

**UTFPR-UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

*Bacharelado em Engenharia de Software - 6º Período*

**DISCIPLINA:** *Oficina de Integração 1 - ES46F-ES61*

**PROFESSOR:** *Eduardo Cotrin Teixeira*

---

# **Documento de Projeto de Software**

---

## **Portal TEDI**

Álison Christian Rebouças Vidal de Carvalho

João Marcos Ribeiro da Costa

João Pedro Correia Leite Moreira

Luccas Philot Gonçalves

Yuri Garcia Yoshida

*Cornélio Procópio*

2025

# Sumário

<b>1 INTRODUÇÃO .....</b>	<b>2</b>
1.1 CONTEXTO .....	2
1.2 JUSTIFICATIVA .....	2
1.3 PROPOSTA .....	3
1.4 ORGANIZAÇÃO DO DOCUMENTO .....	3
<b>2 DESCRIÇÃO GERAL DO SISTEMA.....</b>	<b>4</b>
2.1 OBJETIVOS (GERAIS E ESPECÍFICOS).....	4
2.2 LIMITES E RESTRIÇÕES .....	4
2.3 DESCRIÇÃO DOS USUÁRIOS DO SISTEMA.....	4
<b>3 DESENVOLVIMENTO DO PROJETO .....</b>	<b>5</b>
3.1 TECNOLOGIAS E FERRAMENTAS.....	5
<i>Ambiente de Desenvolvimento</i> .....	5
<i>Tecnologias Utilizadas</i> .....	5
<i>Ferramentas de Apoio</i> .....	6
3.2 METODOLOGIA DE DESENVOLVIMENTO .....	6
<i>Papéis no Scrum</i> .....	6
<i>Adaptação do Scrum no Projeto</i> .....	7
3.3 CRONOGRAMA PREVISTO .....	7
<b>4 REQUISITOS DO SISTEMA.....</b>	<b>9</b>
4.1 REQUISITOS FUNCIONAIS.....	9
<i>Descrição dos Requisitos Funcionais</i> .....	9
4.2 REQUISITOS NÃO-FUNCIONAIS .....	10
4.3 DIAGRAMAS DE CASOS DE USO .....	11
4.4 PROTÓTIPOS DE TELAS.....	11
<b>5 ANÁLISE DO SISTEMA .....</b>	<b>14</b>
5.1 MODELO DO BANCO DE DADOS .....	14
<i>Dicionário de Dados</i> .....	15
5.2 DIAGRAMA DE CLASSES.....	17
5.3 DIAGRAMA DE ATIVIDADES.....	18
<b>6 IMPLEMENTAÇÃO .....</b>	<b>20</b>
6.1 DESCRIÇÃO DO CÓDIGO .....	20
<i>Diagramas da Arquitetura</i> .....	20
6.2 IMPLANTAÇÃO .....	22
<i>Requisitos Necessários</i> .....	22
<i>Execução do Backend</i> .....	23
<i>Execução do Frontend</i> .....	23
6.3 TELAS PRINCIPAIS.....	24
<i>Telas Públicas</i> .....	24
<i>Telas Administrativas</i> .....	26
<b>7 CONSIDERAÇÕES FINAIS .....</b>	<b>28</b>
<b>8 BIBLIOGRAFIA .....</b>	<b>29</b>

# 1 Introdução

## 1.1 Contexto

O TEDI (Tecnologia e Educação Digital para a Terceira Idade) é um projeto de extensão da UTFPR – Campus Cornélio Procopio, criado em 2024, com o objetivo de promover a inclusão digital da população idosa por meio de oficinas e cursos gratuitos. A iniciativa nasceu da necessidade urgente de combater a exclusão tecnológica que afeta a terceira idade, dificultando seu acesso a recursos digitais, como celulares, computadores e serviços online. Ao proporcionar familiaridade com essas tecnologias, o projeto busca resgatar a autonomia, a autoestima e a integração social dos idosos.

O projeto é organizado por professores da UTFPR e executado com o apoio de voluntários — alunos da faculdade — que recebem orientação para atuar com metodologias acessíveis ao público idoso. Esses voluntários ministram oficinas práticas e lúdicas com paciência, respeito e foco na troca intergeracional de saberes, promovendo a construção de vínculos entre jovens e idosos.

As atividades são estruturadas em dois módulos independentes:

- Informática Básica, que inclui noções fundamentais sobre o uso de computadores, periféricos e navegação;
- Smartphones e Aplicativos, voltado ao uso de celulares, redes sociais, ferramentas de comunicação e serviços digitais essenciais.

Além de promover a inclusão digital, o TEDI contribui para a autoestima, o bem-estar e a cidadania da pessoa idosa, fortalecendo sua presença ativa na sociedade contemporânea. Atualmente, o projeto também desenvolve materiais didáticos próprios e estuda a expansão das ações, com o objetivo de oferecer capacitações mais completas e duradouras.

## 1.2 Justificativa

Embora o TEDI já desenvolva ações importantes voltadas à inclusão digital de idosos, ainda carece de uma plataforma online que concentre suas informações, divulgue suas atividades e facilite a participação de novos colaboradores.

A criação de um site institucional visa preencher essa lacuna, oferecendo um meio acessível para que o público conheça o projeto, acompanhe notícias e, principalmente, que voluntários possam se inscrever com facilidade — o que é essencial para a continuidade do trabalho, já que o funcionamento do TEDI depende diretamente dessa participação.

Atualmente, a divulgação e o recrutamento são feitos de forma descentralizada, o que dificulta o alcance e a organização das ações. Um site bem estruturado permitirá maior visibilidade, comunicação eficiente e apoio à expansão do projeto.

## 1.3 Proposta

Com o intuito de apoiar o projeto TEDI, será desenvolvido um site que visa facilitar o acesso às informações do projeto de forma simples e acessível para todos os públicos, especialmente os idosos.

A aplicação permitirá a visualização de notícias e informativos cadastrados pelos administradores do projeto, garantindo que os usuários estejam sempre atualizados sobre oficinas, cursos e demais atividades. Além disso, contará com uma página de formulário destinada ao cadastro de monitores voluntários, promovendo a ampliação da equipe de apoio. A interface será responsiva, acessível e projetada para uma navegação intuitiva, respeitando as diretrizes de acessibilidade.

## 1.4 Organização do Documento

Este documento apresenta, de forma estruturada, todas as etapas do desenvolvimento da aplicação web de apoio ao projeto TEDI. Inicialmente, o Capítulo 1 introduz o contexto, a motivação e os objetivos do projeto, além de justificar a necessidade da solução proposta. O Capítulo 2 descreve o sistema a ser desenvolvido, incluindo os objetivos gerais e específicos, suas limitações e o perfil dos usuários. Em seguida, o Capítulo 3 aborda o desenvolvimento da aplicação, detalhando as tecnologias adotadas, a metodologia de trabalho e o cronograma previsto. O Capítulo 4 é dedicado a apresentar os requisitos do sistema, tanto os funcionais quanto os não funcionais, complementados por diagramas de caso de uso e protótipos das interfaces. Posteriormente, o Capítulo 5 discute a análise do sistema, enquanto o Capítulo 6 trata da sua implementação, com a descrição do código, processo de implantação e as telas principais. O Capítulo 7 reúne as considerações finais sobre o projeto e, por fim, o Capítulo 8 apresenta as referências utilizadas.

## 2 Descrição Geral do Sistema

### 2.1 Objetivos (Gerais e Específicos)

Desenvolver uma aplicação web acessível, intuitiva e centrada no usuário idoso, com o intuito de divulgar e apoiar a administração do projeto TEDI, promovendo a inclusão digital e social.

Objetivos Específicos:

- Permitir a visualização clara e simplificada de informações sobre oficinas, cursos e demais atividades do TEDI;
- Disponibilizar atualizações, comunicados e notícias relacionadas ao projeto;
- Oferecer um formulário acessível para o cadastro de monitores voluntários;
- Garantir que a interface da aplicação atenda às diretrizes de acessibilidade digital, considerando as limitações do público idoso;
- Facilitar o gerenciamento de conteúdo por parte da equipe organizadora do projeto.

### 2.2 Limites e Restrições

A aplicação será desenvolvida exclusivamente para ambiente web, priorizando compatibilidade com os principais navegadores modernos. Inicialmente, não contará com funcionalidades como inscrição automática em cursos, interação entre usuários ou integração com redes sociais. A aplicação exigirá conexão com a internet e seguirá as diretrizes da WCAG 2.2, mas sua experiência pode variar conforme o dispositivo utilizado e o nível de familiaridade tecnológica do usuário.

### 2.3 Descrição dos Usuários do Sistema

O sistema será utilizado por diferentes perfis, cada um com objetivos e necessidades específicas:

- Alunos: interessados em acompanhar notícias e informações gerais sobre o projeto;
- Idosos: público-alvo dos ofícios e cursos, acessam conteúdos de maneira simplificada e acessível;
- Monitores: voluntários que se cadastram para auxiliar nas atividades do TEDI;
- Parceiros: pessoas ou instituições que buscam informações para possíveis colaborações;
- Equipe do Projeto: responsáveis pela administração do conteúdo, cadastros e manutenção das informações exibidas.

## 3 Desenvolvimento do Projeto

### 3.1 Tecnologias e ferramentas

#### Ambiente de Desenvolvimento

A aplicação será desenvolvida no **Visual Studio Code (VS Code)**, um ambiente de desenvolvimento integrado (IDE) gratuito, leve e muito utilizado por programadores. Ele oferece recursos que facilitam a escrita de código, como destaque de sintaxe, sugestões automáticas, controle de versões com Git, depuração integrada e grande variedade de extensões. Além disso, é compatível com várias linguagens, como TypeScript e JavaScript, que serão utilizadas no projeto.

#### Tecnologias Utilizadas

##### Frontend: Angular com TypeScript

A interface do usuário será construída com o **Angular**, um **framework** desenvolvido pelo Google. Um **framework** é um conjunto de ferramentas e regras prontas que facilitam o desenvolvimento de aplicações. Ele fornece estrutura, organização e componentes reutilizáveis, o que torna o processo mais ágil e padronizado.

O Angular é especialmente útil para criar **aplicações web dinâmicas** e de fácil manutenção. Ele funciona em conjunto com o **TypeScript**, uma linguagem que se baseia no JavaScript, mas adiciona recursos como tipagem estática. Isso significa que muitos erros podem ser detectados antes mesmo da execução do código, o que aumenta a segurança e a clareza no desenvolvimento.

##### Backend: JavaScript com Express

No lado do servidor, será usado **JavaScript** com o **framework Express**. O **Express** é um conjunto de bibliotecas que simplifica a criação de servidores e rotas em aplicações web. Ele permite definir facilmente o que o sistema deve fazer ao receber requisições (como salvar dados ou exibir uma página).

O Express é ideal para desenvolver **APIs RESTful**, que são conjuntos de regras para permitir que diferentes sistemas se comuniquem de forma padronizada. Essas APIs funcionam como "pontes" entre o frontend e o backend, usando operações simples (como buscar, adicionar ou excluir dados) através da internet.

##### Banco de Dados: PostgreSQL

O sistema utilizará o **PostgreSQL** como banco de dados, que é uma solução gratuita, robusta e amplamente reconhecida por sua confiabilidade. Ele permite armazenar, consultar e organizar grandes volumes de dados com segurança.

