



Curso Superior de Desenvolvimento de Software Multiplataforma

Ariele Peres, 3011392310000

Fernando Ribeiro, 3011392310000

Guilherme Pazetti, 3011392413008

Guilherme Piva, 3011392413043

Mayara Barros, 3011392413017

Vanessa Capuano, 3011392413031

Projeto Interdisciplinar

Gestão ágil de projetos

Desenvolvimento Web III

Interação Humano Computador

Carità

Orientadores: Prof^a Maria Janaina

Prof^o Thiago Vanderlei

Votorantim

Junho, 2025

Resumo

O projeto tem como objetivo criar um site que contribua para os Objetivos de Desenvolvimento Sustentável (ODS), como fome zero (ODS 2), redução das desigualdades (ODS 10) e consumo responsável (ODS 12). A plataforma será focada na cidade de Votorantim-SP e terá como principal função divulgar pontos de arrecadação de alimentos, tornando as doações mais acessíveis e organizadas.

O site contará com recursos como uma página informativa para doadores, dados sobre a fome no Brasil, um mapa mostrando os locais de coleta e orientações claras para doações seguras e de qualidade. A ideia é facilitar o processo de doação e aumentar a transparência, ajudando instituições a arrecadarem alimentos de forma mais eficaz.

Com isso, o projeto busca reduzir a fome e melhorar o bem-estar das comunidades carentes de Votorantim, incentivando a participação de mais pessoas e fortalecendo o impacto das doações.

Índice

1. Descrição do Projeto	6
1.1. Tema Central do Projeto	6
1.2. Objetivo Geral do Projeto	6
1.3. Motivação	7
2. Interação Humano Computador	8
2.1. Cenários	8
2.2. Personas	9
2.3. Acessibilidade.....	10
2.4. Interface do Site - explicação	11
2.5. CardSorting.....	12
2.6. Definição de símbolos	12
Redes Sociais	12
Acessibilidade	13
2.7. Design system.....	14
2.7.1. Identidade Visual.....	14
Tipografia.....	14
2.7.2. Componentes UI.....	15
2.7.3. Cards	15
2.7.4. Layout.....	15
2.7.5. Imagens e Ícones	16
2.7.6. Tom de Voz.....	16
2.7.7. Navegação	16
2.7.8. Responsividade	16
3. Gestão ágil	17
3.1. User Stories	18
3.2. Sprints	19
3.3. Backlogs	22
3.4. Resultados alcançados	23

3.5 Requisitos do projeto	24
3.5.1. Levantamento de Requisitos	24
3.5.2. Requisitos funcionais	24
3.5.3. Requisitos não funcionais	27
3.6. Diagrama de caso de uso	29
3.7. Projeto de Software	30
3.8. Tecnologias utilizadas	31
3.9. Integração e Papéis da Equipe	32
4. Técnicas de programação	32
4.1. Modelagem de banco de dados.....	32
4.2. DER.....	33
4.3. Modelo Lógico	34
4.4. Diagrama de classe.....	35
4.5. Atributos	37

Lista de Figuras

Figura 1 - Logomarca do projeto Caritá	7
Figura 2 - Personas 1, 2 e 3.	9
Figura 3 - Personas 4, 5 e 6	10
Figura 4 - Cores, Logo e Fonte Caritá.....	11
Figura 5 - redes	12
Figura 6 - Acessibilidade	13
Figura 7 - Acessibilidade	13
Figura 8 - Acessibilidade	14
Figura 9 - Projeto de Software.....	30

Lista de Quadros

Quadro 1 – CardSorting.	12
Quadro 2 – Sprints.	19
Quadro 3 – Backlog.	22
Quadro 4 – Diagrama de caso de uso	29
Quadro 5 – Diagrama de caso de uso	33
Quadro 6 – Modelo Lógico.	35
Quadro 7 – Diagrama de Classe.	36

1. Descrição do Projeto

1.1. Tema Central do Projeto

O Projeto Caritá é uma iniciativa social voltada à doação de alimentos na cidade de Votorantim, com o objetivo de combater a fome e promover a solidariedade na comunidade local.

1.2. Objetivo Geral do Projeto

O software é uma plataforma online que facilita o processo de doação de alimentos ao redirecionar os doadores para pontos de arrecadação confiáveis em Votorantim. Atua como um guia virtual que informa onde e como os doadores podem contribuir de forma eficiente e segura para combater a fome na comunidade.

Os principais objetivos são: Facilitar o acesso dos doadores a informações precisas sobre pontos de arrecadação de alimentos e assegurar que os alimentos doados atendam aos padrões de qualidade e segurança estabelecidos.

O software visa resolver a ausência de uma plataforma centralizada que informe de maneira clara e eficaz os locais onde os doadores podem realizar suas contribuições de alimentos em Votorantim.

O público-alvo são pessoas interessadas em contribuir com doações de alimentos, entidades e instituições que coordenam a distribuição de alimentos para famílias necessitadas em Votorantim e tenham interesse em se cadastrar e aparecer no site. Além disso, a ideia é que o software seja comercializado e, posteriormente, utilizado pela Prefeitura de Votorantim como uma ferramenta para intermediar e controlar as doações e arrecadações.

Para que uma instituição tenha seu endereço divulgado no site, é necessário que um representante oficial realize o cadastro na página de login. Durante o cadastro, o representante deverá incluir as informações da instituição e enviar os documentos que comprovem sua existência. Após a validação e aprovação do cadastro, caso a instituição possua um ponto de arrecadação, ela poderá cadastrar esse local no site. Caso a instituição não possua um ponto de arrecadação, ela será listada na página de doações via Pix, permitindo que os usuários façam doações diretamente.

Para se tornar um parceiro, é necessário que um representante oficial realize o cadastro no site. O parceiro pode cadastrar um ou mais pontos de arrecadação, desde que comprove que cada ponto está vinculado a uma instituição. Para isso, o representante deverá enviar documentos que comprovem essa parceria. Após a aprovação, os pontos de arrecadação serão exibidos no site. Além disso, também é possível que parceiros sem pontos de arrecadação participem divulgando o site em seus estabelecimentos. Nesse caso, eles serão listados na seção "Parceiros" do site. Podem ser parceiros supermercados, lojas, escolas, igrejas, entre outros. A validação da documentação enviada é realizada pelo responsável que adquirir o software.

As ODS abrangidas são ODS 2 Fome Zero, ODS 10 Redução das Desigualdades e ODS 12 Consumo e Produção Responsáveis.

1.3. Motivação

A ideia de desenvolver este site surgiu da necessidade de facilitar as doações de alimentos em Votorantim. Atualmente, não há um local centralizado que reúna informações sobre todos os pontos de arrecadação de alimentos na cidade, o que dificulta o processo para quem deseja contribuir. Muitas pessoas não estão cientes sobre a realidade da fome na região e das formas que podem ajudar. O site tem como objetivo aumentar a conscientização sobre esse problema e incentivar a participação ativa da comunidade, tornando as doações mais acessíveis e organizadas.

Logomarca



Figura 1 - Logomarca do projeto Caritá

2. Interação Humano Computador

Nesta etapa do projeto, foram definidos os principais cenários de uso do site Caritá com base em pesquisas sobre o público-alvo. A proposta visa garantir uma experiência acessível e intuitiva para diferentes perfis de usuários, desde doadores até representantes de instituições. Também foram aplicadas boas práticas de acessibilidade, como contraste adequado de cores, textos alternativos para imagens e navegação simplificada por teclado, visando tornar o site inclusivo para todos.

