

CENTRO PAULA SOUZA
Etec DE TABOÃO DA SERRA
Técnico em Desenvolvimento de Sistemas

Alice Chinoque de Souza
Carolina Fernandes
Diego de Sousa Matias
Isabella Gomes Menezes
Nicollas Isaac Queiroz Batista
Sarah Macedo de Oliveira

GREENPOINT: uma solução para a reduzir a insegurança alimentar

Taboão da Serra - SP
2022
CENTRO PAULA SOUZA

ETEC DE TABOÃO DA SERRA
Técnico em Desenvolvimento de Sistemas

Alice Chinoque de Souza
Carolina Fernandes
Diego de Sousa Matias
Isabella Gomes Menezes
Nicollas Isaac Queiroz Batista
Sarah Macedo de Oliveira

GREENPOINT: uma solução para a reduzir a insegurança alimentar

Trabalho de Conclusão de Curso apresentado à Etec de Taboão da Serra, do Centro Estadual de Educação Tecnológica Paula Souza, como requisito para a obtenção da habilitação profissional de Nível Técnico em Desenvolvimento de Sistemas sob a orientação dos Professores Fabiano Jorge Oliveira Lopes e Rodolfo Votto Filho.

Taboão da Serra - SP
2022

Alice Chinoque de Souza
Carolina Fernandes

Diego de Sousa Matias
Isabella Gomes Menezes
Nicollas Isaac Queiroz Batista
Sarah Macedo de Oliveira

GREENPOINT: uma solução para a reduzir a insegurança alimentar

Trabalho de Conclusão de Curso - TCC

Aprovada em: _____ / _____ / _____

Conceito: _____

Banca Examinadora:

Professor.....
Etec de Taboão da Serra
Orientador

Professor
Etec de Taboão da Serra

Professor
Etec de Taboão da Serra

TABOÃO DA SERRA – SP
2022

DEDICATÓRIA

Dedicamos este trabalho às pessoas em situação de insegurança alimentar do CRAS Trianon, às ONGs e órgãos públicos que os auxiliam cotidianamente.

AGRADECIMENTOS

Agradecemos a Deus, aos nossos familiares, aos nossos orientadores, Fabiano Jorge Oliveira Lopes e Rodolfo Votto Filho e à funcionária da Prefeitura de Taboão da Serra, Nalva Santos Araujo, que foram fundamentais no desenvolvimento deste trabalho.

EPÍGRAFE

“Quando um pobre morre de fome, não é porque Deus não cuidou dele. É porque nem você nem eu quisemos lhe dar o que ele precisava.”

Santa Madre Teresa de Calcutá

RESUMO

A insegurança alimentar no Brasil é um problema que se mostra crescente e alarmante em todas as regiões e não possui a visibilidade que merece, uma vez que o Brasil possui 33,1 milhões de pessoas afetadas pela fome. Portanto, é necessária a devida atenção e, mesmo que de forma menos abrangente, criar um método de auxiliar aqueles que estão vivendo nessas condições. As ONGs que costumam auxiliar as famílias e arrecadar doações de alimentos, muitas vezes não conseguem o alimento ou até os possíveis receptores das doações, por isso, é necessário também um meio de realizar essa arrecadação e distribuição. Sendo assim, o aplicativo será criado com o objetivo de diminuir o índice de pessoas que estão em precariedade alimentar, por meio de um software e site, cadastrando ONGs, doadores e inclusive, as pessoas receptoras. A ideia surgiu após o alerta ao fato do Brasil ter retornado ao mapa da fome e após as pesquisas mostrarem que mais de 116 milhões de brasileiros fazem parte da população que possui problemas quanto à garantia de alimentos pela constituição de 1988.

Palavras-chave: Insegurança. Alimentos. Fome. Alimentação. Doação.

ABSTRACT

Food insecurity in Brazil is a problem that is growing and alarming in all regions and does not have the visibility it deserves, since Brazil has 33.1 million people affected by hunger. Therefore, it is necessary the due attention and, even if in a less comprehensive way, create a method to help those who are living in these conditions. The NGOs that usually help families and collect food donations often do not get the food or even the possible recipients of the donations, so it is also necessary to have a way to perform this collection and distribution. Thus, the application will be created with the aim of reducing the rate of people who are in food poverty, through a software and website, registering NGOs, donors and even the recipients. The idea came up after the alert to the fact that Brazil had returned to the hunger map and after research showed that more than 116 million Brazilians are part of the population that has problems regarding the guarantee of food by the 1988 constitution.

Keywords: Insecurity. Food. Hungry. Feeding. Donation.

RESUMEN

La inseguridad alimentaria en Brasil es un problema creciente y alarmante en todas las regiones y no tiene la visibilidad que merece, ya que Brasil tiene 33,1 millones de personas afectadas por el hambre. Por lo tanto, es necesario prestar la debida atención y, aunque sea de forma menos exhaustiva, crear un método para ayudar a los que viven en estas condiciones. Las ONGs que habitualmente ayudan a las familias y recogen las donaciones de alimentos a menudo no consiguen los alimentos o incluso los posibles destinatarios de las donaciones, por lo que también es necesaria una forma de realizar esta recogida y distribución. Así, la aplicación se creará con el objetivo de reducir el índice de personas que se encuentran en situación de escasez de alimentos, a través de un software y una página web, registrando a las ONGs, a los donantes e incluso a los receptores. La idea surgió tras la alerta de que Brasil había vuelto al mapa del hambre y después de que una investigación demostrara que más de 116 millones de brasileños forman parte de la población que tiene problemas en cuanto a la garantía de la alimentación por la constitución de 1988.

Palabras clave: Inseguridad. Comida. Hambre. Alimentación. Donación

LISTA DE ABREVIATURAS / SIGLAS

CRAS – Centro de Referência de Assistência Social

IBGE - Instituto Brasileiro de Geografia e Estatística

ONG - Organização Não Governamental

ONU - Organização das Nações Unidas

Rede Penssan - Rede Brasileira de Pesquisa em Soberania e Segurança Alimentar e Nutricional

SUMÁRIO

1 INTRODUÇÃO	14
1.1 Contexto de produção do software	14
1.2 Problematização	14
1.3 Objetivos	14
1.3.1 Objetivo geral	14
1.3.2 Objetivos específicos	15
1.4 Metodologia	15
1.4.1 Tipos de pesquisa	15
1.4.2 Procedimento para determinação da proposta do aplicativo	15
1.4.3 Procedimento para a coleta de dados sobre a microrregião	16
2 DESENVOLVIMENTO	17
2.1 Referencial teórico	17
2.2 Diagrama canvas	21
2.3 Levantamento de requisitos	22
2.3.1 Requisitos funcionais do aplicativo	22
2.3.2 Requisitos não funcionais do aplicativo	24
2.3.3 Requisitos funcionais do site	25
2.3.4 Requisitos não funcionais do site	26
2.4 Diagramas	27
2.4.1 Diagrama de classe do aplicativo	27
2.4.2 Diagrama de caso de uso do aplicativo	28
2.4.3 Diagrama Entidade Relacionamento (DER) do aplicativo	29
2.4.4 Diagrama de caso de uso do site	30
2.5 Ciclo de vida do software	31
2.6 Softwares utilizados	32
2.6.1 Figma	32
2.6.2 Visual Studio Code	32
2.6.3 Trello	33

2.6.4 Google Forms	33
2.6.5 Expo	33
2.6.6 Canva	33
2.6.7 Draw.io	33
2.6.8 Firebase	34
2.7 Telas e códigos	34
2.7.1 Telas Iniciais (Sliders) do aplicativo	34
2.7.2 Tela Login/Cadastro do aplicativo	35
2.7.3 Tela Notícias/Menu do aplicativo	37
2.7.4 Tela Doadores do aplicativo	38
2.7.5 Tela Ajude Alguém do aplicativo	39
2.7.6 Tela ONGs do aplicativo	40
2.7.7 Tela Configurações/Edição de perfil do aplicativo	41
2.7.8 Código da tela Cadastro do aplicativo	42
2.7.9 Código da tela Login do aplicativo	46
2.7.10 Código da tela Notícias do aplicativo	47
2.7.11 Código da tela ONGs do aplicativo	48
2.7.12 Código da tela Doadores/Ajude Alguém do aplicativo	50
2.7.13 Código da tela Configurações do aplicativo	52
2.7.14 Código de controle da navegação e dos sliders do aplicativo	53
2.7.15 Página Inicial do site	57
2.7.16 Menu do site	58
2.7.17 Página Notícias do site	59
2.7.18 Página ONGs do site	61
2.7.19 Página Sobre nós do site	62
2.7.20 Página Dúvidas Frequentes do site	64
2.7.21 Página Formulário de contato do site	65
2.7.22 Página Contribua do site	66
2.7.23 Página Formulário de registro do site	68

2.7.24	Página Termos de uso	69
2.7.25	Código da página Inicial do site	72
2.7.26	Código da página Notícias do site	72
2.7.27	Código da página ONGs do site	73
2.7.28	Código da página Sobre Nós do site	74
2.7.29	Código da página Dúvidas Frequentes do site	74
2.7.30	Código da página Contribua do site	75
2.7.31	Código da página Termos de uso do site	75
2.7.32	Código da página Formulário de contato do site	76
2.7.33	Código da página Formulário de registro do site	76
2.8	Rotinas de testes do aplicativo	77
2.8.1	Testes realizados	77
2.8.2	Tabelas de testes da tela de Sliders	77
2.8.3	Tabela de testes da tela Inicial de Login/Cadastro	79
2.8.4	Tabelas de testes da tela Login	80
2.8.5	Tabelas de testes da tela Cadastro	83
2.8.6	Tabelas de testes da tela Notícias	89
2.8.7	Tabelas de testes da tela Perfil	94
2.8.8	Tabelas de testes da tela Doadores	97
2.8.9	Tabelas de testes da tela Ajude Alguém	100
2.8.10	Tabelas de testes da tela ONGs	103
2.8.11	Tabelas de testes da tela Configurações	106
2.9	Principais mudanças realizadas após a rotina de teste do aplicativo	111
2.10	Rotinas de testes do site	111
2.10.1	Testes escolhidos	111
2.10.2	Tabelas de teste da tela Inicial	112
2.10.3	Tabelas de testes da tela Notícias	114
2.10.4	Tabela de testes da tela Sobre Nós	117
2.10.5	Tabela de testes da tela Formulário de Dúvidas	118