O PostgreSQL também oferece suporte a **transações ACID**, um conjunto de regras que garante que as operações realizadas no banco sejam **atômicas, consistentes, isoladas e duráveis**. Em outras palavras, isso significa que os dados não correm o risco de ficarem corrompidos ou inconsistentes, mesmo se ocorrer uma falha no meio de uma operação, como durante um cadastro ou exclusão.

## Ferramentas de Apoio

### Figma

As telas do sistema foram prototipadas no **Figma**, uma ferramenta online muito utilizada para o design de interfaces. Ela permite que todos os membros da equipe visualizem, comentem e colaborem em tempo real, mesmo à distância. Com ela, foi possível construir os protótipos de forma clara, garantindo que o visual do site fosse acessível, intuitivo e adequado ao público idoso.

### Trello (Kanban)

Para organizar as tarefas e o progresso do projeto, foi adotado o **Trello**, uma ferramenta de gerenciamento de atividades baseada no método **Kanban**. O Kanban utiliza colunas (como “A Fazer”, “Em Andamento” e “Concluído”) e cartões visuais que representam cada tarefa.

Essa organização ajuda a equipe a saber exatamente o que está sendo feito, por quem, e o que ainda precisa ser entregue, garantindo transparência e controle durante toda a execução do projeto.

### Astah UML

Os diagramas utilizados para representar o sistema (como os diagramas de classes, atividades e casos de uso) foram criados com o **Astah UML**, um software especializado em modelagem visual. Ele facilita a representação gráfica de como o sistema deve se comportar, como os dados se relacionam e quais são os fluxos de ações, ajudando na comunicação entre os membros da equipe e na fase de codificação.

### Canais de Comunicação: WhatsApp e Discord

A comunicação entre os integrantes é feita principalmente através do **WhatsApp**, usado para atualizações rápidas, e do **Discord**, utilizado para reuniões por chamada de voz e compartilhamento de tela durante sessões de desenvolvimento. Esses canais facilitam a interação fora do ambiente presencial, mantendo todos atualizados e conectados.

## 3.2 Metodologia de desenvolvimento

O projeto adotará a metodologia **Scrum**, um modelo ágil amplamente utilizado no desenvolvimento de software. O Scrum organiza o trabalho em ciclos curtos e fixos, chamados de Sprints, que geralmente duram de uma a quatro semanas. Em cada sprint, a equipe planeja, desenvolve e entrega uma nova versão parcial do sistema, sempre com funcionalidades reais que podem ser avaliadas.

Esse modelo torna o processo mais eficiente, adaptável e colaborativo, pois permite que o sistema seja construído de forma incremental. Isso significa que o produto evolui por meio de entregas frequentes, cada uma representando uma parte já funcional e testável. Com isso, é possível receber feedback constante, corrigir problemas com agilidade e garantir que o projeto atenda de forma mais precisa às necessidades do cliente ou usuário final.

### Papéis no Scrum

O Scrum define papéis específicos dentro da equipe:

- **Product Owner (PO):** é a pessoa responsável por representar os interesses do cliente, priorizar as tarefas e validar o que foi entregue.  
No projeto TEDI, o PO é **Álison Christian Rebouças Vidal de Carvalho**.
- **Scrum Master:** atua como facilitador do time, garantindo que o processo do Scrum seja seguido e que não haja obstáculos impedindo o trabalho da equipe.  
No grupo, esse papel é desempenhado por **João Marcos Ribeiro da Costa**.
- **Desenvolvedores:** são os membros responsáveis pela implementação prática das funcionalidades.  
São eles: **João Pedro Correia Leite Moreira, Luccas Philot Gonçalves e Yuri Garcia Yoshida**.

## Adaptação do Scrum no Projeto

O Scrum tradicional prevê **reuniões diárias** chamadas de **Daily Meetings** (ou "dailies"), mas para adaptar o método à realidade acadêmica do grupo, a equipe optou por se reunir **duas vezes por semana**:

- **Quartas-feiras (presencial):**  
Essa reunião acontece durante a aula e tem a função de **Sprint Review** (revisar a entrega da sprint anterior) e **Retrospective** (avaliar o que funcionou bem ou precisa melhorar).
- **Domingos (online via Discord):**  
É usada como **Sprint Planning**, onde são definidas as metas da próxima semana e distribuídas as tarefas no quadro Kanban do Trello.

Essa adaptação mantém o foco na entrega contínua e organizada, mesmo com o tempo reduzido de encontro entre os membros. Cada tarefa é associada a uma sprint semanal, o que garante avanço constante no desenvolvimento.

## 3.3 Cronograma previsto

Tarefa	Entrega	Responsáveis
<b>Itens 1 a 4 da Documentação</b>	18/04	Álison Carvalho e João Marcos
<b>Roteiro do Vídeo 1</b>	18/04	Álison Carvalho e João Marcos
<b>Estrutura inicial - Front</b>	21/04	Yuri Yoshida
<b>Estruturação do Banco de Dados</b>	21/04	Yuri Yoshida
<b>Login - Front</b>	21/04	Yuri Yoshida e Luccas Philot
<b>Login - Back</b>	21/04	Luccas Philot
<b>Página de Notícias (admin) - Front</b>	28/04	Yuri Yoshida e Luccas Philot
<b>Página de Notícias (admin) - Back</b>	28/04	João Pedro Moreira e Luccas Philot
<b>Página Inicial - Front</b>	05/05	Yuri Yoshida e Luccas Philot
<b>Página Inicial - Back</b>	05/05	João Pedro Moreira e Luccas Philot
<b>Página de Monitores (admin) - Front</b>	12/05	Yuri Yoshida e Luccas Philot
<b>Página de Monitores (admin) - Back</b>	12/05	João Pedro Moreira e Luccas Philot
<b>Item 5 da Documentação</b>	18/05	Álison Carvalho e João Marcos
<b>Roteiro do Vídeo 2</b>	18/05	Álison Carvalho e João Marcos
<b>Página de Equipe (admin) - Front</b>	22/05	Yuri Yoshida e Luccas Philot
<b>Página de Equipe (admin) - Back</b>	22/05	João Pedro Moreira e Luccas Philot



<b>Tarefa</b>	<b>Entrega</b>	<b>Responsáveis</b>
<b>Página de Notícias - Front</b>	26/05	Yuri Yoshida e Luccas Philot
<b>Página de Notícias - Back</b>	26/05	João Pedro Moreira e Luccas Philot
<b>Página Específica de Notícia - Front</b>	02/06	Yuri Yoshida e Luccas Philot
<b>Página Específica de Notícia - Back</b>	02/06	João Pedro Moreira e Luccas Philot
<b>Página de Monitores - Front</b>	09/06	Yuri Yoshida e Luccas Philot
<b>Página de Monitores - Back</b>	09/06	João Pedro Moreira e Luccas Philot
<b>Página de Equipe - Front</b>	16/06	Yuri Yoshida e Luccas Philot
<b>Página de Equipe - Back</b>	16/06	João Pedro Moreira e Luccas Philot
<b>Itens 6 a 8 da Documentação</b>	25/06	Álison Carvalho e João Marcos
<b>Roteiro do Vídeo Final</b>	25/06	Álison Carvalho e João Marcos

## 4 Requisitos do Sistema

### 4.1 Requisitos Funcionais

Código	Descrição	Prioridade
RF01	O usuário deve ser capaz de cadastrar monitores com RA e CPF únicos, impedindo duplicações.	Essencial
RF02	O administrador deve ser capaz de realizar o login e logout na plataforma.	Essencial
RF03	O administrador deve ser capaz de gerenciar monitores (visualizar e excluir).	Essencial
RF04	O administrador deve ser capaz de copiar os e-mails de todos os monitores cadastrados com apenas um clique.	Importante
RF05	O administrador deve ser capaz de gerenciar notícias (cadastrar, editar e excluir).	Essencial
RF06	O sistema deve exibir uma página pública com o feed de notícias.	Essencial
RF07	Na página inicial, o sistema deve exibir miniaturas das últimas notícias.	Desejável
RF08	O usuário deve ser capaz de aplicar filtros na página de notícias por data, palavra-chave e categoria.	Importante
RF09	O sistema deve exibir uma página com informações sobre a equipe (foto, nome e função).	Importante
RF10	O administrador deve ser capaz de gerenciar membros da equipe (cadastrar, editar e excluir).	Essencial
RF11	O sistema deve disponibilizar um formulário para entrar em contato com a equipe.	Importante
RF12	O sistema deve exibir uma barra de navegação fixa visível em todas as páginas.	Essencial
RF13	O administrador deve ser capaz de filtrar informações na página de notícias, de monitores e de equipe.	Importante
RF14	O administrador pode adicionar os usuários cadastrados (monitores e membros da equipe) a um grupo.	Desejável

Tabela 01: Requisitos Funcionais do Sistema

### Descrição dos Requisitos Funcionais

- **RF01 – Cadastro de Monitores:**  
O sistema deve permitir que alunos da UTFPR se cadastrem como monitores preenchendo um formulário com nome, RA, CPF e e-mail. O sistema deve validar RA e CPF para evitar duplicações.
- **RF02 – Login e Logout:**  
O sistema deve permitir que o administrador acesse as funcionalidades restritas com login e senha. O botão de logout deve estar visível nas páginas protegidas.
- **RF03 – Gerenciamento de Monitores:**  
O sistema deve exibir uma lista dos monitores cadastrados, permitindo que o administrador visualize ou exclua os dados.
- **RF04 – Cópia de E-mails:**  
O sistema deve oferecer um botão para copiar os e-mails dos monitores já cadastrados, facilitando a comunicação por fora da plataforma.
- **RF05 – Gestão de Notícias:**

O sistema deve permitir ao administrador publicar, editar e excluir notícias com título, corpo e imagens, com exibição automática no site.

- **RF06 – Feed de Notícias:**  
O sistema deve exibir uma página com o feed público de notícias, ordenadas por data, acessível a todos os visitantes.
- **RF07 – Carrossel de Notícias na Página Inicial:**  
O sistema deve exibir na página inicial um carrossel automático com miniaturas, título e imagem das últimas notícias.
- **RF08 – Filtro de Notícias:**  
O sistema deve permitir filtrar notícias por data, palavras-chave e categorias na página de notícias.
- **RF09 – Página da Equipe:**  
O sistema deve exibir uma página com nome, foto e função dos membros da equipe do projeto.
- **RF10 – Gestão da Equipe:**  
O sistema deve permitir que o administrador cadastre, edite e exclua informações sobre os membros da equipe.
- **RF11 – Formulário de Contato:**  
O sistema deve disponibilizar uma área com formulário de contato, enviando as mensagens para a equipe do projeto.
- **RF12 – Barra de Navegação:**  
O sistema deve conter uma barra de navegação fixa no topo do site com acesso a todas as páginas.
- **RF13 – Filtro de Informações Administrativas:**  
O sistema deve permitir que o administrador filtre informações nas páginas de notícias, monitores e equipe para facilitar a busca e o gerenciamento.
- **RF14 – Criação de Grupos:**  
O sistema deve permitir que o administrador crie grupos de usuários, de forma que ele possa copiar os e-mails de um determinado grupo.