2.1. Cenários

Cenários propostos para o uso do site Caritá (ambiente idealizado conforme pesquisas de público-alvo).

- **Cenário 1 – Fazer Doações.**

Maria, 27 anos, quer começar a fazer doações, mas não sabe por onde começar. Ela acessa o site e se depara com um meio seguro de fazer suas doações em locais próximos de si.

- **Cenário 2 – Cadastrar comércio como ponto de arrecadação.**

Caio, 54 anos, quer que as pessoas saibam que seu comércio também é um ponto de arrecadação. No site ele finalmente pode alavancar as doações e promover seu comércio.

- **Cenário 3 – Cadastrar casa como ponto de arrecadação.**

Simas, 32 anos, deseja contribuir para uma causa usando sua residência como um ponto de arrecadação de doações. Assim ele entra no site e cadastra sua residência em conjunto com uma instituição.

- **Cenário 4 – Doar para uma instituição específica.**

Diogo, 41 anos, adora contribuir com caridades todo ano para uma causa específica, mas infelizmente tem tido pouco tempo para doar propriamente para esta instituição. Ele acessa o site e se depara com a instituição para a qual queria doar, várias formas de contribuir da maneira que preferir e de maneira eficiente.

- **Cenário 5 – Usar outro lugar como ponto de arrecadação.**

Magali, 52 anos, possui uma instituição de caridade, mas seu ponto de arrecadação é longe demais da zona urbana. Acessando o site, Magali então consegue ter outros pontos de arrecadação e então facilitar as doações para sua instituição.

- **Cenário 6 – Achar endereços para deixar doações.**

Simba, 38 anos, deseja doar para uma instituição, mas não sabe onde deve deixar suas doações. Através do site, Simba tem possivelmente mais de uma opção de endereços para deixar suas doações.

2.2. Personas



Figura 2 - Personas 1, 2 e 3.



Figura 3— Person4, 5 e 6

2.3. Acessibilidade

As cores utilizadas no layout foram escolhidas com base em critérios de alto contraste, permitindo que pessoas com deficiências visuais, como o daltonismo, possam acessar o conteúdo sem dificuldades. Essa escolha contribui para a legibilidade e a compreensão das informações apresentadas.

Além disso, o site dispõe de um recurso de tradução em Libras (Língua Brasileira de Sinais), disponibilizado em todas as páginas por meio de um ícone fixo na lateral da tela, atendendo às necessidades de usuários surdos e promovendo maior inclusão comunicativa.

Para ampliar a acessibilidade, foi implementado um botão de leitura automática do conteúdo das páginas, possibilitando que pessoas com dificuldades de leitura ou deficiências visuais possam compreender as informações por meio da escuta. A navegação do site também pode ser realizada de forma integral por meio do teclado, atendendo às necessidades de usuários com limitações motoras ou que utilizam tecnologias assistivas.

2.4. Interface do Site - explicação

As cores e fontes escolhidas para o projeto do "Caritá" foram pensadas para refletir os valores e objetivos da plataforma de arrecadação de alimentos sendo uma plataforma amigável:

Cores:

Cor Primária (#F9C89B): Transmite acolhimento e conforto, essencial para incentivar a solidariedade dos doadores.

Cor Secundária (#F08F36): Vibrante e chamativa, utilizada para destacar ações importantes como "Doe Agora".

Texto Primário (#522916): Marrom escuro, transmite seriedade e confiança, essencial para garantir a credibilidade das informações.

Texto Secundário (#000000): Preto, proporciona clareza e contraste para facilitar a leitura dos títulos e informações destacadas.

Fonte Poppins: A fonte Poppins foi escolhida pela sua legibilidade e contornos suaves, o que a torna amigável, enquanto mantém uma aparência profissional e limpa.

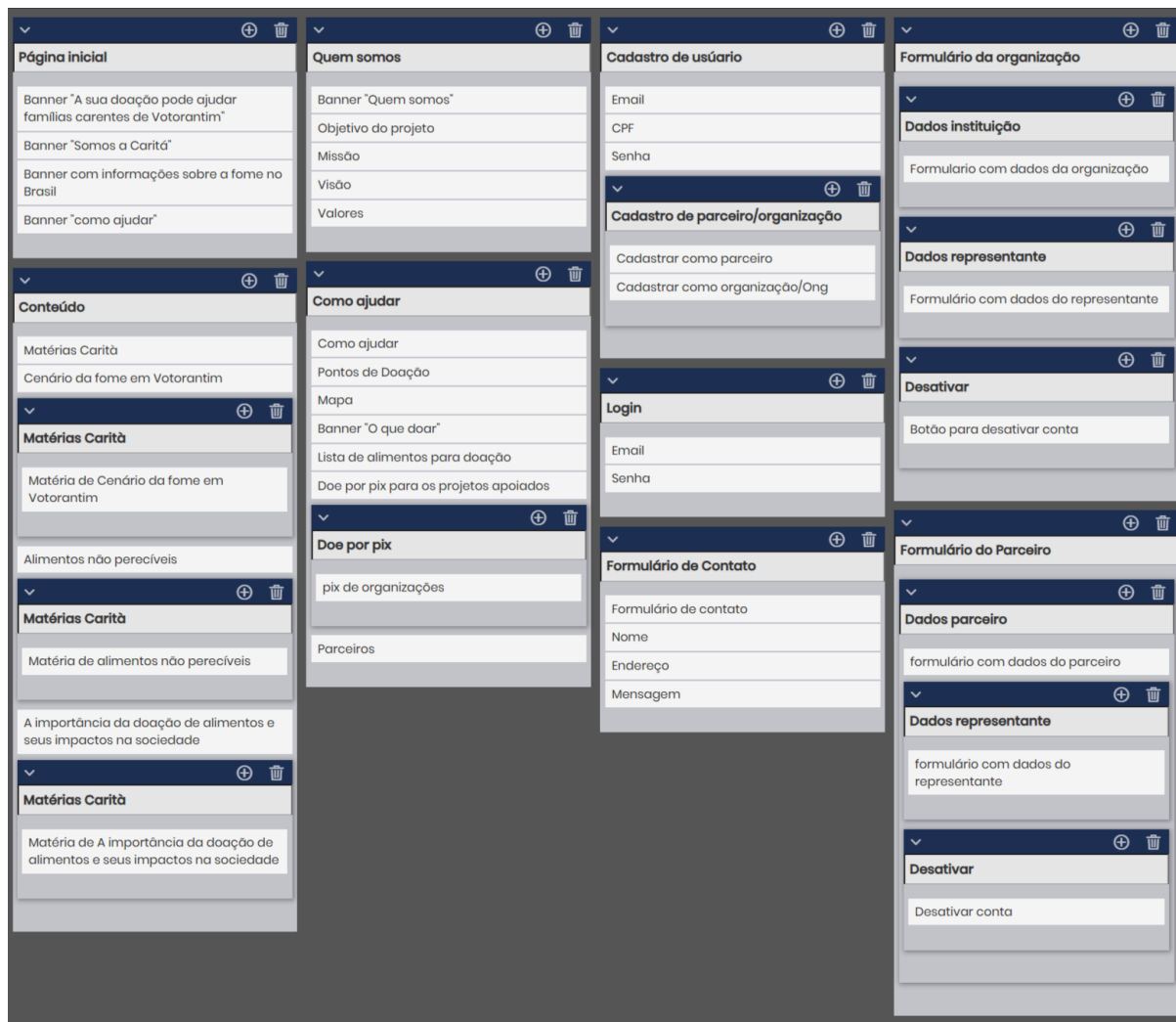
#F9C89B	#F08F36	#522916	#000000	#FFFFFF
Cor Primária	Cor Secundária	Texto Primária	Texto Secundária	Texto Alternativo



Fonte – Poppins

Figura 4 – Cores, Logo e Fonte Caritá.