2.10.6 Tabela de teste da tela Contribua	119
2.10.7 Tabela de testes da tela Registro de ONGs e Empresas	120
2.10.8 Tabela de testes da tela ONGs	121
2.11 Principais mudanças realizadas após a rotina de teste do site	122
2.12 Diferenciais do projeto	122
3 DISCUSSÃO DOS RESULTADOS	125
3.1 Resultados das pesquisas bibliográficas	125
3.2 Resultados das pesquisas de campo	125
3.3 Resultados da pesquisa na microrregião	128
4 CONSIDERAÇÕES FINAIS	128
REFERÊNCIAS	130

1 INTRODUÇÃO

1.1 Contexto de produção do software

O *software* foi desenvolvido visando alterar determinados problemas em relação ao âmbito social ligado à alimentação da população residente nos bairros atendidos pelo CRAS Trianon, sendo aqueles que apresentam mais vulnerabilidade em Taboão da Serra. O projeto foi elaborado com a finalidade de auxiliar na comunicação e facilitar doações de alimentos a esse público por meio do aplicativo GreenPoint, complementado por um *website*.

1.2 Problematização

Como ajudar a combater os índices de fome e incentivar o desenvolvimento sustentável no município de Taboão da Serra, em específico, o território atendido pelo Centro de Referência de Assistência Social (CRAS) Trianon, que apresenta um maior número de pessoas que enfrentam a pobreza e não recebem o Auxílio Brasil?

1.3 Objetivos

1.3.1 Objetivo geral

O objetivo geral do aplicativo em questão é divulgar e facilitar a comunicação entre instituições de distribuição de comida, possíveis doadores e pessoas que necessitam dessa assistência, a fim de diminuir o número de pessoas em situação de insegurança alimentar no município de Taboão da Serra.

1.3.2 Objetivos específicos

- Promover conexão entre possíveis fornecedores (doadores) de alimentos e consumidores que necessitem;
- Incentivar o desenvolvimento sustentável, através de notícias e informações sobre alimentação, agricultura, dados sobre pessoas em vulnerabilidade alimentar, entre outras;
- Diminuir o desperdício de comida, incentivando os usuários a doarem os excedentes para instituições e pessoas carentes.

1.4 Metodologia

1.4.1 Tipos de pesquisa

Os tipos de pesquisas se relacionam com os fins a que se destinam e aos meios de investigação. Sendo assim, a seguinte pesquisa enquadra-se como qualitativa e quantitativa quanto à abordagem e bibliográfica, de levantamento e de campo quanto aos procedimentos. Qualitativa porque expõe os dados possibilitando análise da região escolhida e quantitativa porque contabiliza o número de pessoas em determinadas situações pertinentes à escolha da microrregião. Bibliográfica, pois foram feitas pesquisas em sites e outras fontes, de levantamento, pois foi realizado questionamento direto com os integrantes do projeto, após isso, foi feita uma análise quantitativa dos dados obtidos e, por fim, de campo através dos formulários enviados ao público.

1.4.2 Procedimento para determinação da proposta do aplicativo

Inicialmente, os componentes do grupo pesquisaram sobre o assunto e como ele poderia ser aplicado na macrorregião de Taboão da Serra, assim, foi feito um diálogo com pessoas que tinham mais conhecimento acerca do tema. A equipe se reuniu para debater sobre os conhecimentos apurados e sobre as necessidades

dentro desse assunto. Dessa maneira, para comprovar a viabilidade, pertinência e relevância do projeto, foi elaborado um formulário digital, pelo Google Forms, onde as perguntas foram focadas em descobrir o bairro mais carente da macrorregião, a idade daqueles que responderam e a opinião do público quanto à ideia do grupo, que comprovou a necessidade da proposta através dos dados recebidos.

Para a realização do projeto, foram buscadas ideias inovadoras que sejam acessíveis para todas as classes da sociedade. O grupo procurou ser o mais original possível, a partir da pesquisa de aplicativos com propostas parecidas, unindo diversas funcionalidades que contribuem para a finalidade proposta, como a promoção de possíveis doadores, informações sobre o desenvolvimento sustentável e, conseqüentemente, suporte para pessoas necessitadas através de organizações sem fins lucrativos (ONGs). Desse modo, o aplicativo possibilita assistência a pessoas necessitadas e desfavorecidas economicamente por meio de doadores comovidos com o projeto.

1.4.3 Procedimento para a coleta de dados sobre a microrregião

Dessa forma, a proposta de como seria o aplicativo já estava pronta, restando somente delimitar a microrregião a ser atendida. Para isso, o grupo entrou em contato com o Centro de Referência de Assistência Social (CRAS) de Taboão da Serra para obter dados acerca dos oito CRAS existentes no município e compará-los, para assim decidir qual microrregião é mais viável, pertinente e relevante à implementação do aplicativo.

2 DESENVOLVIMENTO

2.1 Referencial teórico

Atualmente, a fome, problema que atinge um número eminente de pessoas no Brasil e no mundo, provém da falta de alimentos. Esse número é extremamente elevado, tendo em vista a extensão territorial do país que apresenta grande potencial agrícola e que, em teoria, poderia atender às necessidades dos indivíduos brasileiros, porém, grande parte do dinheiro no país está nas mãos de somente 1% da população, assim, dificultando o pleno acesso a uma alimentação saudável para a maioria dos habitantes. Mesmo com programas sociais federais e estaduais, o problema da fome não é solucionado, pois se faz presente em pequenas, médias e grandes cidades e até mesmo em suas áreas rurais, independentemente do estado brasileiro. A solução para a questão parece distante e envolve uma série de fatores estruturais que estão impregnados na sociedade brasileira há anos.

Figura 1 – Insegurança alimentar no Brasil



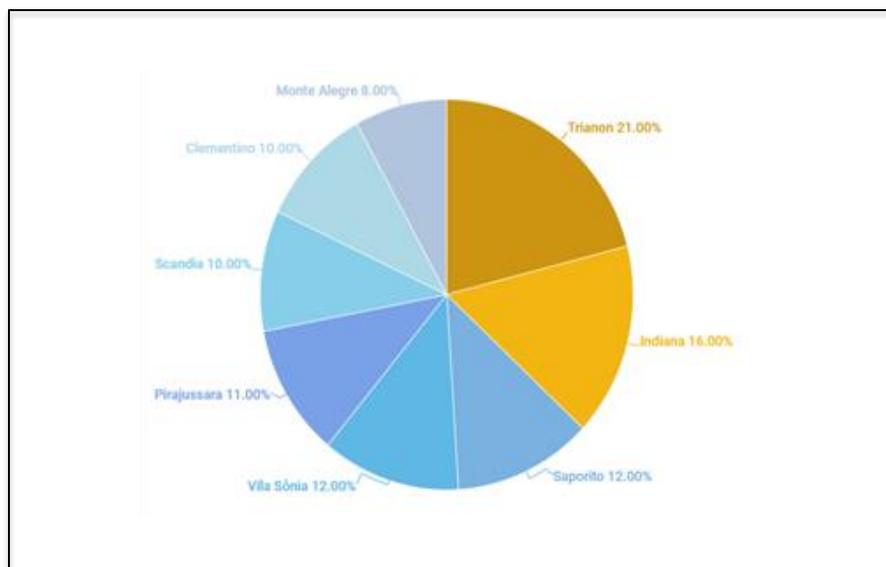
Fonte: IBGE, Pesquisa “Alimento para justiça”, 2020

A partir do ano de 2015, quando os países signatários da ONU desenvolveram os Objetivos de Desenvolvimento Sustentável (ODS), os números da fome sofreram

uma queda em escala global. O rápido crescimento econômico e o desenvolvimento da agricultura nesses países ajudaram a reduzir pela metade a proporção de pessoas subnutridas no mundo. Porém, no Brasil, desde o mesmo ano em que o foi criado, houve um regresso nesse objetivo.