## 4.2 Requisitos Não-funcionais

Código	Descrição	Categoria	Prioridade
RNF01	O sistema deve ser responsivo, funcionando bem em dispositivos móveis e desktops.	Usabilidade	Essencial
RNF02	O sistema deve ser hospedado em domínio e servidor da própria faculdade.	Ambiente	Essencial
RNF03	O sistema deve ser compatível com navegadores modernos (Chrome, Firefox, Edge).	Compatibilidade	Importante
RNF04	O sistema deve armazenar a senha do único administrador de forma criptografada.	Segurança	Essencial
RNF05	O sistema deve implementar validações para evitar entradas inválidas e ataques como XSS e SQL Injection.	Segurança	Importante
RNF06	O sistema deve seguir princípios de acessibilidade (como contraste, navegação por teclado e textos alternativos).	Usabilidade	Importante
RNF07	O sistema deve utilizar ferramentas e bibliotecas gratuitas, compatíveis com o ambiente da faculdade.	Recursos	Essencial
RNF08	O código do sistema deve ser organizado em camadas separadas: apresentação, lógica e dados.	Padronização	Desejável

Tabela 02: Requisitos Não Funcionais do Sistema

## 4.3 Diagramas de Casos de Uso

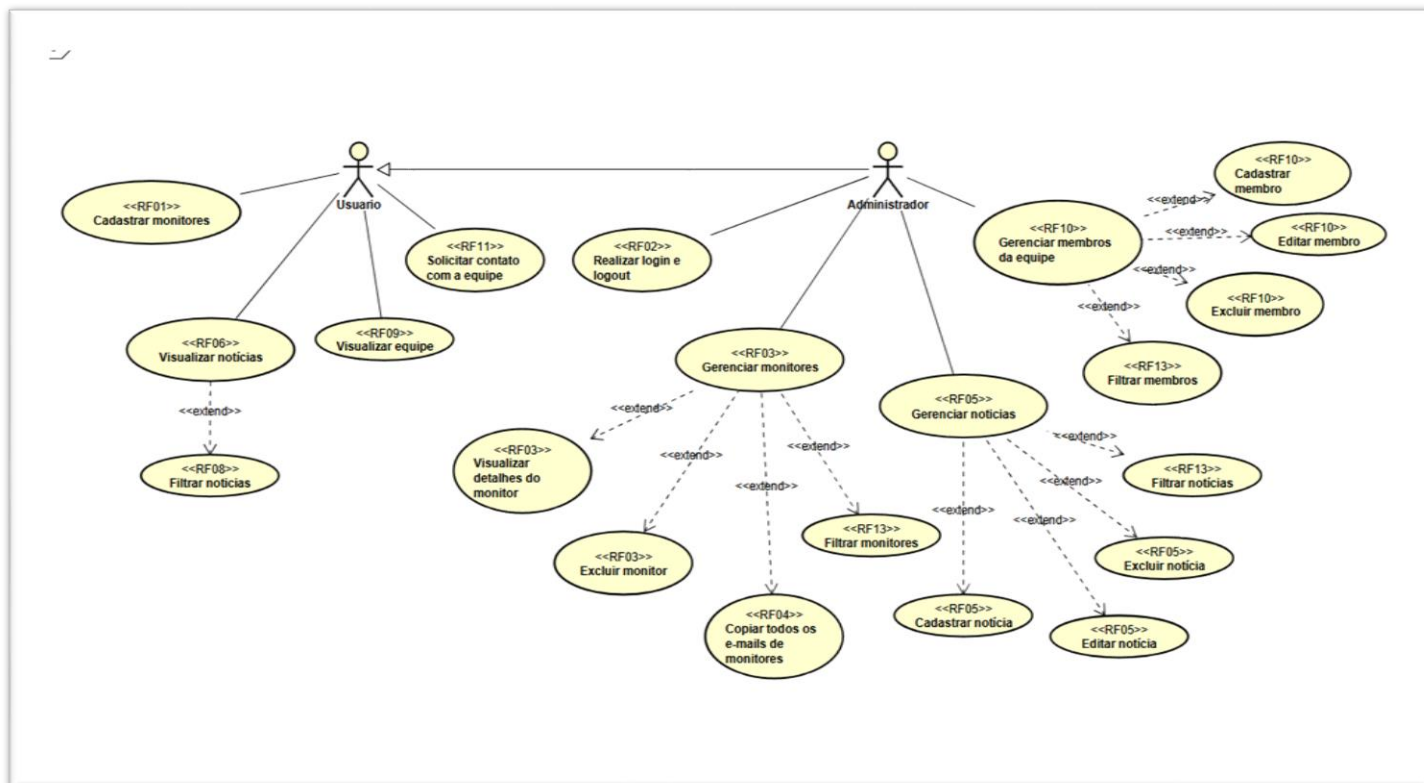


Figura 01: Diagrama de Caso de Uso Geral

## 4.4 Protótipos de Telas

Esta seção apresenta os protótipos de alta fidelidade das principais telas do sistema, elaborados com a ferramenta Figma para guiar o desenvolvimento das interfaces. As figuras a seguir ilustram as telas projetadas para os dois perfis de uso do sistema: Administrador, focado no gerenciamento de conteúdo, e Usuário, focado no consumo de informações de forma acessível.

As interfaces destinadas ao administrador, ilustradas nas **Figuras 02 e 03**, centralizam as funcionalidades de gestão do portal. O fluxo de navegação se inicia na tela de login e, após a autenticação, o administrador tem acesso a um menu que o direciona para as demais áreas restritas. As telas administrativas são:

- **Login Admin (RF02):** Página de autenticação exclusiva onde o administrador insere suas credenciais para acessar as funcionalidades restritas do sistema. Após o login, um botão de logout fica disponível para garantir a segurança da sessão.
- **Notícias Admin (RF05):** Área onde o administrador gerencia as publicações, podendo cadastrar, editar, excluir e pesquisar notícias. Cada publicação é composta por título, corpo e imagem.
- **Monitores Admin (RF03):** Tela que exibe a lista de voluntários cadastrados, com informações como nome, RA, CPF e e-mail. Nela, o administrador pode remover registros e copiar todos os e-mails para comunicação externa.

- **Equipe Admin (RF10):** Permite ao administrador gerenciar os membros do projeto, podendo adicionar, editar ou excluir integrantes, atualizando suas fotos, nomes e funções.

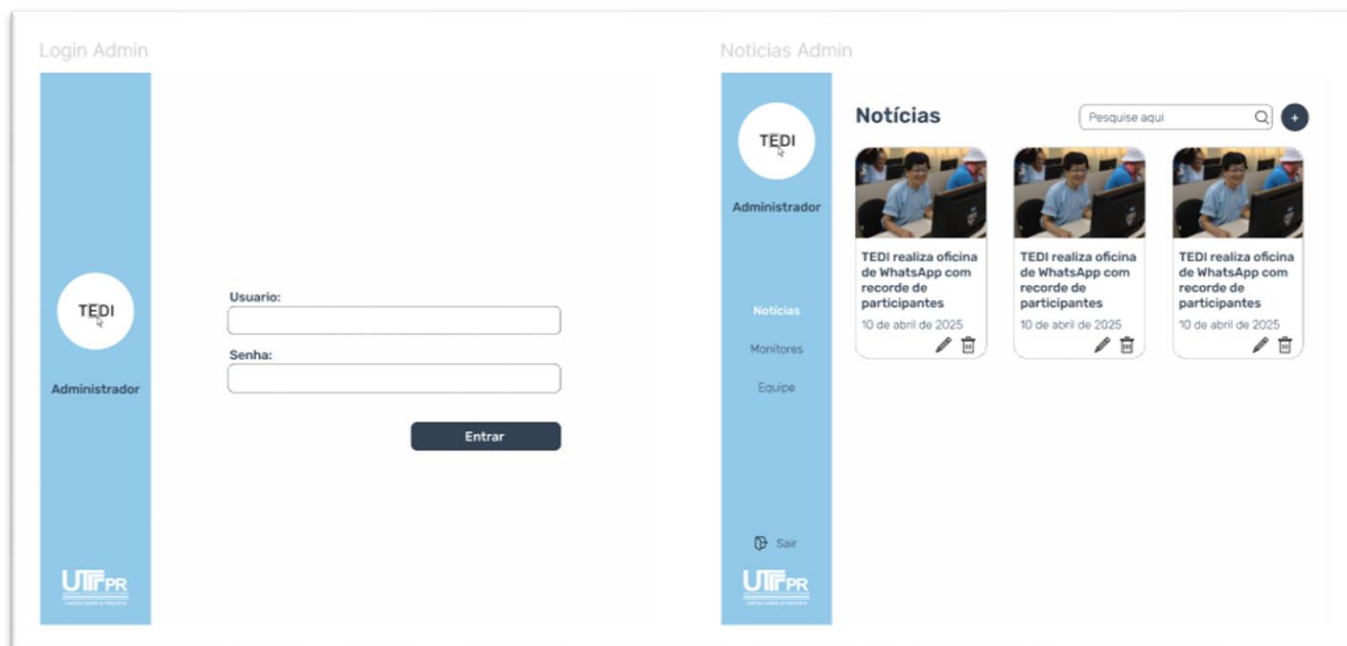


Figura 02: Protótipo de Tela de Administrador

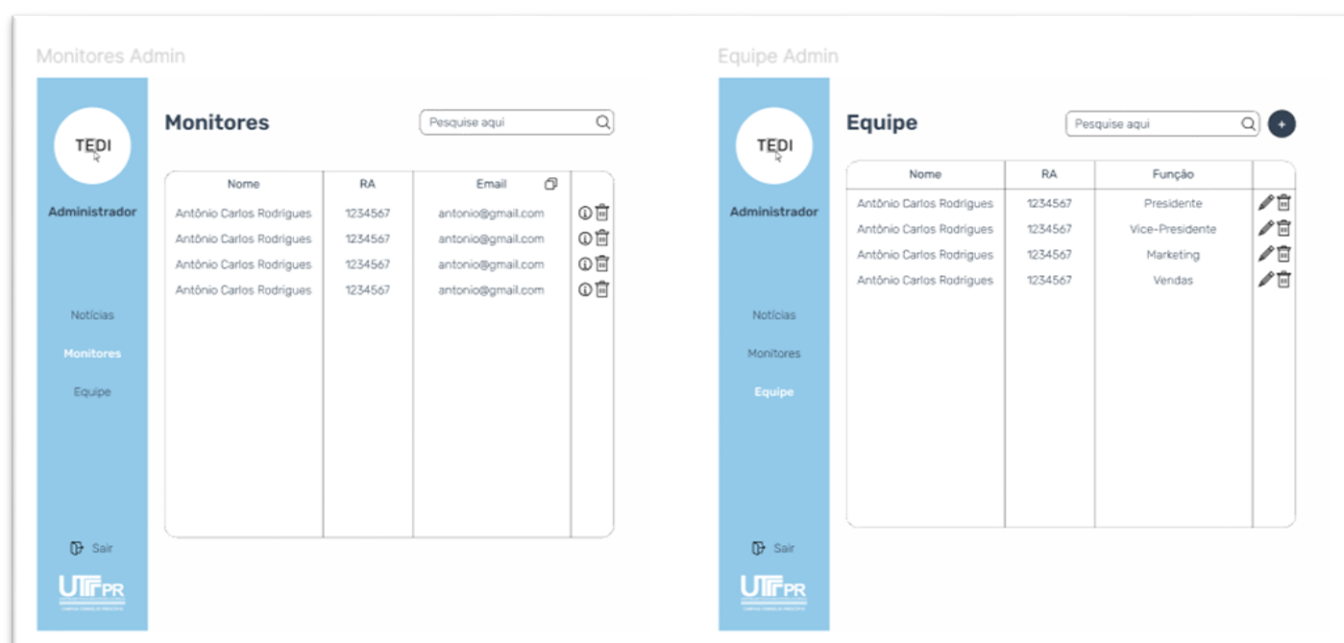


Figura 03: Protótipo de Telas de Administrador

Já as telas voltadas ao usuário final, apresentadas na **Figura 04**, priorizam a navegação intuitiva e o fácil acesso às informações. Todas as páginas de usuário contam com uma barra de navegação fixa no topo, garantindo uma experiência consistente. As principais telas são:

- **Tela Inicial (RFs 07, 11 e 12):** Página principal do site, onde os usuários têm acesso à barra de navegação fixa no topo (presente em todas as páginas). A tela inicial exibe um carrossel com miniaturas das últimas notícias, permite navegar para outras seções do site e conta com um formulário de contato para envio de mensagens à equipe do projeto.
- **Notícias (RFs 06 e 08):** Página dedicada à exibição de todas as notícias publicadas no projeto. Os usuários podem visualizar a lista completa de notícias e aplicar filtros por data, palavras-chave e categoria para refinar a busca.
- **Notícia Descrição (RFs 06 e 08):** Tela individual de uma notícia específica, exibindo todas as informações relacionadas a ela, como título, data, conteúdo completo e imagem destacada. Os filtros aplicados anteriormente podem continuar influenciando a navegação.
- **Equipe (RFs 09 e 08):** Página que apresenta os membros ativos do projeto com suas respectivas fotos, nomes e funções. Pode conter filtros para facilitar a visualização dos integrantes conforme a categoria ou função no projeto.
- **Monitores (RF01):** Tela destinada ao cadastro de monitores da UTFPR interessados em participar das oficinas. Um formulário coleta informações como nome, RA, CPF e e-mail, com validações para evitar duplicidade de RA e CPF no sistema.

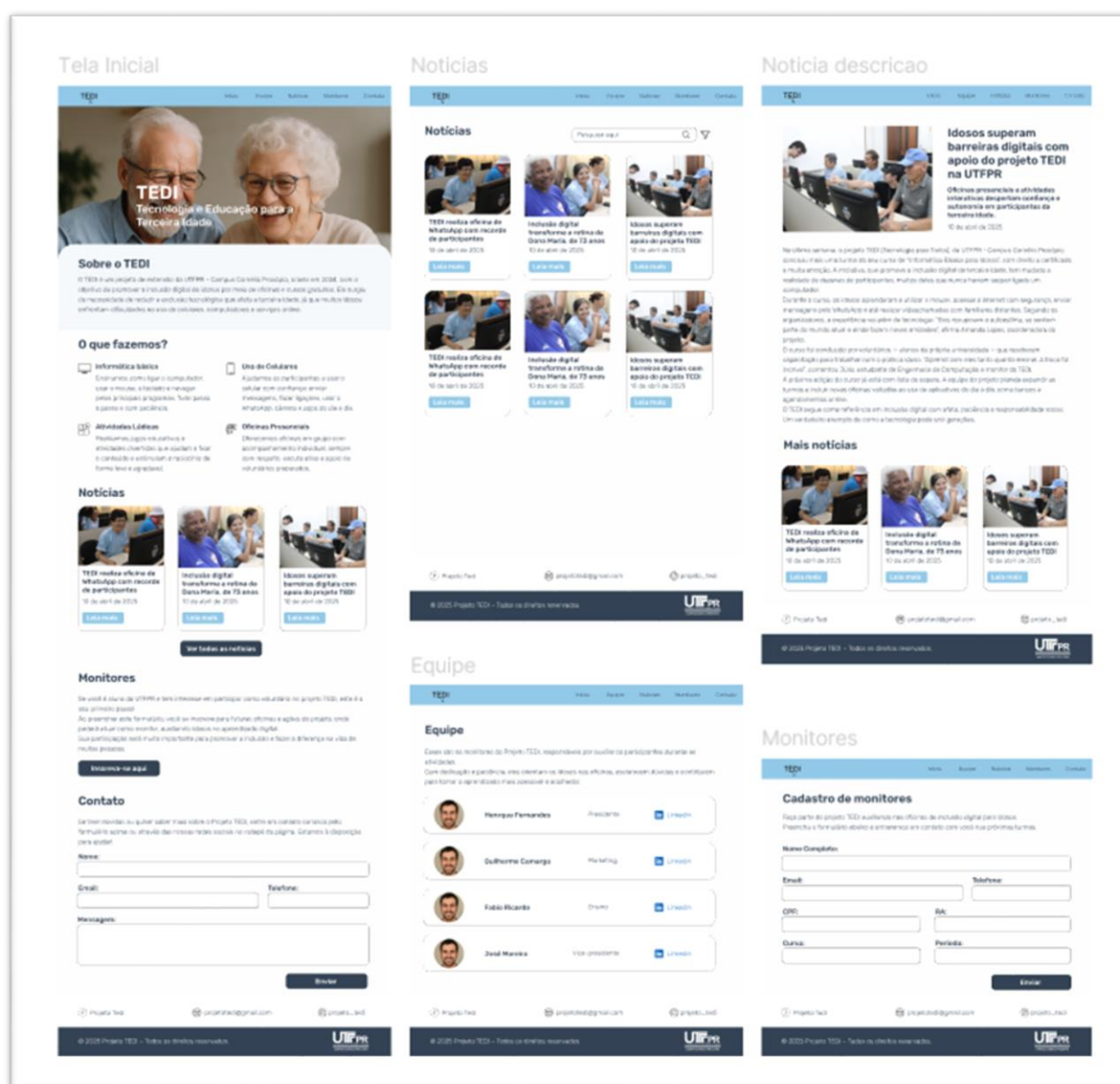


Figura 04: Protótipo de Telas de Usuário

## 5 Análise do Sistema

Esta seção apresenta os principais artefatos utilizados na modelagem e análise do sistema, com o objetivo de apoiar a implementação da aplicação web do projeto TEDI. Os modelos aqui descritos foram elaborados com base nos requisitos levantados e refletem a estrutura lógica e o comportamento esperado da aplicação. Inicialmente, é apresentado o **modelo conceitual do banco de dados**, que descreve as entidades, atributos e relacionamentos necessários para o funcionamento do sistema. Em seguida, é detalhado o **dicionário de dados**, contendo as especificações técnicas de cada campo e suas restrições. Também são incluídos o **diagrama de classes**, que ilustra a estrutura orientada a objetos do sistema, e o **diagrama de atividades**, que representa os fluxos de execução de funcionalidades relevantes, como o cadastro de monitores e a gestão de notícias. Esses artefatos foram elaborados com o intuito de fornecer uma base sólida para o desenvolvimento, facilitar a manutenção futura e garantir que todos os envolvidos tenham uma visão clara da estrutura e do comportamento do sistema. Esses artefatos foram elaborados com o intuito de fornecer uma base sólida para o desenvolvimento, facilitar a manutenção futura e garantir que todos os envolvidos tenham uma visão clara da estrutura e do comportamento do sistema.

Para evitar a perda permanente de informações, o projeto adota a abordagem de **exclusão lógica** (*soft delete*). Todas as tabelas principais contêm um campo booleano chamado `removed`. Ao invés de apagar um registro definitivamente, o valor deste campo é simplesmente alterado para `true`, o que preserva o histórico e a integridade dos dados. As consultas do sistema são, então, construídas para exibir aos usuários apenas os registros onde `removed` é `false`.

### 5.1 Modelo do Banco de Dados

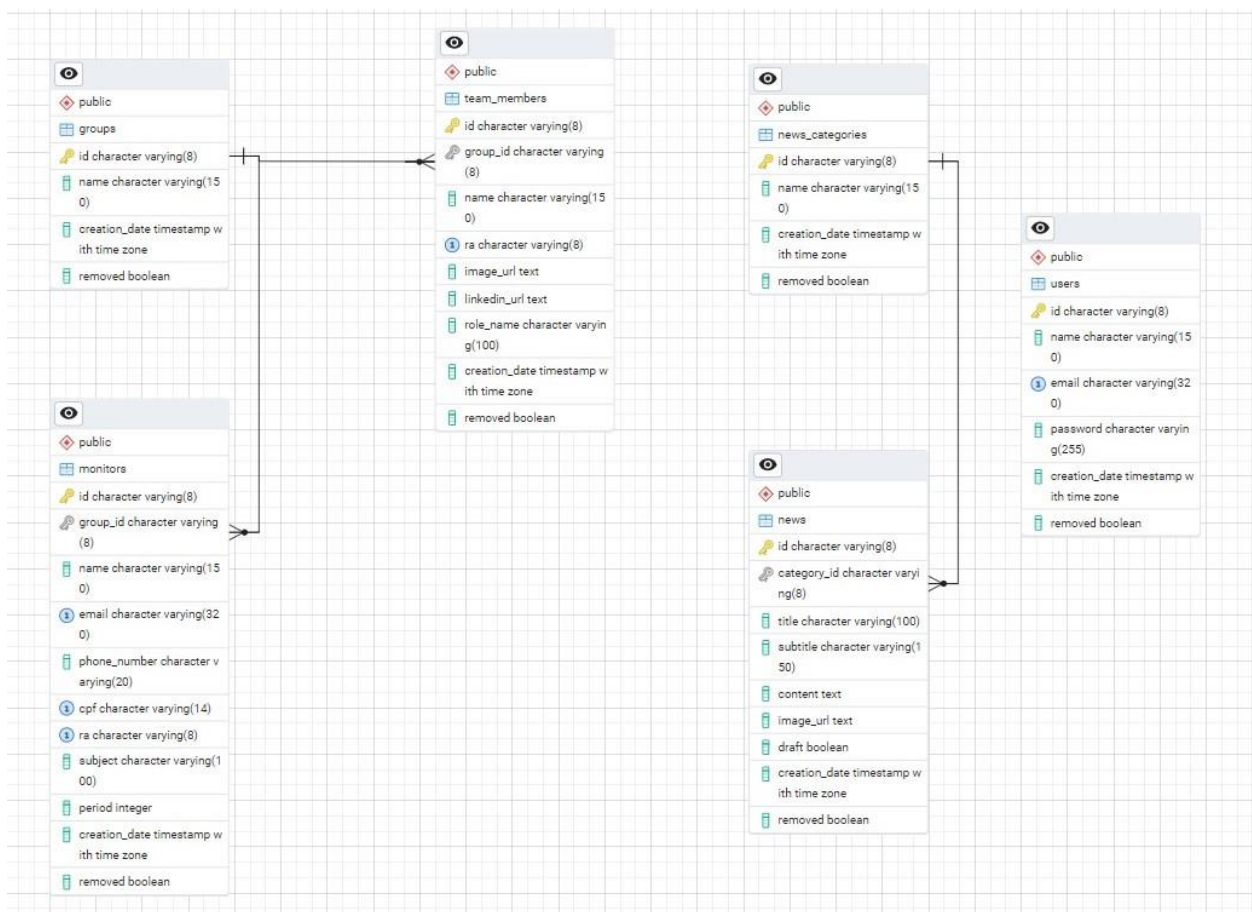


Figura 05: Modelo Conceitual (PostgreSQL)