2.5. CardSorting



Quadro 1 – CardSorting.

2.6. Definição de símbolos

Redes Sociais



Figura 5 - redes

Cada um dos símbolos representa acesso as redes sociais associadas a eles. O símbolo da esquerda representa o acesso ao Facebook enquanto o da direita representa o acesso ao Instagram.

Esses símbolos foram utilizados por conta do conhecimento geral sobre eles, é possível dizer que qualquer um que já mexeu em um celular já deve ter visto pelo menos um.

Acessibilidade



Figura 6 Acessibilidade

Este símbolo adicionado representa a opção de leitura do site via Libras, o símbolo é usado por sites populares de utilidade pública, como o GOV.

Este símbolo foi escolhido por dois fatores:

- Este símbolo já era predeterminado pelo sistema externo que foi implementado para maior acessibilidade.
- O símbolo também é comumente utilizado para representar a Língua Brasileira de Sinais (LIBRAS), o que facilita o processo para quem já é familiarizado com isso e necessita de ajuda.

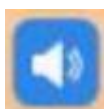


Figura 7 Acessibilidade

Este ícone representa a funcionalidade de leitura em voz alta dos textos presentes na página atual. Ao clicar sobre ele, o usuário ouve todo o conteúdo textual visível, promovendo maior acessibilidade para pessoas com deficiência visual, dislexia ou outras dificuldades de leitura.

O símbolo segue o mesmo padrão visual dos demais componentes de acessibilidade do site, utilizando a mesma cor e estilo, garantindo uma identidade visual coerente e facilitando o reconhecimento dessas funcionalidades.



Figura 8 Acessibilidade

Este ícone ativa a conversão automática do site para tons de cinza, facilitando a navegação para pessoas com daltonismo, sensibilidade a cores ou outras condições visuais que dificultam a distinção de cores.

2.7. Design system

2.7.1. Identidade Visual

- A logo apresenta uma paleta multicolorida em tons terrosos, com o nome “Caritá” em tipografia manuscrita.
- Utilizada em dois tamanhos: no topo (menu) e no rodapé.

Cores Primárias

Cor	Hexadecimal	Uso Principal
Laranja Claro	#F4C295	Fundos de seções, botões.
Laranja Escuro	#C97D4D	Títulos de seções e destaque.
Branco	#FFFFFF	Fundo geral.
Preto/Cinza Escuro	#333333 ou similar	Textos principais.

Tipografia

- **Primária:** Sans-serif (parece Montserrat ou Poppins).
- **Títulos:** Caixa alta, negrito.
- **Corpo do texto:** Regular, alinhamento à esquerda.
- **Fonte manuscrita para a marca “Caritá”** (exclusiva na logo e assinaturas visuais).

2.7.2. Componentes UI

- **Botão Primário:**
 - Cor: Laranja (#F4C295).
 - Texto centralizado, caixa alta.
 - Borda arredondada.
- **Botão Secundário (ex: “Saiba mais”):**
 - Fundo branco com borda laranja.
 - Hover: escurece ou inverte as cores.

2.7.3. Cards

- Utilizados para:
 - Dados estatísticos.
 - Notícias e matérias.
 - Missão, visão e valores.
- Fundo branco com sombra leve.
- Ícones vetoriais ou imagens no topo.

2.7.4. Layout

Estrutura

- **3 colunas principais** para exibição em diferentes páginas.
- **Topo fixo:** com menu de navegação e logo.
- **Rodapé fixo:** com logo, redes sociais e chamada de ação.

Seções

- Hero Section com chamada clara: “A SUA DOAÇÃO PODE AJUDAR...”.
- Seções divididas com alternância de cor (branco / laranja claro).
- Área “Como Ajudar”, “Quem Somos”, “Matérias” com ilustrações ou fotos.
- Destaque para dados e estatísticas em layout horizontal com ícones.

2.7.5. Imagens e Ícones

Estilo de Imagens

- Fotografia realista e humanizada.
- Imagens de alimentos, mãos estendidas, doações etc.
- Transmitem solidariedade e ação social.

Ícones

- Estilo outline, amigáveis.
- Usados para missão, visão, valores e dados estatísticos.

2.7.6. Tom de Voz

- **Tom:** Acolhedor, humano, direto e solidário.
- **Vocabulário:** Positivo, com palavras como “ajudar”, “famílias”, “doar”, “missão”, “impacto”.
- **Objetivo:** Engajar usuários a participarem, com foco na empatia e solidariedade.

2.7.7. Navegação

Menu Principal

- Itens: Home | Sobre | Matérias | Contato | Login.
- Layout horizontal, alinhado à direita.

Rodapé

- Logo Caritá.
- Redes sociais (Facebook, Instagram).
- Reforço da mensagem e identidade.

2.7.8. Responsividade

- Stack vertical de seções e imagens.

3. Gestão ágil

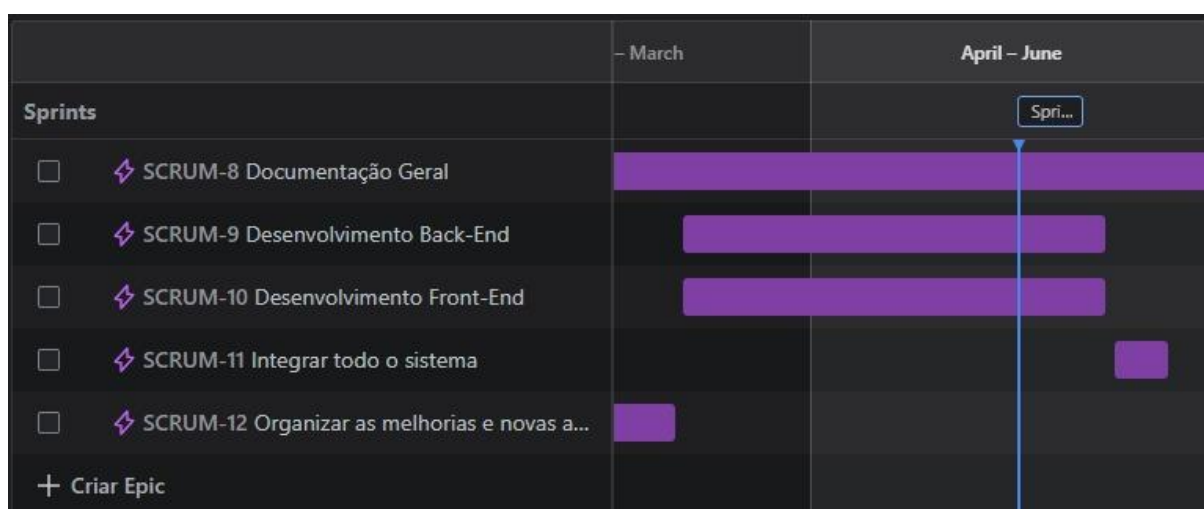
A organização do projeto seguiu os princípios da metodologia ágil, com foco em entregas incrementais e colaboração contínua entre os membros da equipe. As funcionalidades foram priorizadas e desmembradas em user stories claras, permitindo melhor entendimento das necessidades dos usuários. As tarefas foram distribuídas em sprints, iniciadas desde o levantamento de requisitos e definições até a entrega completa do projeto, com acompanhamento por meio de um quadro no Trello, facilitando a visualização do progresso, ajustes rápidos e cumprimento dos prazos estabelecidos.