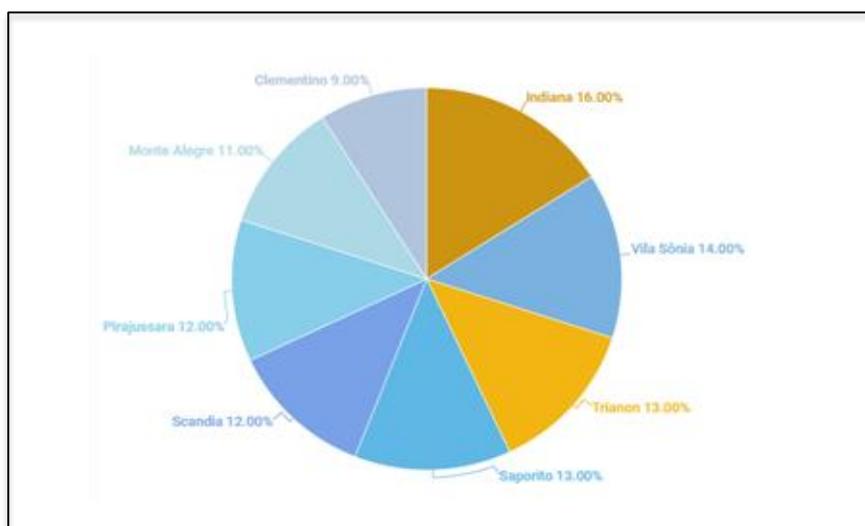
Segundo pesquisas da Rede Penssan, em 2021, 116,8 milhões de brasileiros convivem com algum grau de insegurança alimentar e não têm acesso pleno e permanente a alimentos e 19 milhões de brasileiros enfrentavam a fome, o equivalente a 9% da população do país. Outro levantamento divulgado, também pela Rede Penssan, em 2022, indica que 33,1 milhões de pessoas não têm o que comer no país. Na região Sudeste, cerca de 9,37 milhões de famílias estão em situação de insegurança alimentar, de acordo com uma pesquisa do IBGE. Quando se fala na região de São Paulo, estima-se que entre 6 e 7 milhões pessoas passem por insegurança alimentar grave, segundo pesquisas da Rede Penssan. Já em Taboão da Serra, mais de 16 mil famílias têm problemas de carência alimentar, informa o Centro de Triagem da Secretaria de Assistência Social e Cidadania de Taboão da Serra. Além disso, dados da Prefeitura de Taboão da Serra, mostram que todos os bairros atendidos pelo Centro de Referência de Assistência Social (CRAS) – órgão responsável por coordenar as atividades assistenciais nos bairros – de Taboão da Serra possuem 18.357 famílias que estão enfrentando a linha da pobreza ou extrema pobreza. Além do mais, o território que apresenta uma maior vulnerabilidade social no município, é o atendido pelo CRAS Trianon, no qual fazem parte os seguintes bairros: Jd. Freitas Junior, Jd. Pirajussara, Jd. Record, Jd. Roberto, Sítio das Madres, Jd. Trianon e Vila Mafalda. Os dados também apresentam a faixa de renda das famílias no território: de 2.686 famílias atendidas pelo CRAS, 641 estão enfrentando a linha da pobreza e 2.045 estão enfrentando a linha de extrema pobreza. Dessas famílias, menos da metade recebem o Auxílio Brasil, sendo assim, não é o suficiente para manter os recursos necessários para viver, especialmente, a alimentação.

Figura 2 – Índice de pobreza nos CRAS de Taboão da Serra



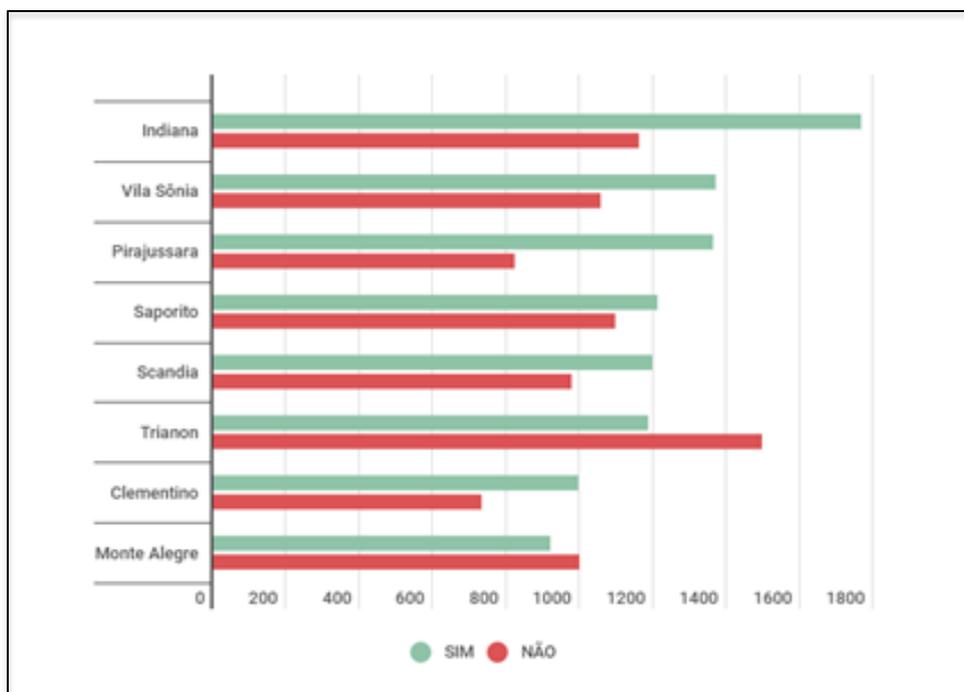
Fonte: Centro de Referência de Assistência Social de Taboão da Serra, 2022

Figura 3 – Índice de extrema pobreza nos CRAS de Taboão da Serra



Fonte: Centro de Referência de Assistência Social de Taboão da Serra, 2022

Figura 4 – Índice de quantas pessoas recebem ou não o Auxílio Brasil em cada CRAS de Taboão da Serra

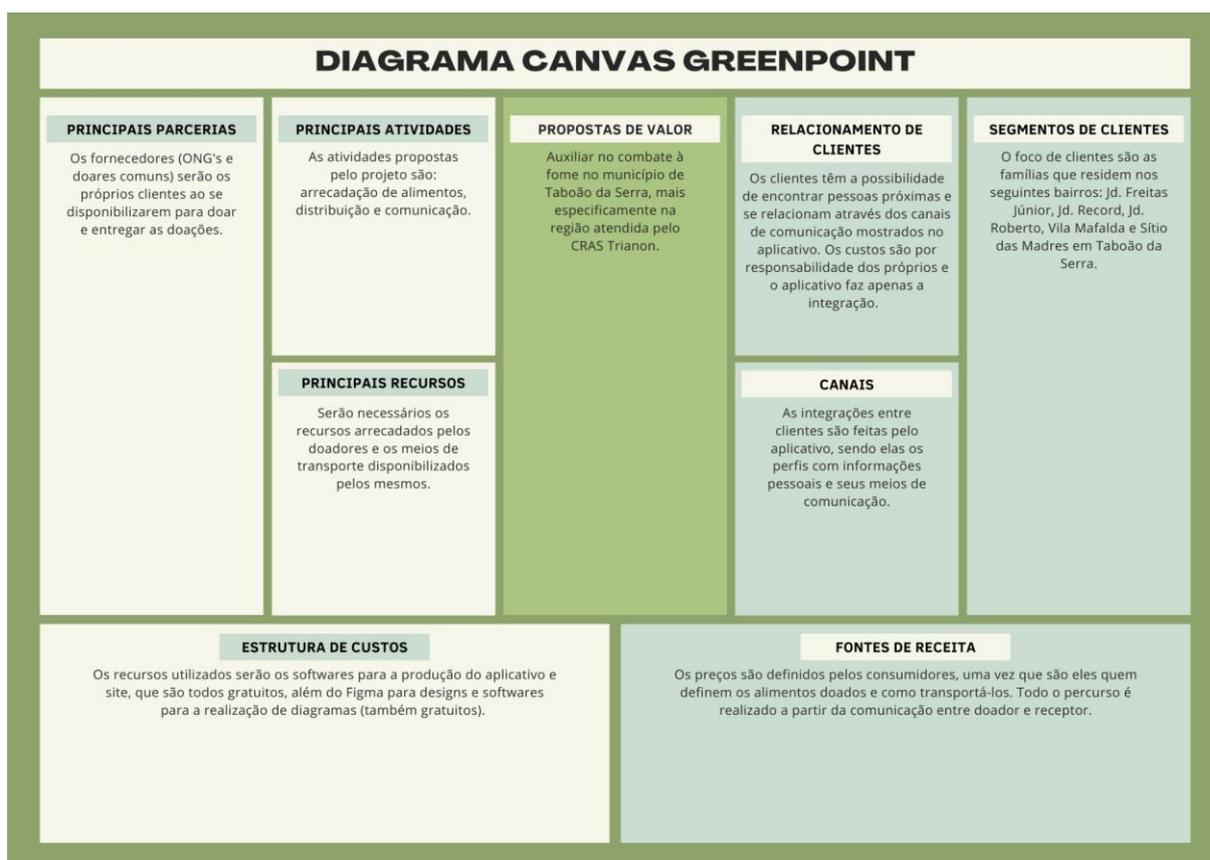


Fonte: Centro de Referência de Assistência Social de Taboão da Serra, 2022

2.2 Diagrama canvas

O Modelo de Negócios Canvas é uma representação gráfica que permite visualizar, de forma compacta, apenas as principais características do modelo de um negócio. O objetivo do Diagrama Canvas é permitir que o desenvolvedor crie um modelo de negócios que possa servir como um guia de hipóteses a serem validadas.