## Dicionário de Dados

### Tabela: Users

**Descrição:** Armazena as informações dos usuários do sistema.

Nome	Descrição	Tipo de dado	Tamanho	Restrições
id	Identificador único	VARCHAR	8	PK, NOT NULL
name	Nome do usuário	VARCHAR	150	NOT NULL
email	Email do usuário	VARCHAR	320	UNIQUE, NOT NULL
password	Senha do usuário	VARCHAR	255	NOT NULL
creation_date	Data de criação	TIMESTAMP WITH TZ	-	NOT NULL, DEFAULT NOW()
removed	Indicador de remoção	BOOLEAN	-	NOT NULL, DEFAULT FALSE

### Tabela: News Categories

**Descrição:** Armazena as categorias de notícias.

Nome	Descrição	Tipo de dado	Tamanho	Restrições
id	Identificador único	VARCHAR	8	PK, NOT NULL
name	Nome da categoria	VARCHAR	150	NOT NULL
creation_date	Data de criação	TIMESTAMP WITH TZ	-	NOT NULL, DEFAULT NOW()
removed	Indicador de remoção	BOOLEAN	-	NOT NULL, DEFAULT FALSE

### Tabela: News

**Descrição:** Armazena notícias publicadas no sistema.

Nome	Descrição	Tipo de dado	Tamanho	Restrições
id	Identificador único	VARCHAR	8	PK, NOT NULL
category_id	Categoria da notícia	VARCHAR	8	FK, NOT NULL
title	Título da notícia	VARCHAR	100	NOT NULL
subtitle	Subtítulo da notícia	VARCHAR	150	-
content	Conteúdo da notícia	TEXT	-	NOT NULL
image_url	URL da imagem	TEXT	-	NOT NULL
draft	Indica se é rascunho	BOOLEAN	-	NOT NULL, DEFAULT TRUE
creation_date	Data de criação	TIMESTAMP WITH TZ	-	NOT NULL, DEFAULT NOW()
removed	Indicador de remoção	BOOLEAN	-	NOT NULL, DEFAULT FALSE

### Tabela: Monitors

**Descrição:** Armazena informações sobre monitores.



Nome	Descrição	Tipo de dado	Tamanho	Restrições
id	Identificador único	VARCHAR	8	PK, NOT NULL
group_id	Grupo associado	VARCHAR	8	FK
name	Nome do monitor	VARCHAR	150	NOT NULL
email	Email do monitor	VARCHAR	320	UNIQUE, NOT NULL
phone_number	Telefone do monitor	VARCHAR	20	NOT NULL
cpf	CPF do monitor	VARCHAR	14	UNIQUE, NOT NULL
ra	Registro acadêmico	VARCHAR	8	UNIQUE, NOT NULL
subject	Disciplina monitorada	VARCHAR	100	NOT NULL
period	Período do curso	INTEGER	-	NOT NULL, CHECK (1–10)
creation_date	Data de criação	TIMESTAMP WITH TZ	-	NOT NULL, DEFAULT NOW()
removed	Indicador de remoção	BOOLEAN	-	NOT NULL, DEFAULT FALSE

**Tabela: Team Members**

**Descrição:** Armazena informações dos membros da equipe.

Nome	Descrição	Tipo de dado	Tamanho	Restrições
id	Identificador único	VARCHAR	8	PK, NOT NULL
group_id	Grupo associado	VARCHAR	8	FK
name	Nome do membro	VARCHAR	150	NOT NULL
ra	Registro acadêmico	VARCHAR	8	UNIQUE, NOT NULL
image_url	URL da imagem	TEXT	-	NOT NULL
linkedin_url	URL do LinkedIn	TEXT	-	-
role_name	Nome do cargo	VARCHAR	100	NOT NULL
creation_date	Data de criação	TIMESTAMP WITH TZ	-	NOT NULL, DEFAULT NOW()
removed	Indicador de remoção	BOOLEAN	-	NOT NULL, DEFAULT FALSE