3.1. User Stories

Épico	Ator	Ação	Motivo
RF01	Usuário representante de uma instituição ou parceiro	cadastra um usuário com login, CPF e senha	Ter acesso a área administrativa e cadastrar as informações necessárias.
RF02	Usuário representante da instituição	faz cadastro como instituição.	Ter a organização listada e receber doações.
RF03	Usuário representante do parceiro	faz cadastro como parceiro.	Contribuir divulgando ou recebendo doações.
RF04	Usuário logado	preencher os formulários de cadastro da instituição e local de doação.	Fornecer todas as informações necessárias e ser validado.
RF05	Parceiro	Desativa sua própria conta.	Retirar seu próprio acesso às funcionalidades do site.
RF06	Parceiro	Atualiza os dados de sua própria conta.	Esperar a validação dos administradores e alterar os dados registrados para correção ou atualização do usuário para com o site.
RF07	Parceiro	Sugere parceria de empresa e site.	Esperar a validação dos administradores e se divulgar no site.
RF08	Parceiro	Desfaz sua parceria com o site.	Retirar as informações de sua empresa do site.

RF09	Parceiro	Entra em contato através do formulário de contato.	Para fazer solicitações em caso que necessita a interferência da equipe de suporte
RF10	Visitante do site	Acesso o mapa de pontos de arrecadação	Para visualizar pontos próximos em que posso doar os alimentos

3.2. Sprints



Quadro 2 – Sprints.

Sprint 1 – Levantamento de Requisitos e Documentações - (01/03 a 31/03)

Durante a fase de desenvolvimento, as sprints foram organizadas com base nas prioridades identificadas no projeto. Inicialmente, analisamos a estrutura do frontend já implementado em Angular, o que nos permitiu visualizar as principais funcionalidades existentes e os pontos de melhoria. A partir dessa análise, definimos as APIs necessárias para atender às funcionalidades do sistema, como toda a parte de cadastro, atualização, listagem e desativação das tabelas definidas no SQL (Usuário, Parceiros, Ponto de Arrecadação e Instituição).

A sprint foi planejada com o foco de:

- Definir as APIs necessárias para integração do backend
- Criação e ajustes nos formulários para cadastro de doações e pontos de arrecadação.
- Implementação de melhorias na interface, visando melhor usabilidade e acessibilidade.
- Levantamento de requisitos.
- Detalhamento da documentação

Com essa abordagem, foi possível evoluir de forma gradativa para a definição inicial dos requisitos necessários e com o andamento do projeto.

Sprint 2 – Atualização e melhorias no Front-End (01/04 - 30-04)

No frontend, desenvolvido com angular, foram realizadas diversas atualizações para integrar corretamente as funcionalidades do backend e proporcionar uma experiência mais completa e acessível ao usuário. Com base nas APIs definidas, as telas foram pensadas para facilitar a navegação e permitir o uso completo das funcionalidades implementadas no backend.

Foram adicionadas validações reativas nos formulários, garantindo que os dados fossem preenchidos corretamente antes do envio. Além disso, implementamos tratativas de erro e mensagens de feedback, possibilitando que o usuário compreenda eventuais falhas (como campos obrigatórios não preenchidos ou erros de servidor) e possa corrigi-las de forma clara.

Também foram aplicadas melhorias voltadas à acessibilidade, como uso adequado de contraste de cores, inclusão de textos alternativos em imagens, navegação por teclado e uso de elementos semânticos do HTML. Essas práticas garantem que o sistema seja acessível a pessoas com deficiência visual ou motora, ampliando o alcance da aplicação.

Por fim, melhorias de layout e organização das informações contribuíram para uma navegação mais fluida, tornando o site mais responsivo e intuitivo.

Sprint 3 – Desenvolvimento do Backend - (01-05 A 31-05)

O desenvolvimento do backend teve como objetivo fornecer as funcionalidades necessárias para dar suporte às operações do site de doações, garantindo segurança, escalabilidade e integração com o frontend. A arquitetura foi estruturada utilizando Node.js com o framework Express, permitindo a criação de rotas organizadas, controllers e modelos de dados de forma modular.

Inicialmente, foram definidas as principais APIs com base nas necessidades levantadas durante a etapa de planejamento, como todas as operações de CRUD para usuários, cadastro de parceiros, instituições e pontos de arrecadação. Em seguida, foram implementadas as funcionalidades de atualização de dados, onde utilizamos o ORM Sequelize para facilitar a comunicação com o banco de dados MySQL, permitindo modelar e gerenciar as tabelas de forma mais prática e organizada, além de utilizar migrations para controle de versões do banco de dados.

Durante o desenvolvimento, também foram aplicadas boas práticas como tratamento de erros, uso de middlewares, validações de entrada e organização do código em camadas, garantindo manutenibilidade e clareza ao projeto.

Sprint 4 - Integração do sistema e melhorias (01-06 a 20-06)

3.3. Backlogs

<input type="checkbox"/> ▾	Backlog (28 tickets)	00
<input checked="" type="checkbox"/>	CAR-54 GET, POST, PUT de usuários.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-55 GET, POST, PUT de pontos de Arrecadação.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-56 Validação no Back-End de usuários.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-57 Filtros no Maps.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-58 Rotas de usuários.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-59 Documentação API.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-60 Documentação de IHC.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-61 Sprints detalhadas no GitHub.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-62 Documentação completa.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-63 Integrar Banco de Dados ao Back-End.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-64 Rotas de pontos de arrecadação.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-65 Validação do Back-End de Organizações.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-66 Inserir Botões de atualizar para parceiros e organizações.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-67 Atualizar Div do Ponto de Arrecadação no Front-End.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-68 Fazer validação no Back-End de pontos de arrecadação.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-69 Fazer validação no Back-End de parceiros.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-70 Atualizar componentes do Front-End.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-71 Fazer o Jira.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-72 GET, POST e PUT de parceiros.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-73 Fazer toda acessibilidade.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-74 Documentação de técnicas de programação.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-75 Fazer todo o ReadMe.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-76 Rotas de Organizações.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-77 Linkar o Email com a página de formulário.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-78 Validação do Login e cadastro no Front-End.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-79 Criar as redes sociais do projeto.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-80 Vincular Maps no Front-End.	FEITO NO 3º ...
<input checked="" type="checkbox"/>	CAR-81 Modelagem do banco de dados.	FEITO NO 3º ...

Quadro 3 – Backlog.

3.4. Resultados alcançados

Primeiramente foram feitos os objetivos que foram anteriormente postergadas. Dentre elas estão:

- Validação no Back-End.
- Implementação de APIs.
- Ligação de Front-End e Back-End.
- Vinculação do mapa no site.