Figura 5 – Diagrama Canvas



Fonte: Autoria própria, 2022

2.3 Levantamento de requisitos

2.3.1 Requisitos funcionais do aplicativo

Os requisitos funcionais são as necessidades que devem ser atendidas pelo software por meio dos serviços que ele propõe. No caso do GreenPoint, são eles:

Código	Identificação	Categoria	Ator	Objetivo
[RF01]	Internet	Essencial	Sistema	A utilização do <i>software</i> depende de conexão com internet
[RF02]	Visualizar instruções	Importante	Usuário	O <i>software</i> deve permitir a visualização de instruções iniciais
[RF03]	Efetuar cadastro	Essencial	Usuário	O <i>software</i> deve permitir o cadastro de usuários doadores ou necessitados
[RF04]	Efetuar login	Essencial	Usuário	O <i>software</i> deve permitir o login de usuários doadores ou necessitados
[RF05]	Possibilitar navegação	Essencial	Sistema	O <i>software</i> deve apresentar a opção de navegar entre as telas
[RF06]	Visualizar notícias	Importante	Usuário	O <i>software</i> deve permitir que o usuário acesse a tela de notícias
[RF07]	Buscar notícias	Importante	Usuário	O <i>software</i> deve permitir a busca de notícias

[RF08]	Visualizar a área de doadores	Essencial	Usuário	O <i>software</i> deve permitir a visualização de usuários doadores disponíveis
[RF09]	Permitir a alteração de status do doador	Importante	Usuário doador	O <i>software</i> deve permitir que o usuário doador altere o seu status (ativo/desativo)
[RF10]	Alterar dados da conta	Essencial	Usuário doador, e usuário auxiliado	O <i>software</i> e deve permitir que os usuários efetuem a manutenção de suas contas (consulta, inclusão, exclusão e alteração);
[RF11]	Visualizar configurações	Essencial	Sistema	O <i>software</i> deve apresentar uma tela de configurações
[RF12]	Possibilitar o contato com desenvolvedores	Importante	Sistema	O <i>software</i> deve apresentar opções externas para contato com desenvolvedores
[RF13]	Fornecer segurança	Essencial	Sistema	O <i>software</i> deve garantir a proteção dos dados, e só devem ser acessados com autorização
[RNF05]	Fornecer compatibilidade Android	Essencial	Sistema	O <i>software</i> deve fornecer compatibilidade para o sistema Android

2.3.2 Requisitos não funcionais do aplicativo

Os requisitos não funcionais são aqueles que não estão ligados à funcionalidade do sistema de forma direta, são funções complementares. Sem esse tipo de requisito, o *software* funcionará, porém, pode o deixar com baixa qualidade. Sendo assim, os requisitos não funcionais do aplicativo GreenPoint são:

Código	Identificação	Categoria	Ator	Objetivo
[RNF01]	Alterar visualização do aplicativo	Não necessário	Sistema	O <i>software</i> deve possibilitar a alteração de visualização do aplicativo (tema de cores, áudio descrição)
[RNF02]	Possibilitar adaptações futuras	Não necessário	Sistema	O <i>software</i> deve ser desenvolvido de modo que facilite as futuras adaptações
[RNF03]	Fornecer linguagem simples	Não necessário	Sistema	O <i>software</i> deve fornecer uma linguagem de fácil entendimento e compreensão
[RNF04]	Fornecer gratuidade	Não necessário	Sistema	O <i>software</i> deve fornecer todas as funcionalidade de forma gratuita

Fonte: Autoria própria, 2022

2.3.3 Requisitos funcionais do site

Código	Identificação	Categoria	Ator	Objetivo
[RF01]	Internet	Essencial	Sistema	A utilização do <i>software</i> depende de conexão com internet
[RF02]	Possibilitar navegação	Essencial	Sistema	O <i>software</i> deve apresentar a opção de navegar entre as telas
[RF03]	Visualizar notícias	Importante	Usuário	O <i>software</i> deve permitir que o usuário acesse a tela de notícias
[RF04]	Buscar notícias	Importante	Usuário	O <i>software</i> deve permitir a busca de notícias
[RF05]	Visualizar configurações	Essencial	Sistema	O <i>software</i> deve apresentar uma tela de configurações
[RF06]	Possibilitar o contato com desenvolvedores	Importante	Sistema	O <i>software</i> deve apresentar opções externas para contato com desenvolvedores
[RNF04]	Fornecer gratuidade	Essencial	Sistema	O <i>software</i> deve fornecer todas as funcionalidade de forma gratuita
[RNF05]	Fornecer compatibilidade	Essencial	Sistema	O <i>software</i> deve fornecer compatibilidade para os principais navegadores (Google, Firefox, Opera, Edge)

2.3.4 Requisitos não funcionais do site

Código	Identificação	Categoria	Ator	Objetivo
[RNF01]	Alterar visualização do aplicativo	Não necessário	Sistema	O <i>software</i> deve possibilitar a alteração de visualização do aplicativo (tema de cores, audiodescrição)
[RNF02]	Possibilitar adaptações futuras	Não necessário	Sistema	O <i>software</i> deve ser desenvolvido de modo que facilite as futuras adaptações
[RNF03]	Fornecer linguagem simples	Não necessário	Sistema	O <i>software</i> deve fornecer uma linguagem de fácil entendimento e compreensão

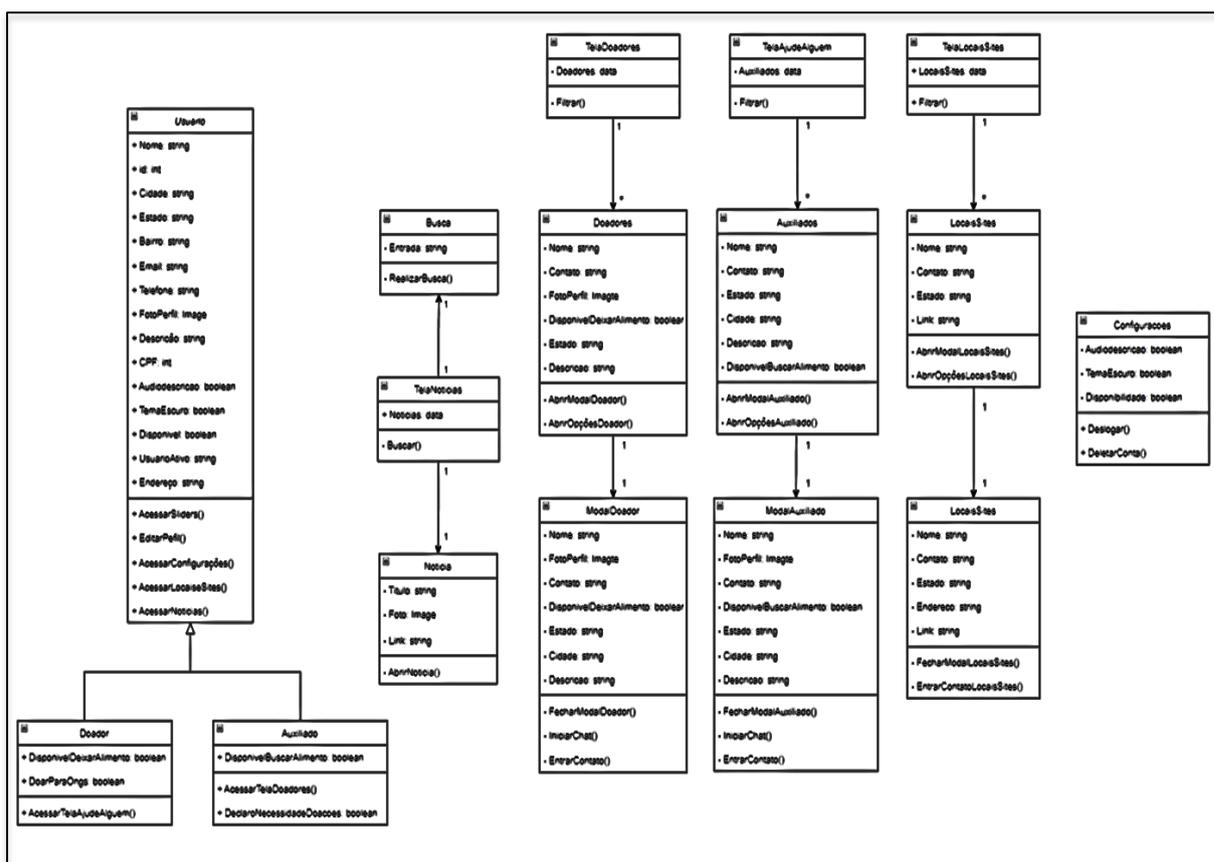
Fonte: Autoria própria, 2022

2.4 Diagramas

2.4.1 Diagrama de classe do aplicativo

Os diagramas de classe são utilizados para modelar os objetos que compõem o *software*, para exibir os relacionamentos entre os objetos e descrever qual a função desses objetos e os serviços que eles fornecem.