## 5.2 Diagrama de Classes

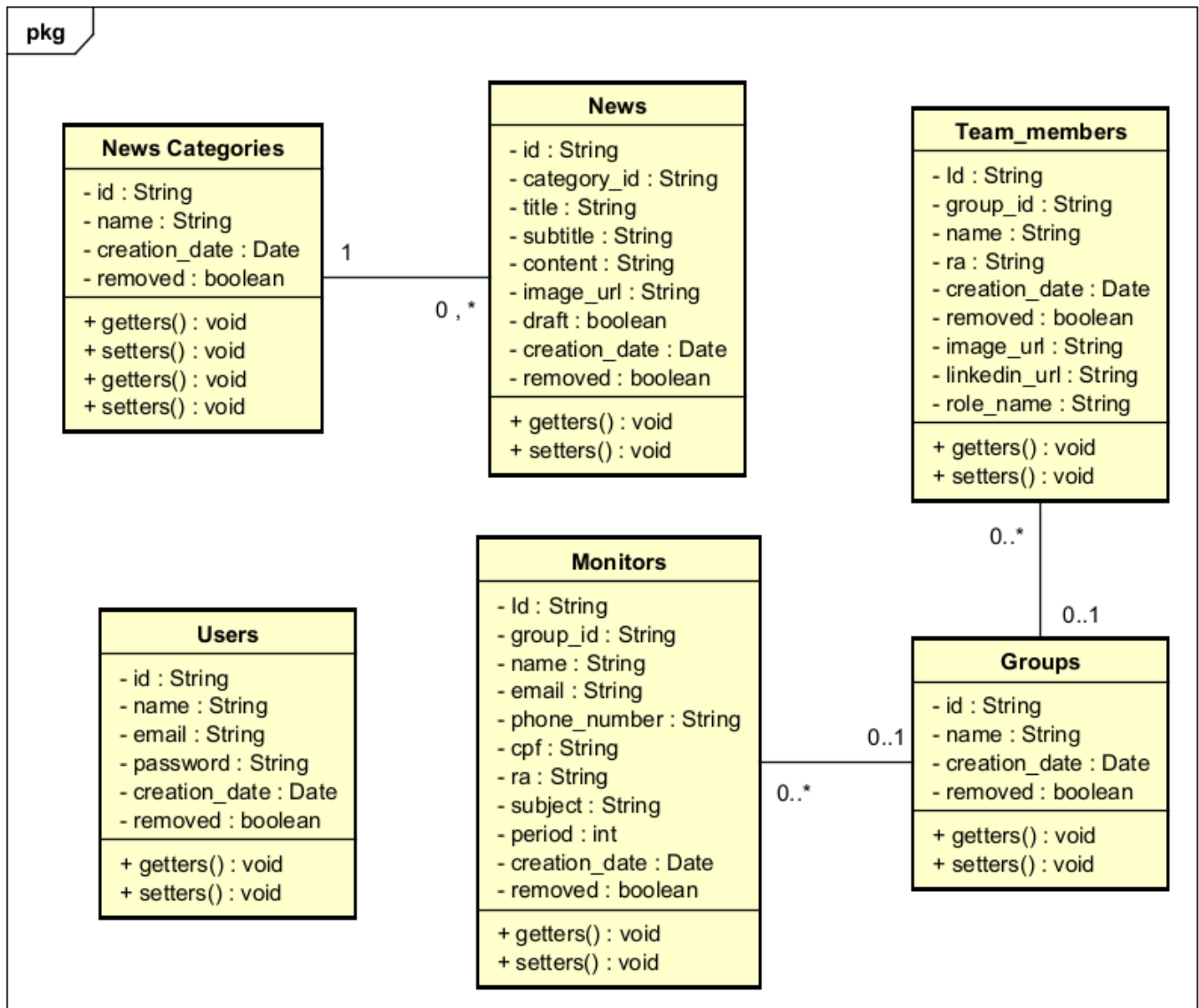


Figura 06: Diagrama de Classes

### 5.3 Diagrama de Atividades

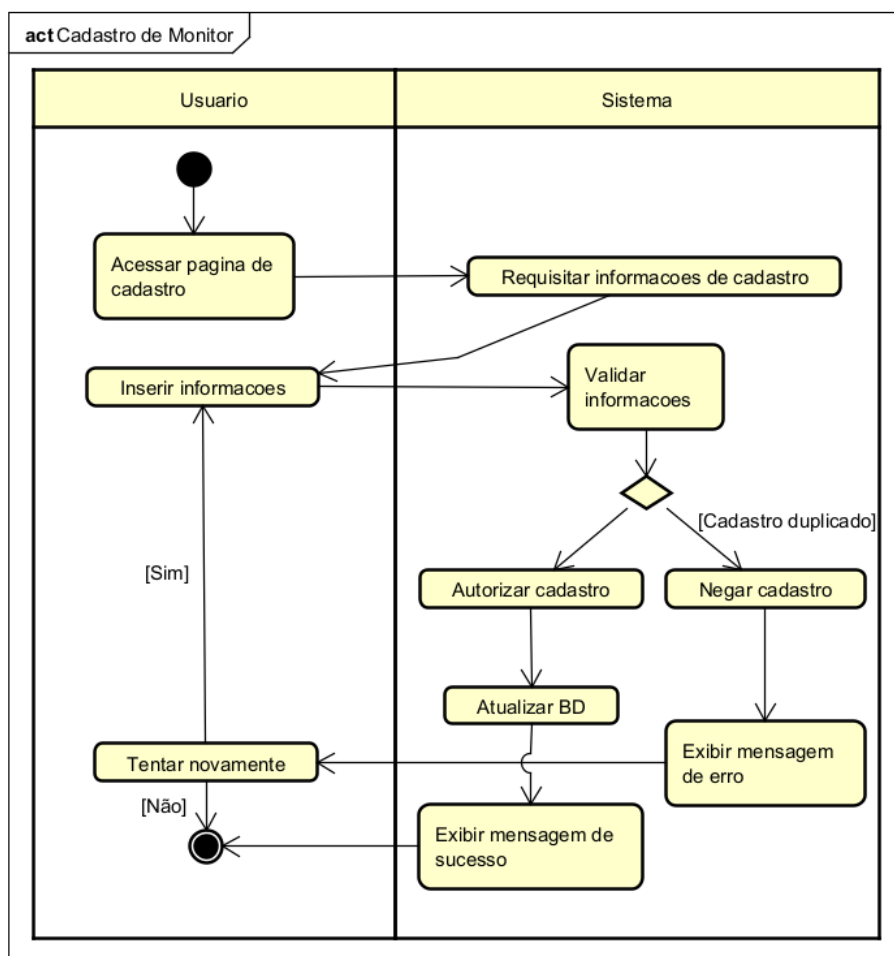


Figura 07: Diagrama de Atividades do Cadastro de Monitor

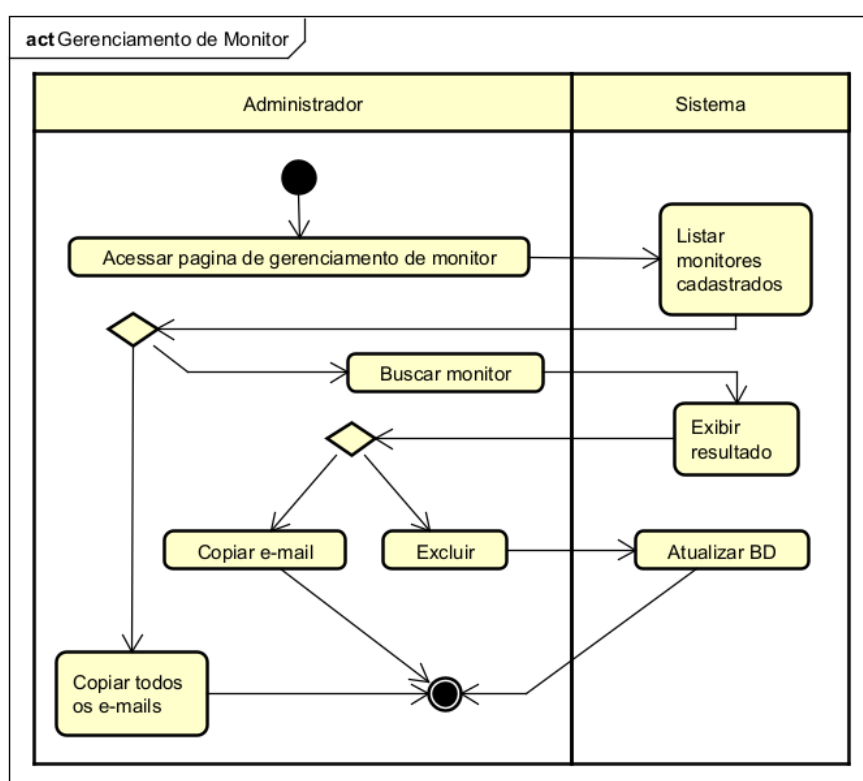


Figura 08: Diagrama de Atividades do Gerenciamento de Monitor

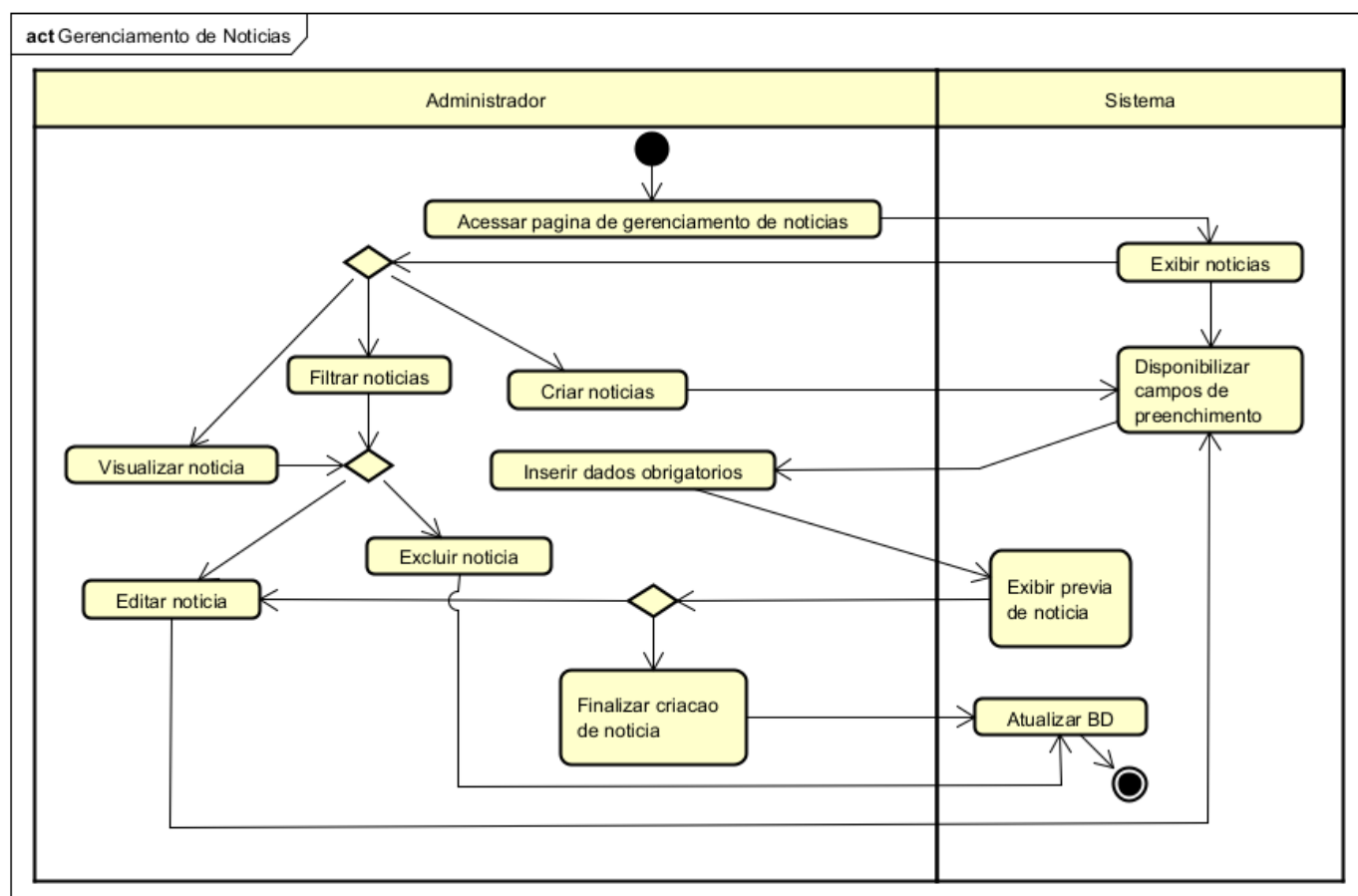


Figura 09: Diagrama de Atividades do Gerenciamento de Notícias

## 6 Implementação

### 6.1 Descrição do código

O sistema foi desenvolvido seguindo uma arquitetura de software moderna, com uma separação clara entre as responsabilidades do **Frontend** (interface do usuário) e do **Backend** (servidor e lógica de negócio), comunicando-se através de uma API RESTful.

#### Estrutura do Frontend (Angular)

A aplicação frontend foi construída com Angular e organizada de forma modular para facilitar a manutenção e a escalabilidade. A estrutura de diretórios principal, localizada em `src/app/`, foi dividida da seguinte forma:

- **admin:** Contém os componentes e a lógica das páginas restritas ao administrador, como os painéis de gerenciamento de notícias, monitores e equipe.
- **user:** Agrupa os componentes das páginas públicas do portal, como a página inicial, o feed de notícias e a página da equipe.
- **guards:** Armazena as guardas de rota, responsáveis por proteger o acesso às páginas do módulo `admin`, permitindo a entrada apenas de usuários autenticados.
- **shared:** Centraliza todos os elementos reutilizáveis do projeto, como componentes visuais (cabeçalho, rodapé), serviços de comunicação com a API e modelos de dados (interfaces TypeScript).

#### Estrutura do Backend (Express)

O backend, desenvolvido com Express, foi estruturado em camadas para garantir o baixo acoplamento e a alta coesão do código. A organização segue um fluxo de requisição claro:

- **routes:** Define os endpoints da API. É a porta de entrada das requisições.
- **middleware:** Contém funções que interceptam as requisições antes de chegarem aos controllers, utilizadas para validações e autenticação.
- **controller:** Responsável por receber a requisição, validar os dados de entrada e chamar a camada de serviço correspondente.
- **service:** Orquestra a lógica de negócio da aplicação.
- **repositories:** Camada de abstração de dados, responsável por toda a comunicação com o banco de dados.
- **models:** Define as entidades do banco de dados para a manipulação dos dados na aplicação.
- **config:** Armazena as configurações do ambiente, como as credenciais de acesso ao banco de dados PostgreSQL.

### Diagramas da Arquitetura

Os diagramas a seguir ilustram visualmente a arquitetura de software e a implantação do sistema.

#### Diagrama de Componentes

O diagrama abaixo representa os principais componentes de software do sistema e as interfaces de comunicação entre eles.

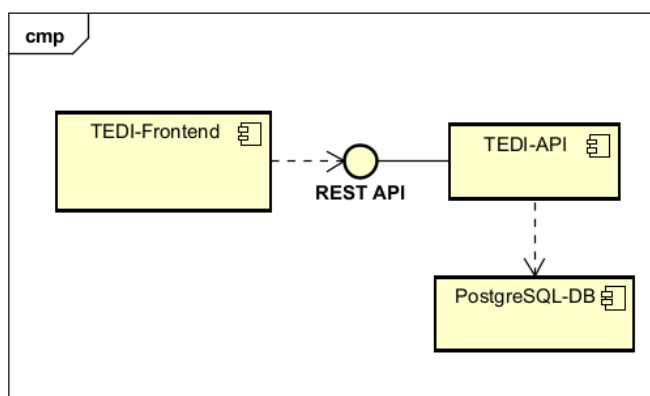


Figura 10: Diagrama de Componentes

### Diagrama de Pacotes

Este diagrama detalha a organização lógica dos pacotes de código do Frontend e do Backend, assim como as dependências entre eles.

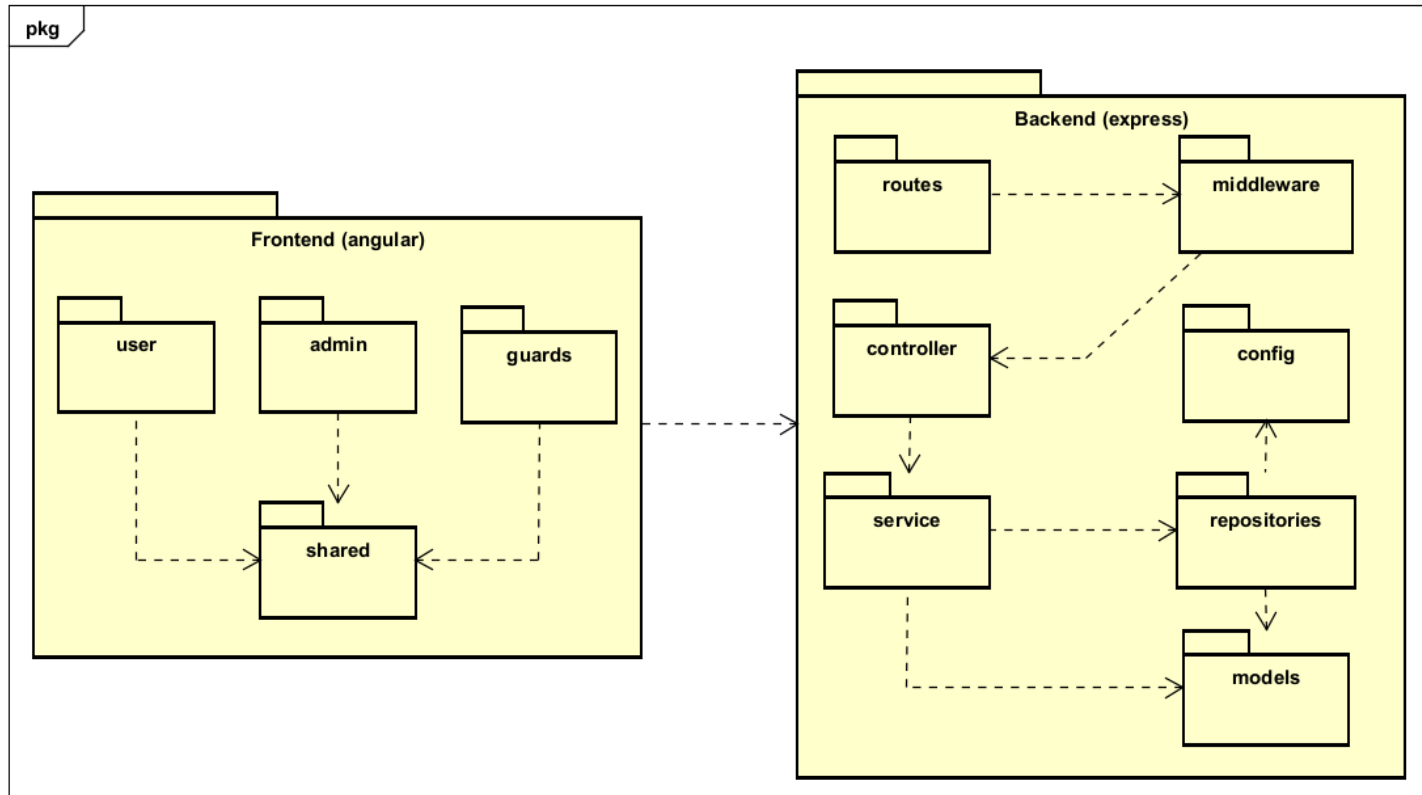


Figura 11: Diagrama de Pacotes

## Diagrama de Implantação

O diagrama a seguir mostra a arquitetura física, indicando em quais ambientes (nós) cada componente do sistema é executado.