Para que fosse possível prosseguir, primeiro foi usado um kanban via Trello e posteriormente Jira. Tendo as tarefas bem definidas começamos a trabalhar nas implementações exigidas. A maioria das implementações pertenceram ao Back-End e foram feitas alterações no Front-End para acomodar certas mudanças, assim como no banco de dados.

Conforme feitas as tarefas seguimos documentando toda nossa experiência, algumas outras tarefas foram executadas sem a necessidade de uma ordem certa:

- Criação das redes sociais.
- Idealização e iniciação do ReadMe.
- Link do Email a página de usuário.
- Documentações de API, IHC e Técnicas de Programação.
- Modelagem do banco de dados.
- Atualização de componentes no Front-End.
- Implementação de Acessibilidade no site.

Posteriormente adicionamos as Sprints detalhadas ao GitHub, Acabamos a atualização do Front-End e pudemos enfim focar no Back-End, que por si só tomou a maior parte do tempo no processo.

3.5 Requisitos do projeto

3.5.1. Levantamento de Requisitos

Foi realizada uma pesquisa para verificar a existência de sites semelhantes em Votorantim, não foi encontrada nenhuma plataforma específica que indicasse todos os pontos de doação de alimentos na cidade. Além disso, pesquisas informais com amigos e familiares, mostraram que muitos enfrentavam dificuldades para encontrar pontos de doação devido à falta de divulgação.

3.5.2. Requisitos funcionais

Requisitos Funcionais		
Nº. Requisito	Identificação	Objetivo
RF001	Cadastro de usuário	O sistema deve permitir que usuários (pessoas físicas) criem uma conta, preenchendo dados obrigatórios como nome completo, CPF, e-mail, telefone e senha. Após o cadastro, o usuário poderá acessar funcionalidades restritas da plataforma.
RF002	Validação CNPJ/CPF	O sistema deve validar o número de CPF (para pessoa física) ou CNPJ (para pessoa jurídica), garantindo que os dados sejam reais por meio de consulta a uma API ou verificação de estrutura.
RF003	Validação de senha	O sistema deve verificar se a senha criada pelo usuário atende aos critérios de segurança definidos, como número mínimo de caracteres.

RF004	Login	O sistema deve permitir que usuários e instituições acessem suas contas por meio de autenticação com e-mail/CNPJ/CPF e senha, verificando as credenciais e redirecionando para o ambiente correspondente.
RF005	Acessar o perfil	O sistema deve disponibilizar uma área de perfil onde o usuário ou representante da instituição possa visualizar, editar ou atualizar informações cadastrais registradas.
RF006	Cadastro de instituição	O sistema deve permitir que representantes legais registrem uma nova instituição, preenchendo informações como razão social, CNPJ, endereço, área de atuação e responsável legal. Os dados devem ser armazenados e exibidos após validação.
RF007	Cadastro de ponto de doação	O sistema deve possibilitar que usuários autorizados cadastrem novos pontos de doação, informando localização, dias e horários de funcionamento e o responsável pelo ponto. Esses pontos serão exibidos no site após validação.

RF008	Validação de Cadastro	O sistema deve validar o cadastro de instituições ou parceiros com base nos documentos fornecidos (como CNPJ, RG), aprovando ou recusando conforme os critérios estabelecidos pela administração do sistema.
RF009	Desativação	O sistema deve oferecer a opção de desativação da conta por parte do usuário ou administrador. A conta desativada ficará indisponível para login e exibição pública.
RF010	Gerenciar parceiros	O sistema deve permitir a administração de instituições parceiras, possibilitando o cadastro, visualização, edição das informações e desativação de parcerias ativas.
RF011	Gerenciamento de pontos de doação	O sistema deve permitir que pontos de doação sejam gerenciados por usuários com permissão, garantindo que apenas informações corretas e atualizadas estejam disponíveis no site.
RF012	Maps	Exibição do google maps no site.
RF013	Formulário de contato	O sistema deve exibir uma página com um formulário para que os usuários possam enviar dúvidas, sugestões ou mensagens à equipe responsável pela plataforma. As mensagens devem ser enviadas por e-mail.

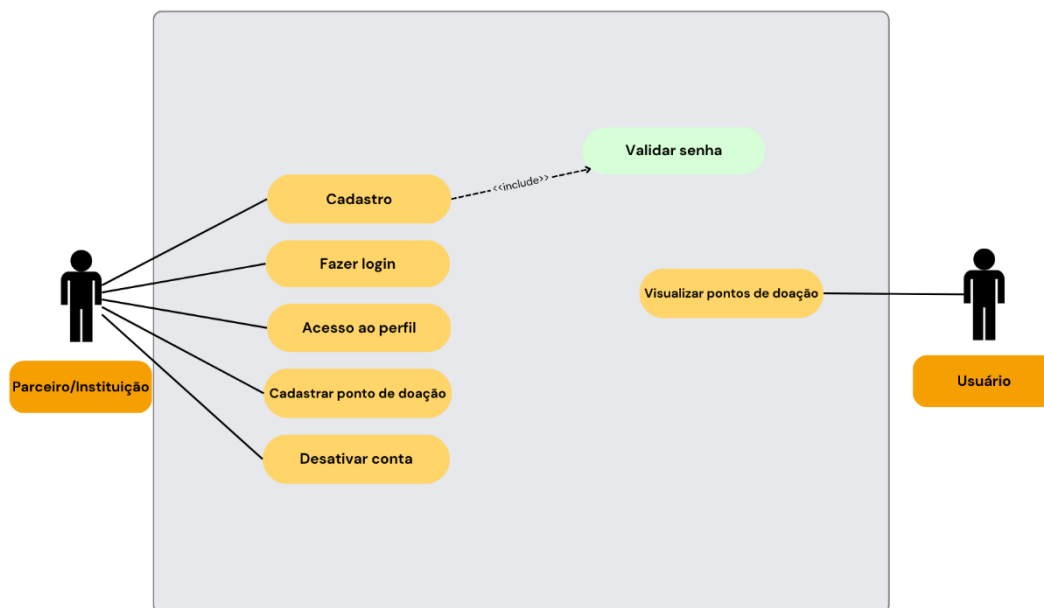
RF014	Anexar arquivos	O sistema deve permitir que usuários e instituições anexem documentos durante o processo de cadastro ou validação, como RG, comprovante de endereço.
RF015	Validar troca de responsável	O sistema deve oferecer uma funcionalidade para que instituições possam solicitar a troca do responsável legal, e validar a solicitação por meio da análise de documentos comprobatórios enviados no processo.

3.5.3. Requisitos não funcionais

Requisitos Não Funcionais		
Nº Requisito	Nome	Descrição
RNF001	Usabilidade	A interface do sistema deve ser desenvolvida com HTML e CSS, proporcionando uma experiência de uso intuitiva, acessível e fácil de navegar para todos os públicos.
RNF002	Disponibilidade	O sistema deve estar disponível para uso pelo menos 99,9% do tempo, com monitoramento contínuo e comunicação prévia aos usuários sobre manutenções programadas.
RNF003	Dependabilidade	O sistema deve ser capaz de se recuperar automaticamente em caso de falhas, e realizar backups regulares para garantir a integridade e segurança das informações armazenadas.