Figura 6 – Diagrama de Classe do Aplicativo GreenPoint

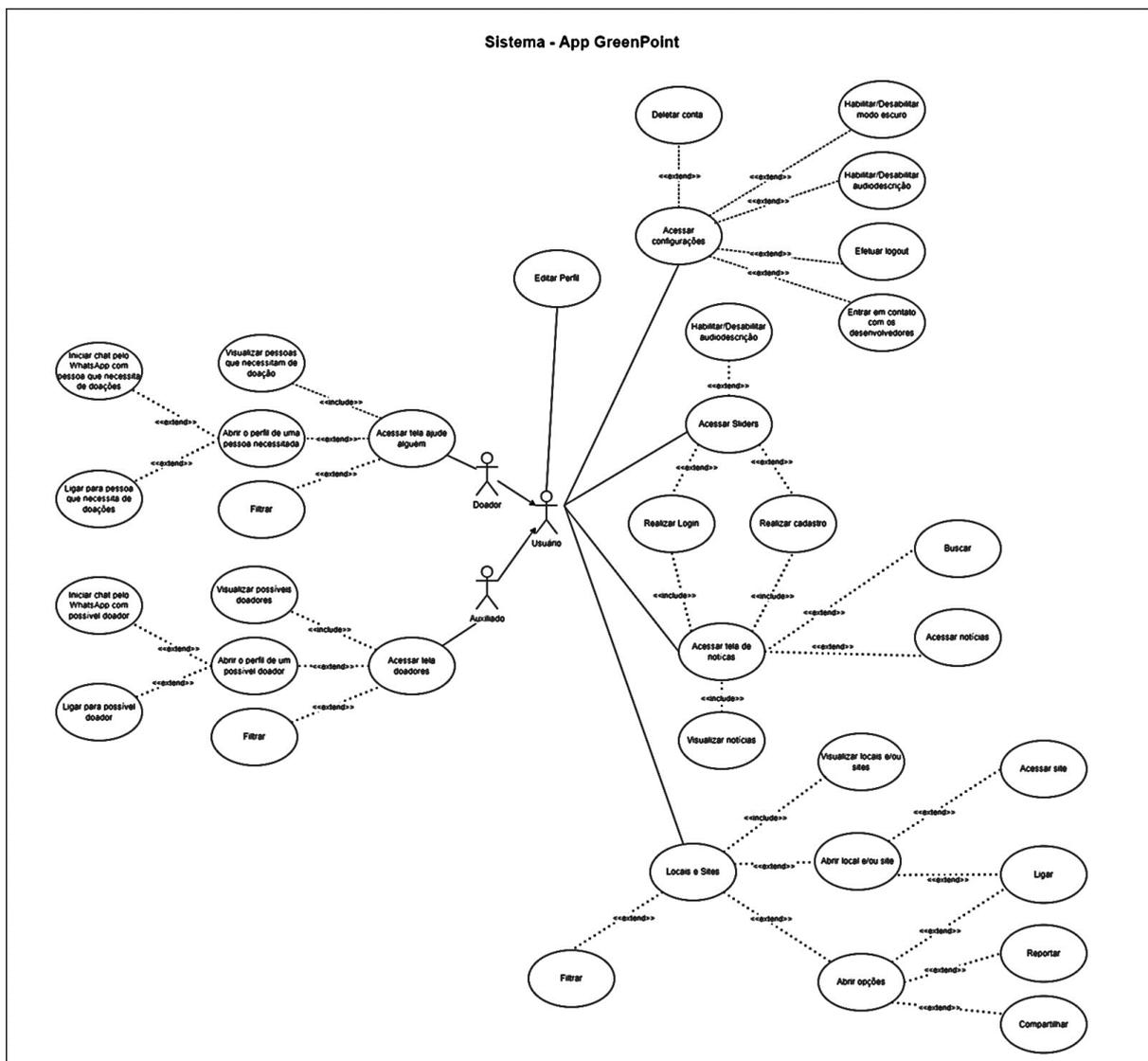


Fonte: Autoria própria, 2022

2.4.2 Diagrama de caso de uso do aplicativo

O diagrama de caso de uso descreve como os usuários interagem com o sistema. Os elementos de modelo mais importantes são: casos de uso, atores e as relações entre eles.

Figura 7 – Diagrama de Caso de Uso do Aplicativo GreenPoint

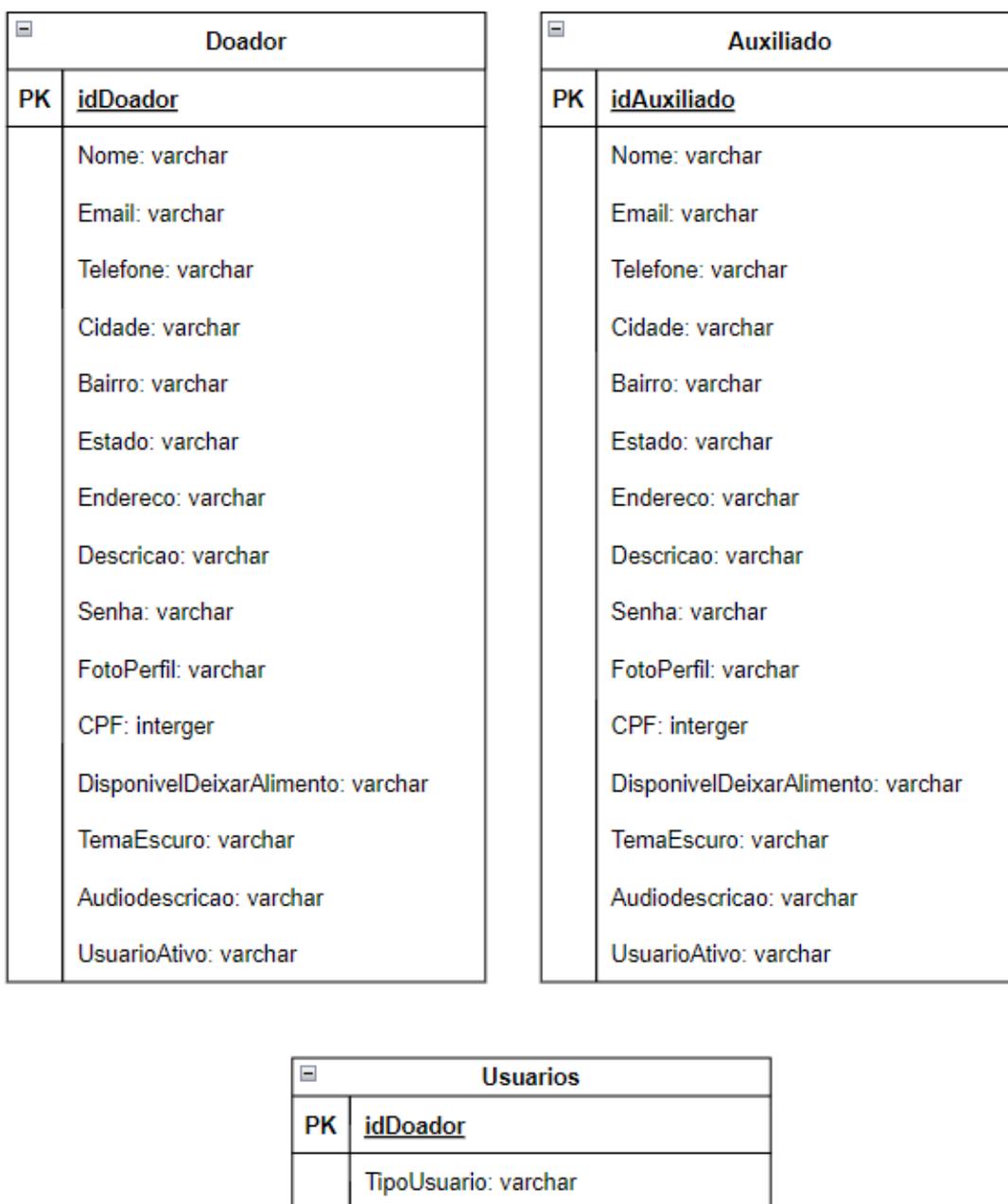


Fonte: Autoria própria, 2022

2.4.3 Diagrama Entidade Relacionamento (DER) do aplicativo

O Diagrama Entidade Relacionamento é a representação gráfica do modelo conceitual. O diagrama facilita a comunicação entre os integrantes da equipe, pois unifica a linguagem entre o responsável por levantar os requisitos e os desenvolvedores.

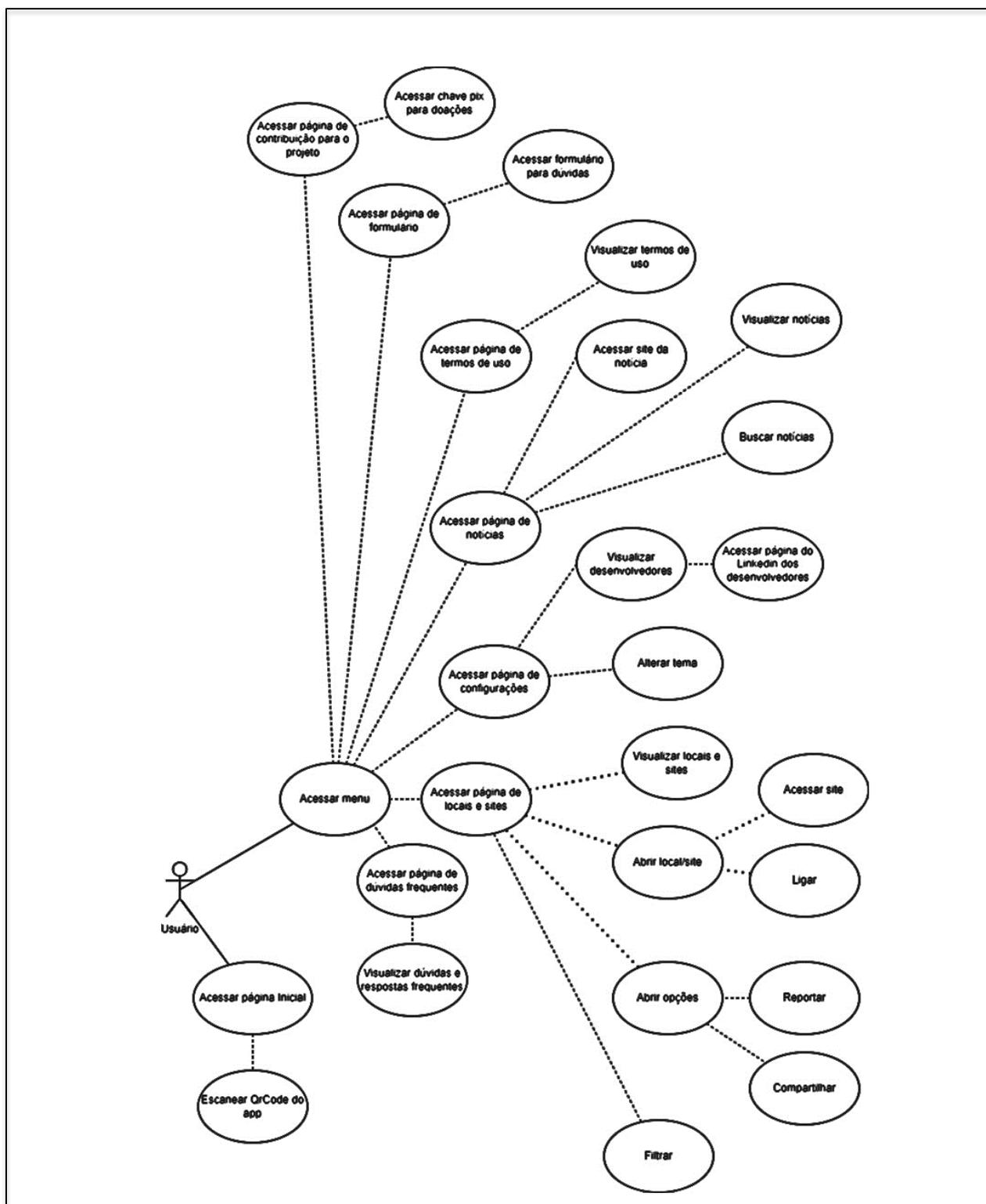
Figura 8 – Diagrama Entidade Relacionamento do Aplicativo GreenPoint



