ANTERIOR (BIÊNIO 2021-2022)

Grupo da Meta/Ação	ID da Meta/Ação	Descrição da Meta/Ação do PDTIC 2021-2022

Link para o PDTIC 2021-2022: <http://www.reitoria.ufu.br/Resolucoes/resolucaoCONDIR-2021-4.pdf>

5. ALINHAMENTO AO PIDE VIGENTE OU PREVISÃO LEGAL

Dispositivo Legal:	<Informar no caso da necessidade ser para o cumprimento de algum dispositivo legal (leis, decretos, instruções normativas...)>	
Eixo Temático	ID da Meta	Descrição da Meta do PIDE
Pesquisa e Pós-Graduação	PP02	Elevar o Conceito CAPES médio dos programas de pós-graduação stricto ser

Link para o PIDE 2022-2027: <http://www.proplad.ufu.br/pide/pide-2022-2027#tab-0-1>

6. FUNCIONALIDADES TÉCNICAS DESEJADAS

Interface e/ou ambiente de uso: <i>O ambiente será utilizado em diferentes visões:</i> 1- Nível do coordenador do Programa de Pós Graduação 2- Nível da PROPP 3- Nível PROGRAD 4- Nível UFU Desta forma, é necessário que o sistema forneça a visão de gestão apropriada para cada nível.
Funcionalidades necessárias: As funcionalidades necessárias são: 1- Na Fase I e II, fazer o acompanhamento dos egressos conforme especificado nos documentos de área e fichas de avaliação de área, da CAPES, para os PPGs. 2- Na Fase III, fazer o acompanhamento dos egressos da graduação, conforme for relevante para a avaliação MEC dos cursos de graduação.
Integração com outras soluções de TIC: A ferramenta escrita neste documento de formalização da demanda precisa integrar com toda a cadeia da UFU utilizada na graduação e pós-graduação. Por exemplo, precisa integrar com o sistema de gestão de estudantes da pós-graduação, para ter uma base única de cadastro de estudantes, assim, a base de estudantes precisa s à solução desta demanda.
Pré-requisitos da Meta/Ação: Disponibilidade de acesso ao banco de dados. Dashboard com visão para os coordenadores de graduação e programas de pós-graduação gerenciar os egressos sob sua gestão Dashboard com visão para a pró-reitoria de pesquisa e pós graduação gerenciar os egressos dos PPGs Dashboard com visão para a pró-reitoria de graduação gerenciar os egressos da graduação Dashboard com visão para a UFU gerenciar todos seus egressos
Aspectos Legais: Todos aspectos respectivos à LGPD (Lei Geral de Proteção de Dados), pois envolve informações de egressos (pessoas) e, assim, envolve informações que podem ser sensíveis.

7. DESCRIÇÃO DO PROCESSO ATUAL

Atualmente este processo é realizado manualmente por cada PPG, sem um padrão institucional para acompanhamento e gestão.

8. RESULTADOS A SEREM ALCANÇADOS

Alcançar o indicador PP02 da Diretriz Estratégica do PIDE 2022 - 2027.

9. VERSÃO ATUALMENTE DISPONÍVEL NA UFU

Informe qual o equipamento/sistema/website (pode ser mais de um): A nível da PROPP não há equipamento/sistema/website disponível em uso.
Justifique porque o equipamento/sistema/website atual não atende: N.A.

10. CATEGORIAS DE USUÁRIOS QUE SERÃO AFETADAS

X	Gestores UFU	X	Discentes de graduação
	Docentes	X	Discentes de pós-graduação
	Técnicos		Discentes ESTES
	Terceiros		Discentes ESEBA
	Residentes		Outro:

11. **OBSERVAÇÕES GERAIS**

Uberlândia, 04 de agosto de 2022

CARLOS HENRIQUE DE CARVALHO
Pró-Reitor de Pesquisa e Pós-Graduação
Portaria R 065/2017



Documento assinado eletronicamente por **Carlos Henrique de Carvalho, Pró-Reitor(a)**, em 05/08/2022, às 15:18, 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 **3814864** e o código CRC **654CDB76**.