RNF004	Linguagem de Programação	Utilizar JavaScript e TypeScript como linguagens principais, visando garantir consistência, escalabilidade e facilidade na manutenção do código.
RNF005	Frameworks	Utilizar os frameworks Angular e Bootstrap para garantir interfaces modernas, responsivas e com melhor experiência do usuário em diferentes dispositivos.
RNF006	Banco de Dados	Utilizar o banco de dados relacional MySQL para armazenar, gerenciar e manipular as informações de forma segura e eficiente.
RNF007	Versão Web	O sistema deve contar com uma versão adaptada para acesso via navegadores de desktop, garantindo uma boa experiência em resoluções maiores.
RNF008	Versão Mobile	A interface deve ser responsiva, garantindo funcionamento adequado e boa usabilidade em dispositivos móveis, como smartphones e tablets.
RNF009	Acessibilidade	Seguir as diretrizes WCAG 2.1 para garantir acesso a pessoas com deficiência (uso de leitores de tela, contraste, navegação por teclado).
RNF010	Desempenho	O sistema deve carregar completamente em até 3 segundos em conexões padrão

3.6. Diagrama de caso de uso



Quadro 4 – Diagrama de caso de uso

3.7. Projeto de Software

Arquitetura da Aplicação

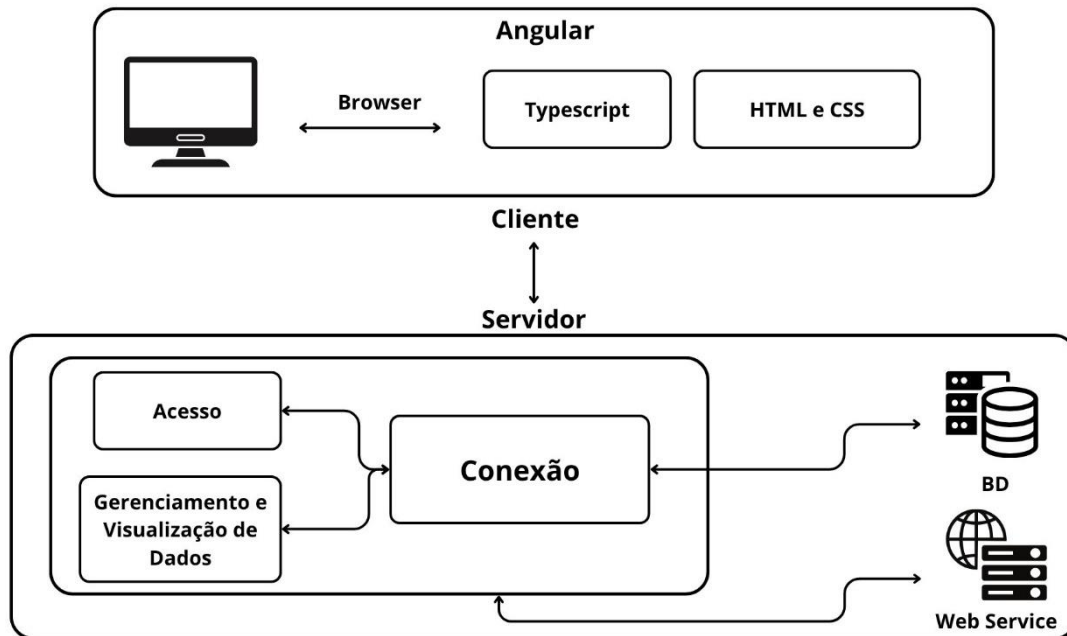


Figura 9– Projeto de Software.

A arquitetura de software foi dividida em duas principais camadas: cliente e servidor. Na camada cliente, a aplicação é desenvolvida com o framework Angular, que utiliza tecnologias como TypeScript para a lógica da aplicação e HTML com CSS para estrutura e estilo da interface. O usuário interage com o sistema por meio de um navegador (browser), que se comunica com o servidor através de requisições.

A camada Servidor é responsável por processar essas requisições, aplicar regras de negócio e interagir com fontes de dados. Ela que controla autenticação e permissões, que organiza as informações a serem processadas e exibidas e que realiza a comunicação com o banco de dados (BD) e com web services externos. O servidor recebe os dados enviados pelo cliente, processa as informações conforme a lógica implementada e retorna uma resposta adequada para ser exibida no navegador.

Essa arquitetura evidencia uma separação que permite uma maior organização, reutilização de código e facilidade na manutenção e evolução do sistema.

3.8. Tecnologias utilizadas

HTML5: Linguagem de programação front-end para criação de aplicações web, que permite a estruturação semântica do conteúdo.

CSS3: Estilização e design responsivo, garantindo que a interface se adapte a diferentes dispositivos.

JavaScript: Linguagem de programação para adicionar funcionalidades dinâmicas à aplicação, permitindo interatividade.

TypeScript: Oferecer tipos estáticos, o que facilita a manutenção e a legibilidade do código.

Angular: Framework escolhido para o desenvolvimento de aplicações web interativas, permitindo uma experiência rica para os usuários.

Bootstrap: Utilizado para garantir um design responsivo e fornecer componentes prontos que melhoram a usabilidade.

MySQL: Banco de dados, responsável pelo armazenamento de dados e que inclui suporte a backup e recuperação, assegurando a integridade das informações.

Astah: Ferramenta de diagramação e modelagem UML que permite a criação de uma variedade de diagramas

BRModelo: Ferramenta voltada para o desenvolvimento de projetos de banco de dados relacional, incluindo as etapas conceitual, lógico e físico.

Trello: Ferramenta visual para o gerenciamento do time, definição do Backlog do projeto e monitoramento de tarefas.

Jira: Ferramenta para organização de Backlog e Sprints.

GitHub: Ferramenta para hospedagem do código-fonte e README.

Express: Framework web para Node.js, utilizado na construção do back-end da aplicação. Ele facilita a criação de rotas, middleware e integração com banco de dados, sendo responsável por receber e processar as requisições da API.

3.9. Integração e Papéis da Equipe

Mayara Barros: Product Owner

Ariele Peres: Dev Team

Vanessa Capuano: Dev Team

Fernando Ribeiro: Dev Team

Guilherme Tobias: Scrum Master

Guilherme Pazetti: DBA

4. Técnicas de programação

Introdução

O projeto Caritá surgiu como uma iniciativa pensando na dificuldade de se obter informações sobre ações de doação de caridade e como as ajudar.

Idealizamos o site como uma forma de instituições de caridade se conectarem com pessoas interessadas em realizarem doações.