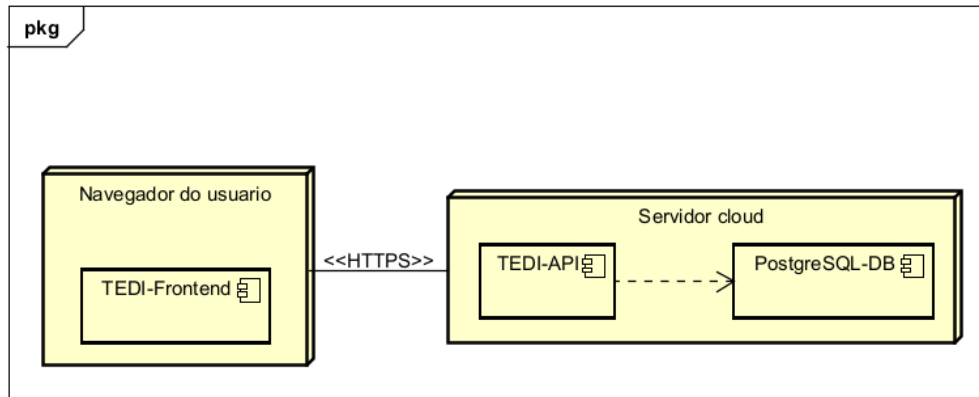


Figura 12: Diagrama de Implantação

## Convenções de Código

Para garantir a legibilidade e a manutenibilidade do código, o projeto adotou as seguintes convenções:

- **Nomenclatura:** Classes e interfaces seguem o padrão `PascalCase`, enquanto variáveis, funções e nomes de arquivos utilizam `camelCase` ou `kebab-case`, seguindo as diretrizes da comunidade de cada tecnologia (Angular e JavaScript).
- **Comentários:** Foram utilizados comentários no código para explicar blocos de lógica complexa ou decisões de implementação que não são imediatamente óbvias.

## 6.2 Implantação

Para executar o software em um ambiente de desenvolvimento local, é necessário seguir os passos abaixo, garantindo que os pré-requisitos de ambiente estejam devidamente instalados e configurados.

### Requisitos Necessários

- **Node.js:** Versão 16 ou superior.
- **npm:** Gerenciador de pacotes do Node.js (geralmente instalado com o Node.js).
- **Angular CLI:** Interface de Linha de Comando do Angular, versão 19 ou superior.
- **PostgreSQL:** Uma instância do banco de dados ativa e acessível.

### Passos para Execução

A execução do sistema é dividida em duas etapas principais: a inicialização do Backend (servidor) e, em seguida, a do Frontend (interface do usuário).

## Execução do Backend

**Navegação e Instalação:** Navegue pelo terminal até a pasta raiz do projeto Backend e execute o comando abaixo para instalar todas as dependências necessárias: `npm install`

**Configuração de Ambiente:** Crie um arquivo nomeado `.env` na raiz da pasta Backend. Este arquivo deve conter as seguintes variáveis de ambiente, preenchidas com os dados correspondentes ao seu ambiente local:

- `PORT`: Porta onde a API será executada (ex: 3333).
- `SECRET`: Chave secreta para a geração de tokens JWT.
- `TOKEN_EXPIRATION`: Tempo de expiração do token (ex: 86400000 para 24 horas).
- `FRONTEND_URL`: URL onde a aplicação frontend será executada (ex: `http://localhost:4200`).
- `EMAIL_USER`: E-mail de uma conta Gmail para envio de notificações.
- `EMAIL_PASS`: Senha de aplicativo da conta de e-mail especificada.
- `DB_NAME`: Nome do banco de dados no PostgreSQL.
- `DB_USER`: Usuário de acesso ao banco de dados.
- `DB_PASS`: Senha do usuário do banco de dados.
- `DB_HOST`: Host do servidor de banco de dados (ex: `localhost`).
- `DB_PORT`: Porta do servidor de banco de dados (ex: 5432).

**Inicialização do Banco de Dados:** É necessário inicializar a estrutura e os dados do banco. Os scripts SQL para esta tarefa estão localizados no diretório `/Documentacao - TEDI/Database/`.

Primeiro, execute o script `DB_structure.sql` para criar todas as tabelas e relacionamentos.

Em seguida, execute o script `Initial Inserts.sql` para popular o banco com dados iniciais essenciais para o funcionamento do sistema.

**Inicialização do Servidor:** Após a configuração, execute o comando abaixo para iniciar o servidor do backend: `npm start`

## Execução do Frontend

**Navegação e Instalação:** Em um novo terminal, navegue até a pasta raiz do projeto Frontend e execute o seguinte comando para instalar suas dependências: `npm install`

**Configuração de Ambiente:** Acesse o diretório `src/environments/` e localize o arquivo de configuração de ambiente (ex: `environment.ts`). Nele, certifique-se de que a variável que aponta para a URL da API corresponde ao endereço e porta em que o backend está sendo executado (ex: `http://localhost:3333`).

**Inicialização da Aplicação:** Execute o comando abaixo para iniciar a aplicação frontend: `npm start`

Após a conclusão desses passos, a aplicação web do Portal TEDI estará acessível em um navegador, geralmente no endereço `http://localhost:4200`.



## 6.3 Telas principais

A seguir, são apresentadas as telas finais do Portal TEDI, demonstrando o resultado da implementação e o funcionamento das principais funcionalidades do sistema. A descrição foca na dinâmica de navegação e nas ações que podem ser executadas em cada tela.

### Telas Públicas

As telas públicas foram projetadas com foco na navegação intuitiva. Uma barra de navegação superior fixa está sempre visível, permitindo que o usuário acesse qualquer seção principal do site a qualquer momento.

#### Página Inicial

Funciona como o hub central do portal. O corpo da página exibe uma seção com miniaturas das últimas notícias publicadas; ao clicar em uma, o usuário é levado para a leitura completa do conteúdo. A página também oferece acesso direto à área de voluntariado através de um botão de cadastro de monitores e, ao final, apresenta as informações de contato da equipe junto a um formulário para envio de mensagens.

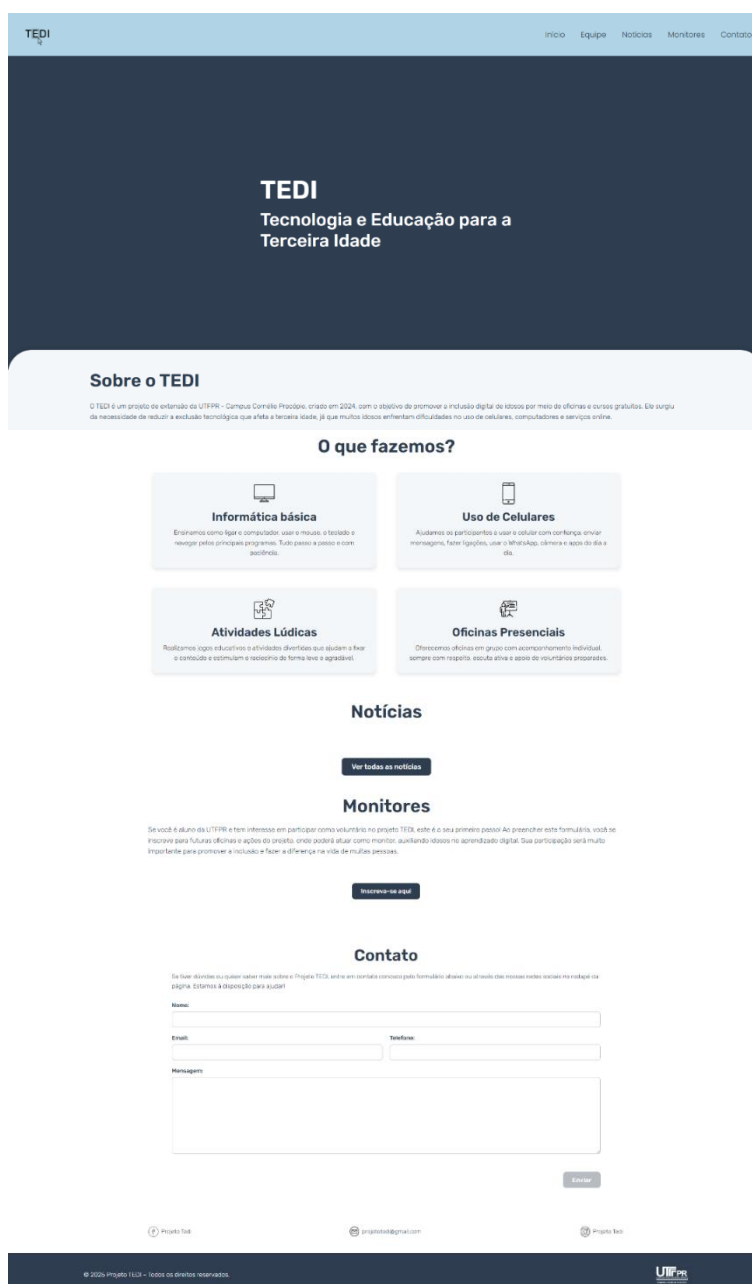


Figura 13: Página Inicial

## Página da Equipe

Apresenta os membros que compõem a equipe do projeto TEDI, exibindo nome, foto e a função de cada integrante (RF09).