Fonte: Autoria própria, 2022

2.4.4 Diagrama de caso de uso do site

Figura 9 – Diagrama de Caso de Uso do Site GreenPoint



Fonte: Autoria própria, 2022

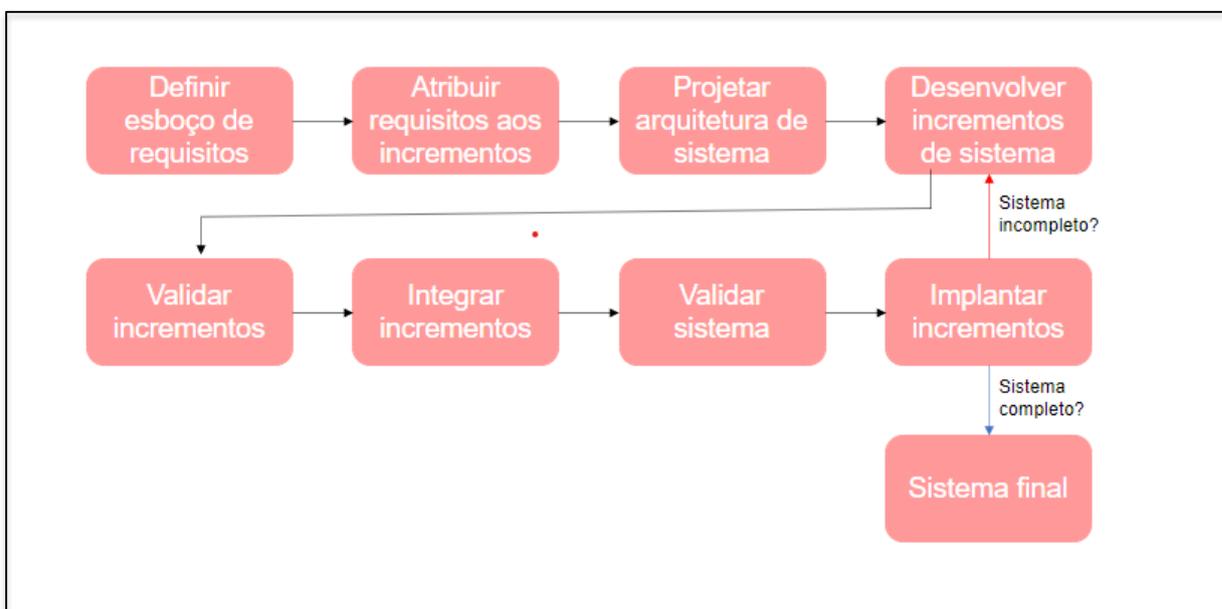
2.5 Ciclo de vida do software

O ciclo de vida de um *software* é uma estrutura que expõe os processos e as atividades realizadas no desenvolvimento e manutenção de um *software*, indicando toda a vida do mesmo. Neste ciclo, existem modelos que definem como o software será desenvolvido, lançado, aprimorado e finalizado.

O modelo utilizado no projeto foi o modelo incremental, já que é um modelo que não exige conexão prévia com os consumidores.

O incremento 1 é a parte que apresenta a visão dos programadores a partir do desenvolvimento do aplicativo.

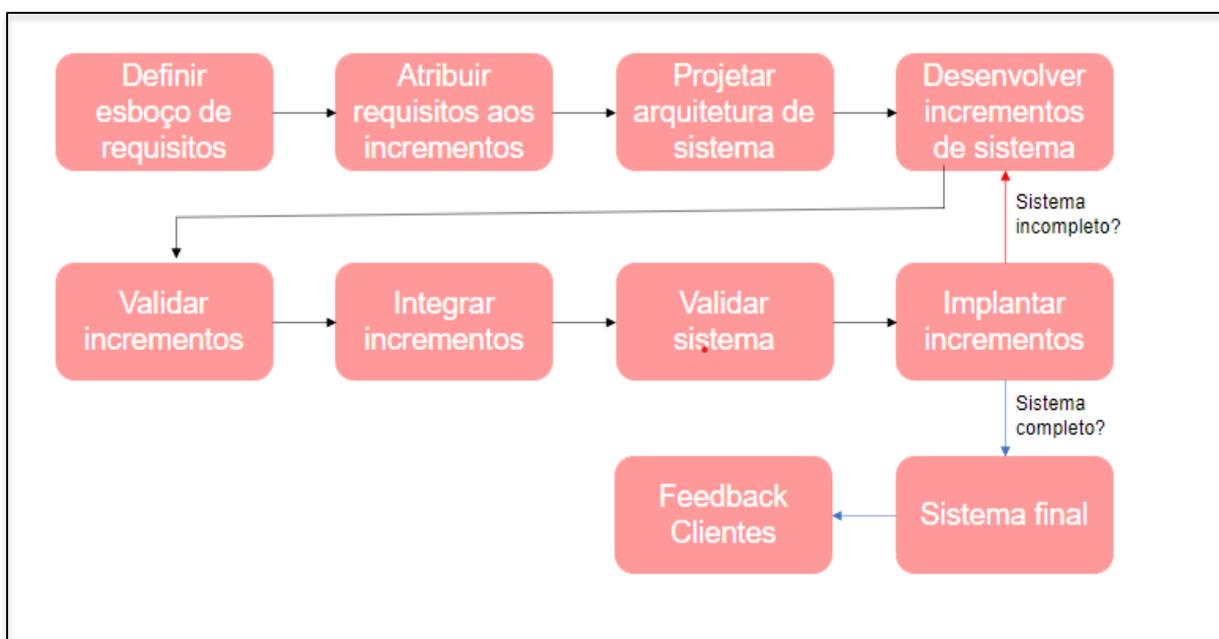
Figura 10 – Diagrama de Ciclo de Vida - Incremento 1



Fonte: Autoria própria, 2022

Após o incremento 1, com a validação do projeto pela banca de avaliadores, há a realização do incremento 2, no qual são adicionados os apontamentos feitos pelos espectadores junto ao *feedback* dos clientes.

Figura 11 – Diagrama de Ciclo de Vida - Incremento 2



Fonte: Autoria própria, 2022

2.6 Softwares utilizados

2.6.1 Figma

O Figma é um *software* de design digital e utilizado no projeto a fim de desenvolver o design das telas do aplicativo, assim como o logotipo e todas as marcas d'água.

2.6.2 Visual Studio Code

O Visual Studio Code é um editor de código-fonte criado pela Microsoft que é utilizado para desenvolver todo o código do aplicativo, com linguagem Javascript juntamente à biblioteca React Native, e do *website*, com linguagem HTML, Javascript e CSS.

2.6.3 Trello

O Trello é um *software* dedicado à organização, onde foram inseridas as atividades feitas, a fazer e em andamento. Foi utilizado para garantir que todas as partes do projeto seriam realizadas de acordo com o tempo estipulado.

2.6.4 Google Forms

O Google Forms é um aplicativo de gerenciamento de pesquisas produzido pelo Google e pode ser utilizado para pesquisar e coletar informações sobre outras pessoas e para questionários e formulários de registro. No projeto, a utilização desse *software* foi importante para a coleta de informações visando confirmar a relevância, viabilidade e pertinência do mesmo.

2.6.5 Expo

O Expo é um *software* utilizado para o desenvolvimento mobile com React Native que permite o fácil acesso às APIs nativas do dispositivo sem a necessidade de instalar dependências ou alterar o código nativo. O aplicativo GreenPoint foi desenvolvido através do Expo dentro do Visual Studio Code.

2.6.6 Canva

O Canva é uma plataforma de design e utilizado para desenvolver o Diagrama Canvas com o modelo de negócios do GreenPoint.

2.6.7 Draw.io

O Draw.io é um *software* para desenvolvimento de diagramas e utilizado para a realização dos principais diagramas, tanto do *website*, quanto do aplicativo.

2.6.8 Firebase

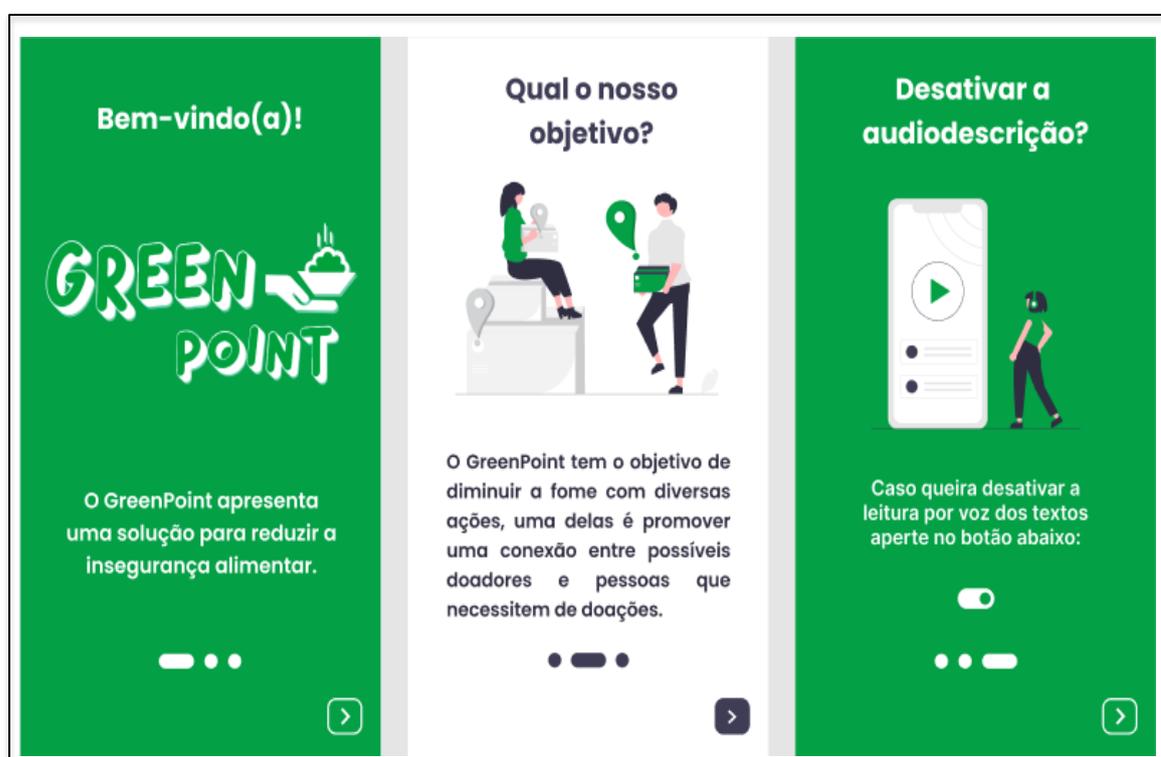
O Firebase é um banco de dados NoSQL que é hospedado na nuvem. Ele é capaz de armazenar e sincronizar dados entre os usuários em tempo real. Foi utilizado para a produção do banco de dados do aplicativo e do *website*.