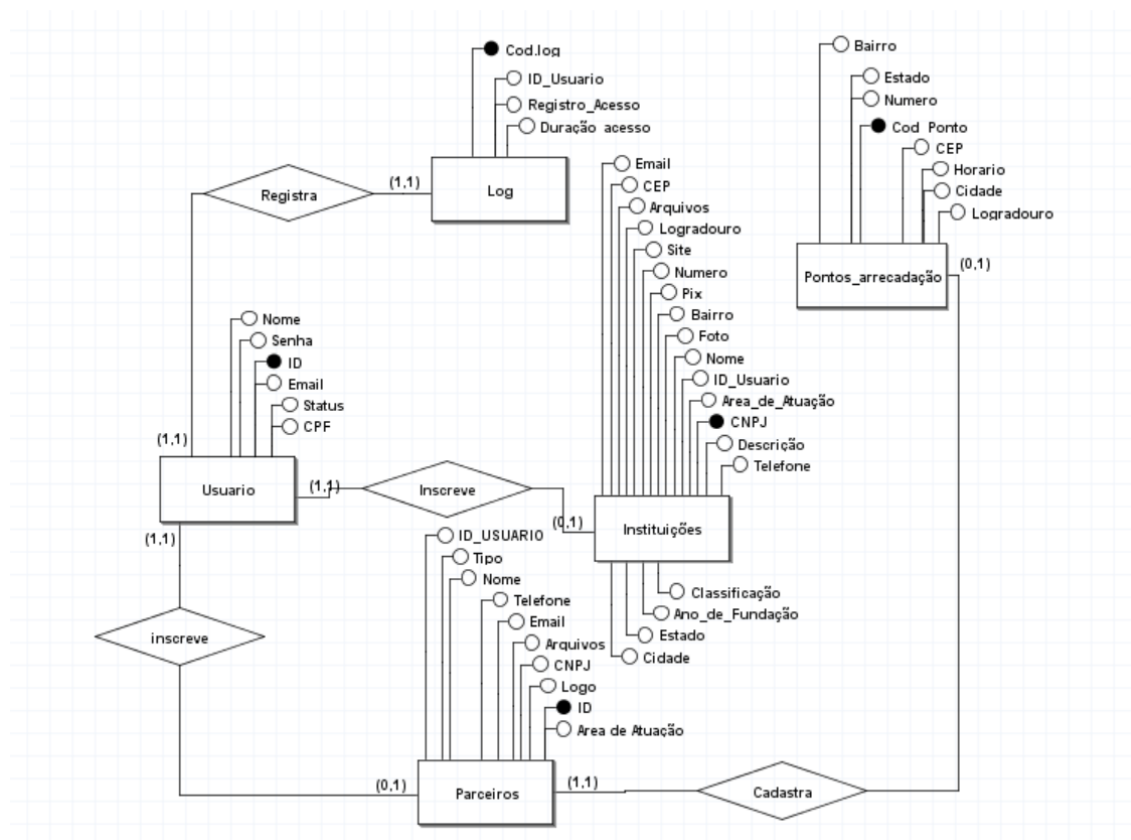
4.1. Modelagem de banco de dados

Foram realizadas atualizações no Diagrama Entidade-Relacionamento (DER), no modelo lógico e no diagrama de classes, com base na evolução das funcionalidades desenvolvidas neste semestre.

O modelo lógico foi revisado para manter a consistência com o banco de dados MySQL utilizado com Sequelize, garantindo integridade referencial e estrutura adequada para as operações do sistema. O diagrama de classes também foi atualizado para representar com mais fidelidade a estrutura do backend atual, auxiliando na organização do código e na clareza da implementação das funcionalidades.

4.2. DER

O diagrama representa um sistema onde os usuários da plataforma de doações podem se cadastrar, logar, registrar instituições e parceiros. Os usuários são responsáveis por registrar instituições e parceiros no sistema e tem acesso aos logs de acesso do sistema. Os parceiros podem cadastrar pontos de arrecadação para receber as doações e fazer o repasse para as instituições. As instituições (ongs) possuem os dados cadastrais para ficarem visíveis no site e recebem as doações dos usuários parceiros.



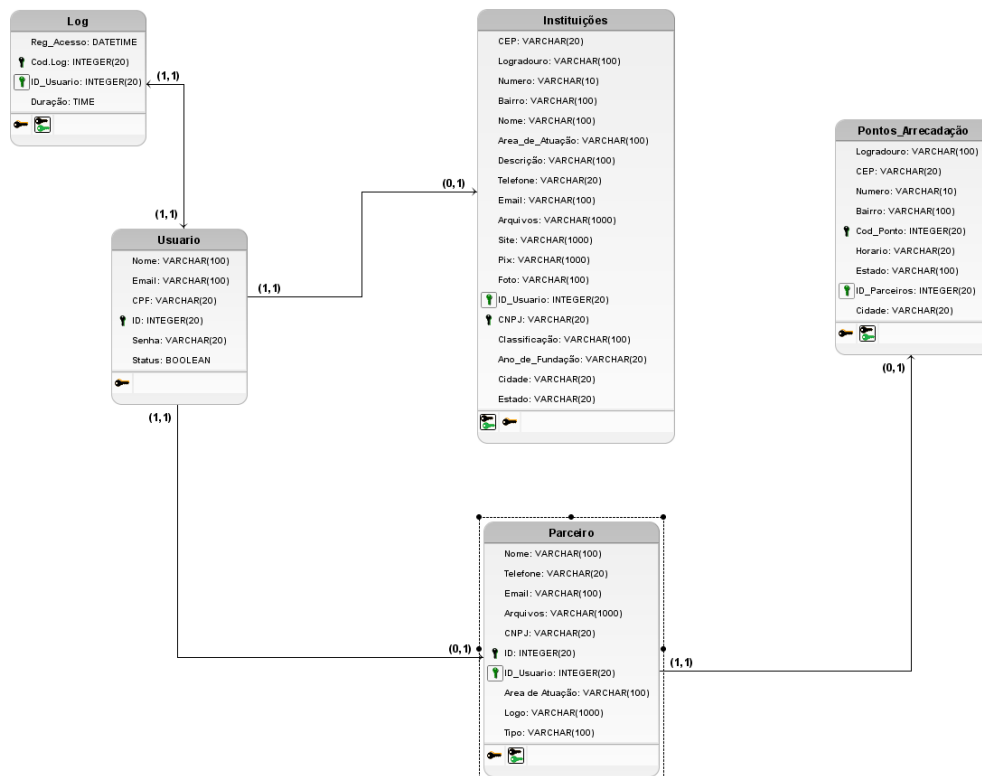
Quadro 5 – Diagrama de caso de uso

4.3. Modelo Lógico

O modelo lógico proposto desenvolvido através do Diagrama Entidade-Relacionamento e representa a estrutura que será implementada no Banco de dados relacional. Ele contempla tabelas como:

- Usuários que se cadastra como o representante e cadastra a instituição que receberá a ong ou o parceiro divulgador/captador, como usuário administrativo terá acesso aos registros da tabela de logs.
- As instituições possuem todos os dados completos para serem validados para recebem as doações
- Parceiros que podem se cadastrar como parceiro captador que cadastra os pontos de arrecadação para receber as doações, ou como parceiro divulgador que auxilia com a parte de divulgação e propagação do site.
- Ponto de arrecadação são os locais cadastrados pelos parceiros para receber as doações e serão exibidos no site onde os visitantes da plataforma poderá verificar os pontos próximos a ele.

O modelo foi estruturado com **chaves primárias e estrangeiras** para garantir a integridade dos dados, permitindo o controle preciso de relacionamentos e a rastreabilidade de cada cadastro no sistema.



Quadro 6 – Modelo Lógico.