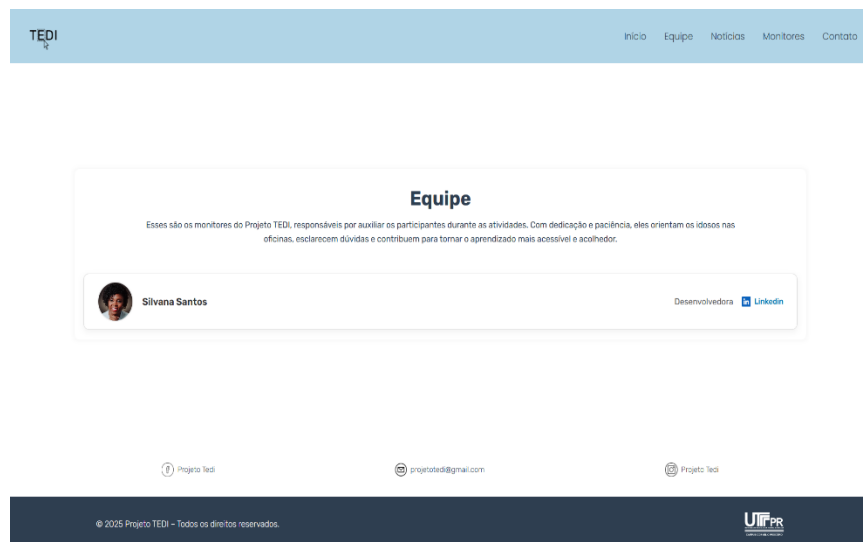


Figura 14: Página de Equipe

## Página de Cadastro de Monitores

Página com um formulário acessível para que alunos da UTFPR possam se voluntariar para o projeto (RF01), com validações para impedir o cadastro de RA e CPF duplicados.

Figura 15: Página de Cadastro de Monitores

## Telas Administrativas

Acesso restrito à equipe do projeto para gerenciamento do conteúdo do portal. Após a autenticação, todas as páginas administrativas mantêm uma barra de navegação lateral que permite o acesso rápido e consistente às diferentes seções de gerenciamento (RF12).

### Tela de Login

Portal de entrada para a área administrativa, onde o administrador insere suas credenciais para obter acesso às funcionalidades restritas do sistema.

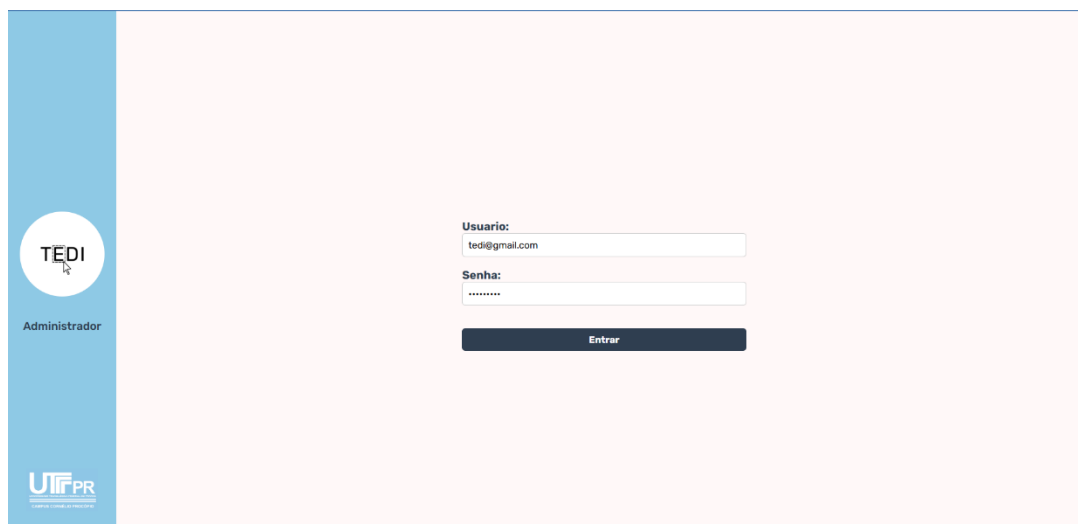


Figura 16: Página de Login de Administrador

### Página de Gerenciamento de Notícias

É a primeira tela acessada pelo administrador após o login. Ela exibe uma lista de todas as notícias criadas, tanto as que já estão públicas quanto as salvas como rascunho. Para cada notícia, o administrador tem a opção de editar seu conteúdo, excluí-la permanentemente ou alterar sua visibilidade (publicar uma notícia em rascunho ou despublicar uma já existente).

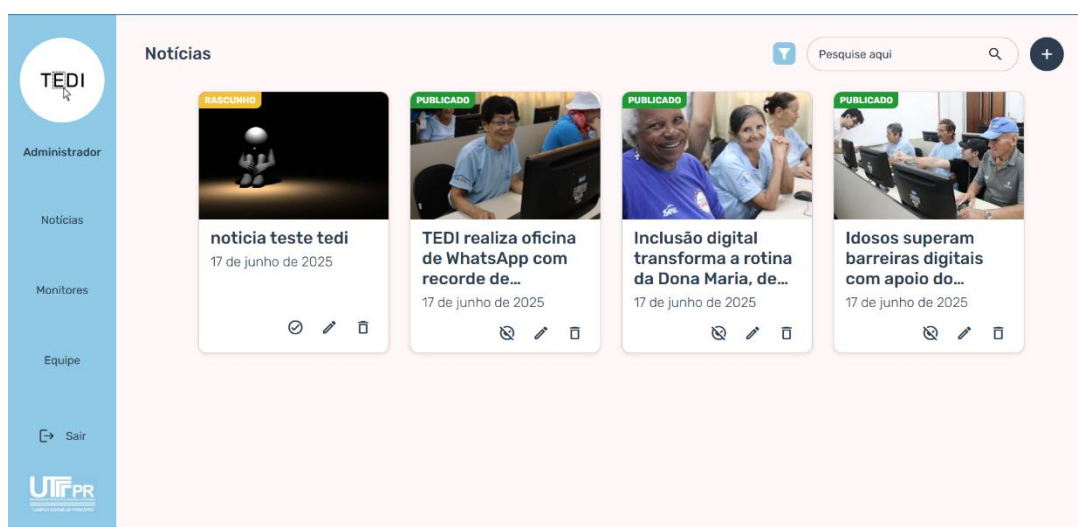
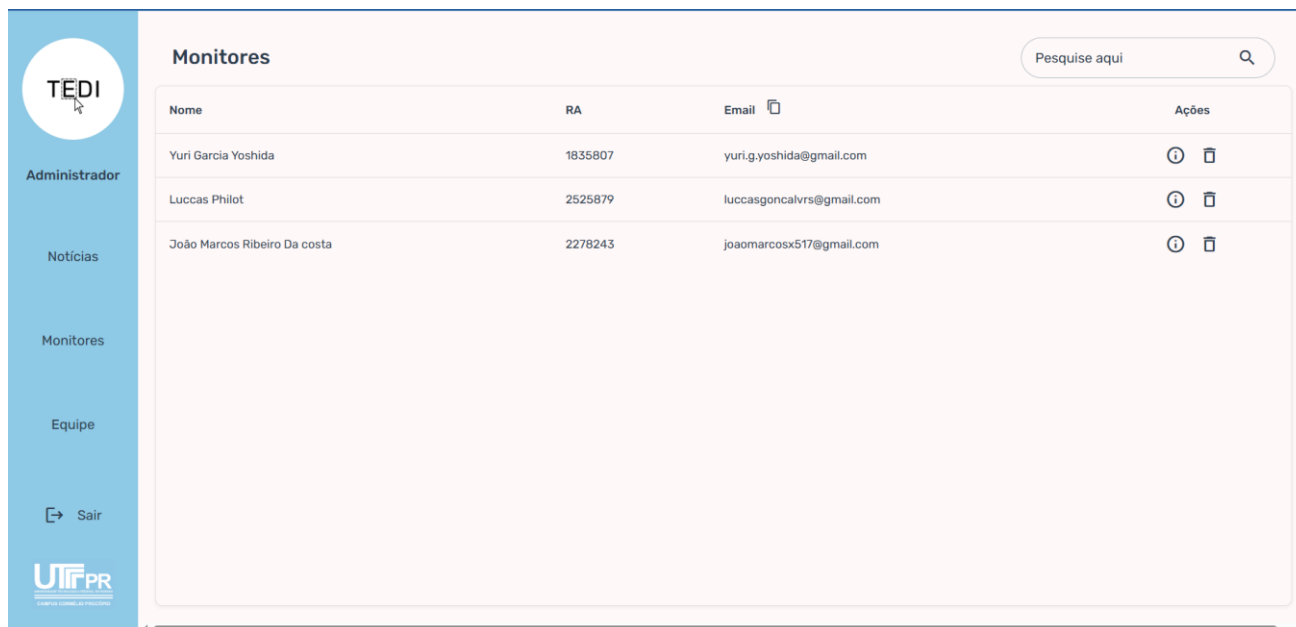


Figura 17: Página de Gerenciamento de Notícias

## Página de Gerenciamento de Monitores

Oferece ao administrador as ferramentas para gerenciar os voluntários cadastrados. Nesta tela, é possível visualizar a lista de monitores, excluir registros (RF03) , filtrar as informações para facilitar a busca (RF13) e copiar todos os e-mails dos voluntários com um único clique para comunicação externa (RF04).









Nome	RA	Email	Ações
Yuri Garcia Yoshida	1835807	yuri.g.yoshida@gmail.com	 
Lucas Philot	2525879	lucasgoncalvrs@gmail.com	 
João Marcos Ribeiro Da costa	2278243	joaomarcosx517@gmail.com	 

Figura 18: Página de Gerenciamento de Monitores

## 7 Considerações Finais

O desenvolvimento do Portal TEDI alcançou seus objetivos gerais e específicos, resultando em uma plataforma web funcional que serve como um ponto central de comunicação e organização para o projeto de extensão TEDI. O portal foi apresentado à equipe do projeto de extensão TEDI, que aprovou o resultado final e expressou grande satisfação com a ferramenta desenvolvida, validando o sucesso da iniciativa. O sistema entrega os requisitos essenciais propostos, como um portal para a divulgação de notícias (RF06) , um formulário para o cadastro de novos monitores voluntários (RF01) e uma área administrativa para o gerenciamento de conteúdo (RF03, RF05, RF10) , preenchendo a lacuna de comunicação que motivou o projeto.

As limitações do projeto são aquelas definidas no escopo inicial, como a ausência de funcionalidades de interação direta entre usuários ou integração com redes sociais. O foco se manteve na entrega de uma ferramenta de divulgação e gestão.

Como trabalhos futuros, o sistema pode ser expandido para agregar ainda mais valor ao projeto TEDI. Sugere-se a implementação de um sistema de cadastro e gerenciamento de eventos (oficinas e cursos), permitindo a inscrição online dos idosos. Adicionalmente, o requisito desejável de criação de grupos de monitores (RF14) poderia ser implementado, facilitando a comunicação direcionada da equipe. A criação de uma galeria de fotos e uma área de depoimentos também poderiam enriquecer a experiência do usuário e aumentar o engajamento com o projeto.

## 8 Bibliografia

A seguir, são apresentadas as obras e documentações técnicas utilizadas como referência para a elaboração deste documento e para a implementação do projeto Portal TEDI.

- **ANGULAR. Overview.**  
Disponível em: <https://angular.dev/overview>.
- **EXPRESS. API reference.**  
Disponível em: <https://expressjs.com/en/api.html>.
- **POSTGRESQL. Documentation.**  
Disponível em: <https://www.postgresql.org/docs/>.
- **SCHWABER, Ken; SUTHERLAND, Jeff. The Scrum Guide.**  
Disponível em: <https://scrumguides.org>.
- **ASTAH. Astah UML - UML Modeling Tool.**  
Disponível em: <https://astah.net/products/astah-uml/>.