2.7 Telas e códigos

2.7.1 Telas Iniciais (Sliders) do aplicativo

As telas iniciais (sliders) são aquelas mostradas quando o usuário tem seu primeiro contato com o aplicativo. Elas mostram o intuito do projeto e, além disso, há a opção de ativar ou desativar a audiodescrição durante o uso.

Figura 12 – Telas Iniciais (sliders) do Aplicativo



Fonte: Autoria própria, 2022

2.7.2 Tela Login/Cadastro do aplicativo

As telas de login/cadastro são as telas na qual o usuário será direcionado para realizar login no aplicativo ou seu cadastro nele. Se o usuário já tiver uma conta, o botão "Entrar" o direciona à tela de login, na qual deverão ser preenchidos os campos de e-mail e senha. Caso o usuário esteja entrando pela primeira vez no aplicativo, ele terá que clicar no botão "Cadastrar-se". Terão que ser preenchidos os campos do nome, e-mail, CPF, telefone, senha e confirme sua senha.

Figura 13 – Login/Cadastro do Aplicativo (1)

The figure displays three sequential screens of the application's login and registration process:

- Screen 1:** A hand interacts with a smartphone. Below, a text box asks "Você é novo(a) por aqui? Cadastre-se e nos ajude a mudar a realidade de quem precisa!". There are two buttons: a green "Entrar" button and a white "Cadastrar-se" button.
- Screen 2:** Titled "Bem-vindo(a) de volta!". It features a woman holding a large leaf. Below are input fields for "Email:" (placeholder: "Insira o seu Email") and "Senha:" (placeholder: "Insira o sua senha"). A link "Esqueceu sua senha?" is next to the password field. A green "Entrar" button is at the bottom.
- Screen 3:** Titled "Dados". It has a progress indicator with steps 1, 2, and 3, where step 1 is active. Below is a profile picture placeholder with a camera icon. Input fields are provided for "Nome:" (placeholder: "Insira o seu nome completo"), "Email:" (placeholder: "Insira o seu Email"), "CPF:" (placeholder: "Insira o seu cpf"), "Telefone:" (placeholder: "Ex: (xx) xxxxx-xxxx"), "Senha:" (placeholder: "Insira o sua senha"), and "Confirme sua senha:" (placeholder: "Insira o sua senha novamente"). A right arrow button is at the bottom right.

Fonte: Autoria própria, 2022

Após o preenchimento dos outros dados, o usuário deverá preencher os campos de sua localização, no caso, CEP. Automaticamente aparecerá o resto de seu endereço, com estado, cidade, bairro, número e rua. No final o usuário deverá colocar na opção que desejar entre "doar" ou "receber doações". E por fim, aceitar os termos de uso.

Se o usuário optar por receber doações, ele deverá informar sua renda familiar mensal e o número de dependentes dessa renda. Ele poderá colocar ou não a opção de buscar as doações em um local próximo. Por fim, deverá informar se é maior de idade. Assim, seu cadastro poderá ser realizado com sucesso. Caso o usuário opte por ser doador, ele deverá escolher se levaria ou não doações em um local próximo, e, também, se ele é maior de idade. Ao realizar esses passos, seu cadastro será concluído.

Figura 14 – Login/Cadastro do Aplicativo (2)

The figure displays three sequential screens of the application's registration process:

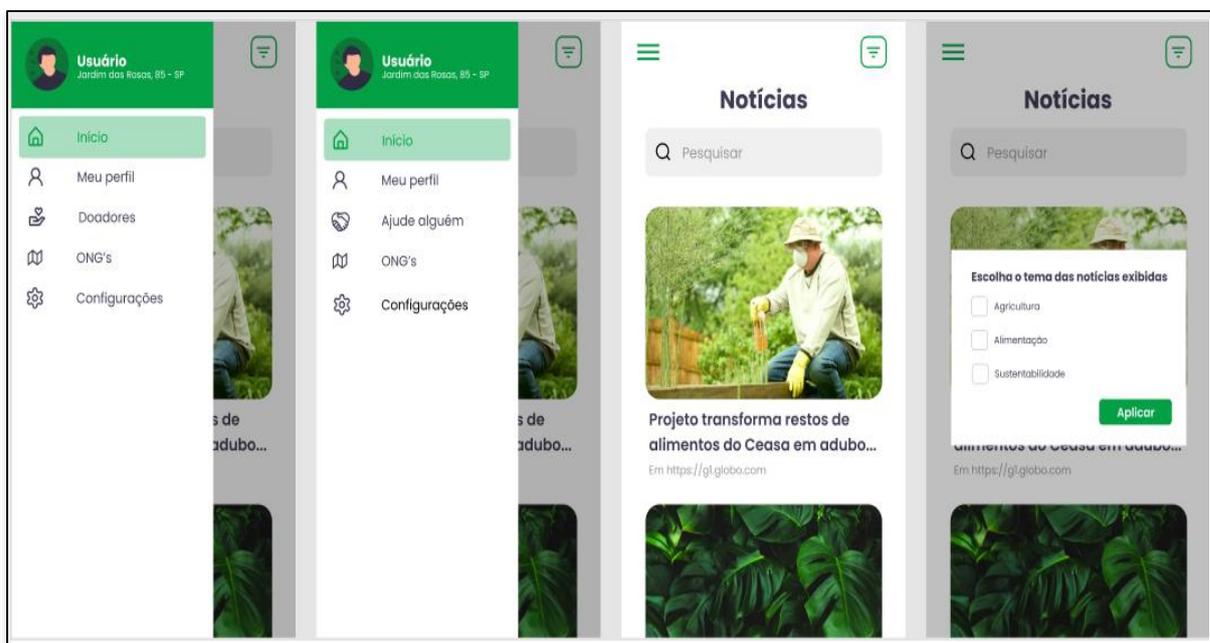
- Localização (Location):** This screen features a progress indicator with three steps (1, 2, 3), where step 1 is active. It includes input fields for CEP, Estado, Cidade, Bairro, Número, and Endereço. At the bottom, there are radio buttons for "O que você deseja?" (Doar or Receber doações) and a checkbox for "Aceitar nossos Termos de Uso".
- Finalização (Finalization) - Donor:** This screen shows the second step of the process. It includes input fields for "Renda familiar mensal (R\$)" and "Número de dependentes dessa renda". It also has a checkbox for "Eu me disponibilizo a buscar alimentos próximos a mim" and radio buttons for "Você tem mais de 18 anos?". A green "Cadastrar-se" button is at the bottom.
- Finalização (Finalization) - Recipient:** This screen shows the third step. It includes radio buttons for "Você se disponibiliza a levar alimentos próximo ao seu local?" and "Você tem mais de 18 anos?". A green "Cadastrar-se" button is at the bottom.

Fonte: Autoria própria, 2022

2.7.3 Tela Notícias/Menu do aplicativo

A primeira tela que o usuário será direcionado quando entrar no aplicativo, é a tela de notícias. Nela haverá notícias sobre agricultura, alimentação e sustentabilidade. Na barra de pesquisa poderão ser escritas palavras-chave para achar determinadas notícias e, também, terá um filtro, no qual a pessoa poderá escolher os tipos de notícias que quer ver. No canto superior esquerdo há o "Menu", que terá todas as opções pelas quais o usuário pode navegar pelo aplicativo: "Início", "Meu perfil", "Doadores" (caso o usuário seja alguém que precise de ajuda), "Ajude alguém" (caso o usuário seja doador), "ONGs" e "Configurações".