4.4. Diagrama de classe

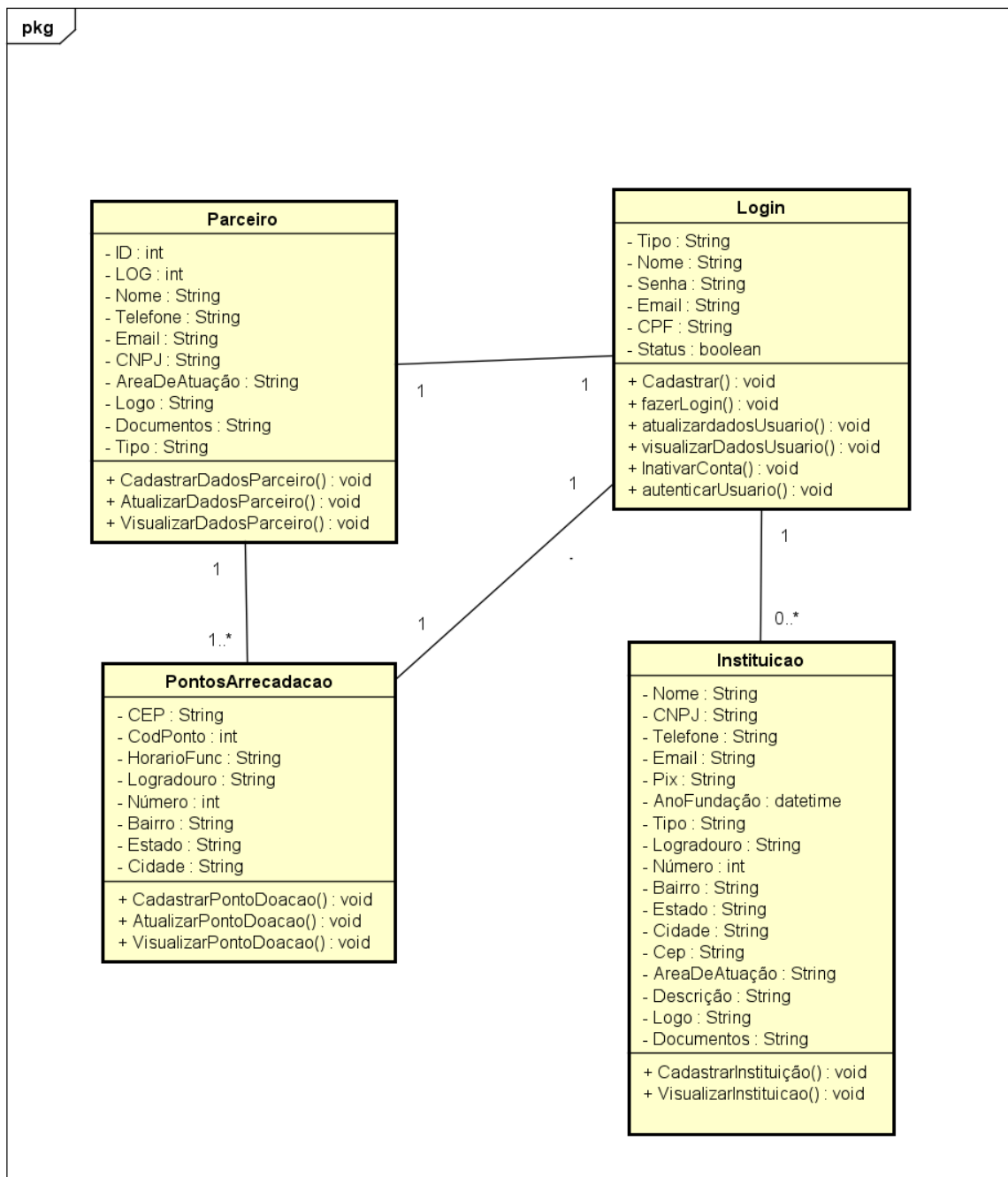
O diagrama de classe apresentado modela a estrutura do sistema com as entidades voltadas para gestão de usuários, instituições, parceiros e pontos de arrecadação.

A classe Login representa o cadastro do usuário representante no sistema e contempla os dados necessários para o cadastro na plataforma, os métodos presentes nessa entidade são o de login e controle de acesso.

A instituição possui informações para o cadastro da instituição no site para ser validada como uma e estar apta a receber as doações.

O parceiro representa as empresas que desejam ser parceiras divulgando ou captando os alimentos.

A classe PontosArrecadação detalha os locais de doação ligado aos parceiros, onde será captado os alimentos para a distribuição para ongs.



Quadro 7 – Diagrama de Classe.

4.5. Atributos

Parceiros

ID: Int que armazena o id único de cada parceiro

LOG: Int que armazena o momento de acesso do parceiro

Nome: String que armazena o nome do parceiro

Telefone: String que armazena o número de contato do parceiro

Email: String que armazena o E-mail de contato do parceiro

CNPJ: String que armazena o CNPJ do parceiro

Área de atuação: String que recebe as informações da área de atuação do parceiro

Logo: String que recebe o link da logo do parceiro

Documentos: String que recebe os documentos para validação do parceiro

Tipo: String que recebe o tipo de parceiro

Login

Tipo: String que especifica se o login é de parceiro ou organização

Nome: String que especifica o nome de quem está logando

Senha: String que recebe a senha de login

Email: String que especifica o email de quem está logando

CPF: String que especifica o CPF de quem está logando

Status: Boolean que identifica se o login está ativo ou não

Pontos de Arrecadação

CEP: String que identifica o cep do endereço

CodPonto: Int que identifica o código único do ponto

HorarioFunc: String que identifica o horário de funcionamento do ponto

Logradouro: String que armazena o logradouro do ponto

Número: Int que armazena o número físico do ponto

Bairro: String que armazena o bairro em que se localiza o ponto

Estado: String que armazena o estado em que o ponto se localiza o ponto

Cidade: String que armazena a cidade em que se localiza o ponto

Instituição

Nome: String que identifica o nome da instituição

CNPJ: String que identifica o CNPJ da instituição

Telefone: String que identifica o número de contato da instituição

Email: String que identifica o endereço de e-mail da
instituição

Pix: String que identifica a chave PIX da instituição

AnoDeFundação: Datetime que identifica o ano de fundação da empresa

Tipo: String que identifica o que a instituição faz

Logradouro: String que identifica o logradouro da instituição

Número: Int que identifica o número físico da instituição

Bairro: String que identifica o bairro onde se localiza a instituição

Estado: String que identifica o estado de localização da instituição

Cidade: String que identifica a cidade em que a instituição se localiza

CEP: String que identifica o cep

AreaDeAtuação: String que detalha a área de atuação da instituição

Descrição: String que recebe uma breve descrição da instituição

Logo: String que recebe o link da logo da instituição

Documentos: String que recebe os documentos para validação da instituição

Parceiros

CadastrarDadosParceiro(): Função para cadastrar o parceiro no site

AtualizarDadosParceiro(): Função para atualizar os dados do parceiro já cadastrado

VisualizarDadosParceiro(): Função de visualização dos dados do parceiro já cadastrado

Login

Cadastrar(): Processo do qual o usuário cadastra suas informações para formar um perfil no site

fazerLogin(): Processo do qual o usuário faz login com suas credenciais cadastradas no site

autenticarUsuario(): Processo do qual o sistema faz os processos de validação de um usuário para formalizar seu cadastro

inativarConta(): Inativa a conta cadastrada

atualizardadosUsuario(): Função para atualizar os dados cadastrados do usuário

visualizarDadosUsuario(): Função para visualizar os dados cadastrados do usuário

Pontos de arrecadação

CadastrarPontoDoacao(): Processo do qual o usuário cadastra um ponto de doação

VisualizarPontoDoacao(): Visualização do ponto de doação no site

AtualizarPontoDoacao(): Processo para atualização dos dados cadastrados

Instituição

VisualizarInstituicao(): Processo para visualização da instituição no site

AlterarInstituicao(): Processo para alteração dos dados cadastrados

CadastrarInstituição(): Processo de cadastro da instituição no site