Figura 15 – Notícias/Menu do Aplicativo

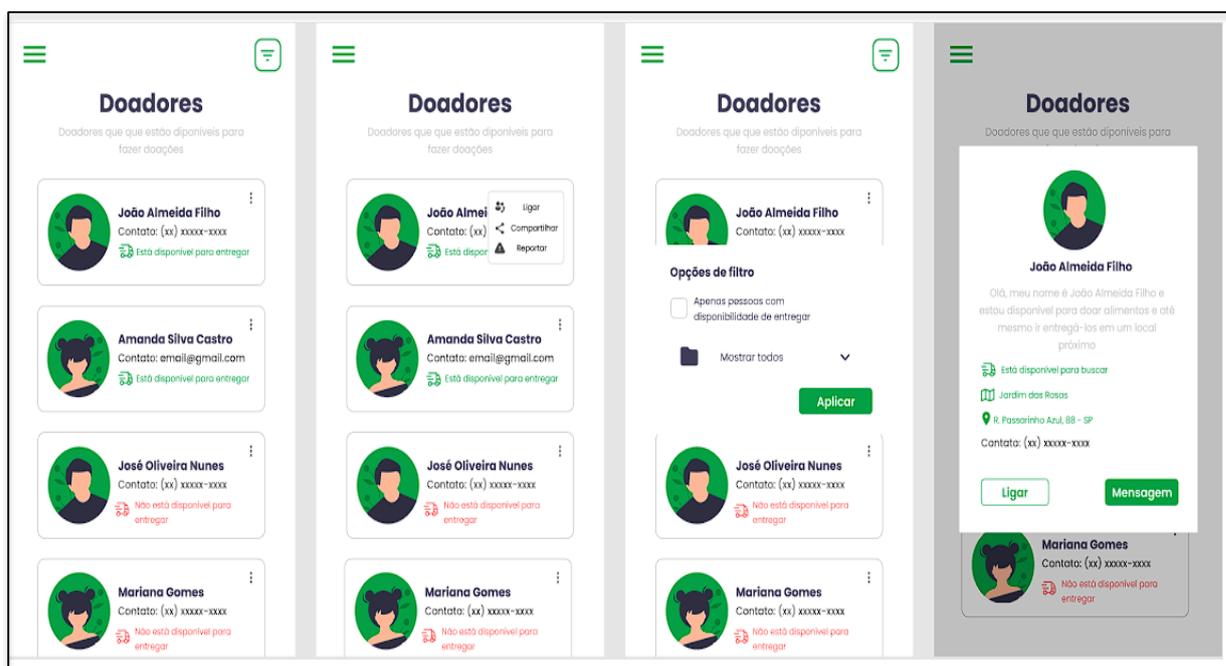


Fonte: Autoria própria, 2022

2.7.4 Tela Doadores do aplicativo

Na tela de doadores, será possível ver as pessoas disponíveis a realizar uma doação para quem precisa. Nessa tela haverá vários perfis, que mostrarão suas fotos, nomes, endereços e se a pessoa está ou não disponível para entregar esse alimento ao usuário. Haverá um filtro para que o usuário encontre somente pessoas que estão disponíveis para entrega, caso ele não possa buscar a doação. Quando o usuário clicar em um dos perfis, será mostrado informações adicionais do doador, além disso, terá a opção de ligar ou mandar mensagem. Também será possível clicar nos três pontos do perfil e o usuário poderá ligar, compartilhar o perfil ou reportar.

Figura 16 – Doadores do Aplicativo

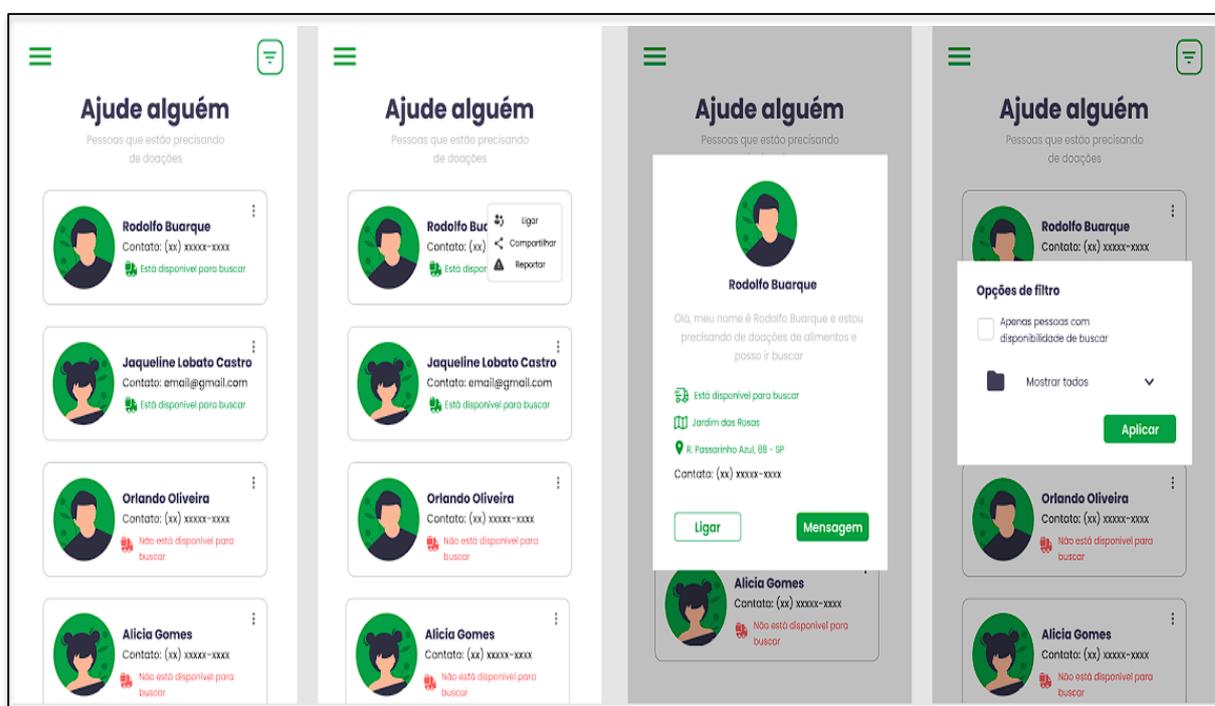


Fonte: Autoria própria, 2022

2.7.5 Tela Ajude Alguém do aplicativo

Na tela de ajudar alguém, será possível ver as pessoas que estão precisando de doações. Nessa tela haverá vários perfis, que mostrarão suas fotos, nomes, endereços e se a pessoa está ou não disponível para buscar esses alimentos do usuário doador. Haverá um filtro para que o usuário encontre somente pessoas que estão disponíveis para buscar, caso ele não possa entregar a doação. Quando o usuário clicar em um dos perfis, será mostrado informações adicionais da pessoa que precisa de doação, além disso, terá a opção de ligar ou mandar mensagem. Também será possível clicar nos três pontos do perfil e o usuário poderá ligar, compartilhar o perfil ou reportar.

Figura 17 – Ajude Alguém do Aplicativo

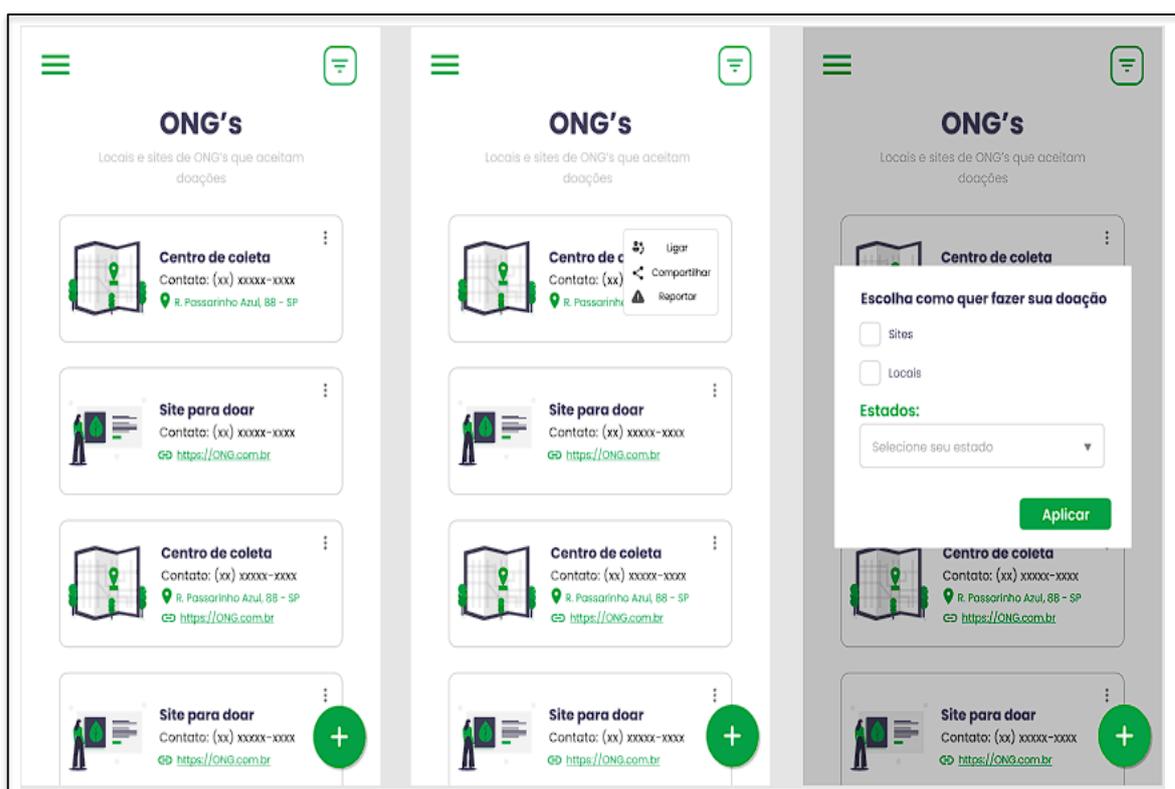


Fonte: Autoria própria, 2022

2.7.6 Tela ONGs do aplicativo

Na tela de ONGs haverá vários perfis de ONGs físicas ou que possuem um site (os ícones de cada um serão diferenciados), seus nomes e endereços ou links. Haverá um filtro para que o usuário encontre somente ONGs físicas (poderá ser filtrado por estado) ou sites. Será possível clicar nos três pontos do perfil e o usuário poderá ligar, compartilhar o perfil ou reportar.

Figura 18 – ONGs do Aplicativo



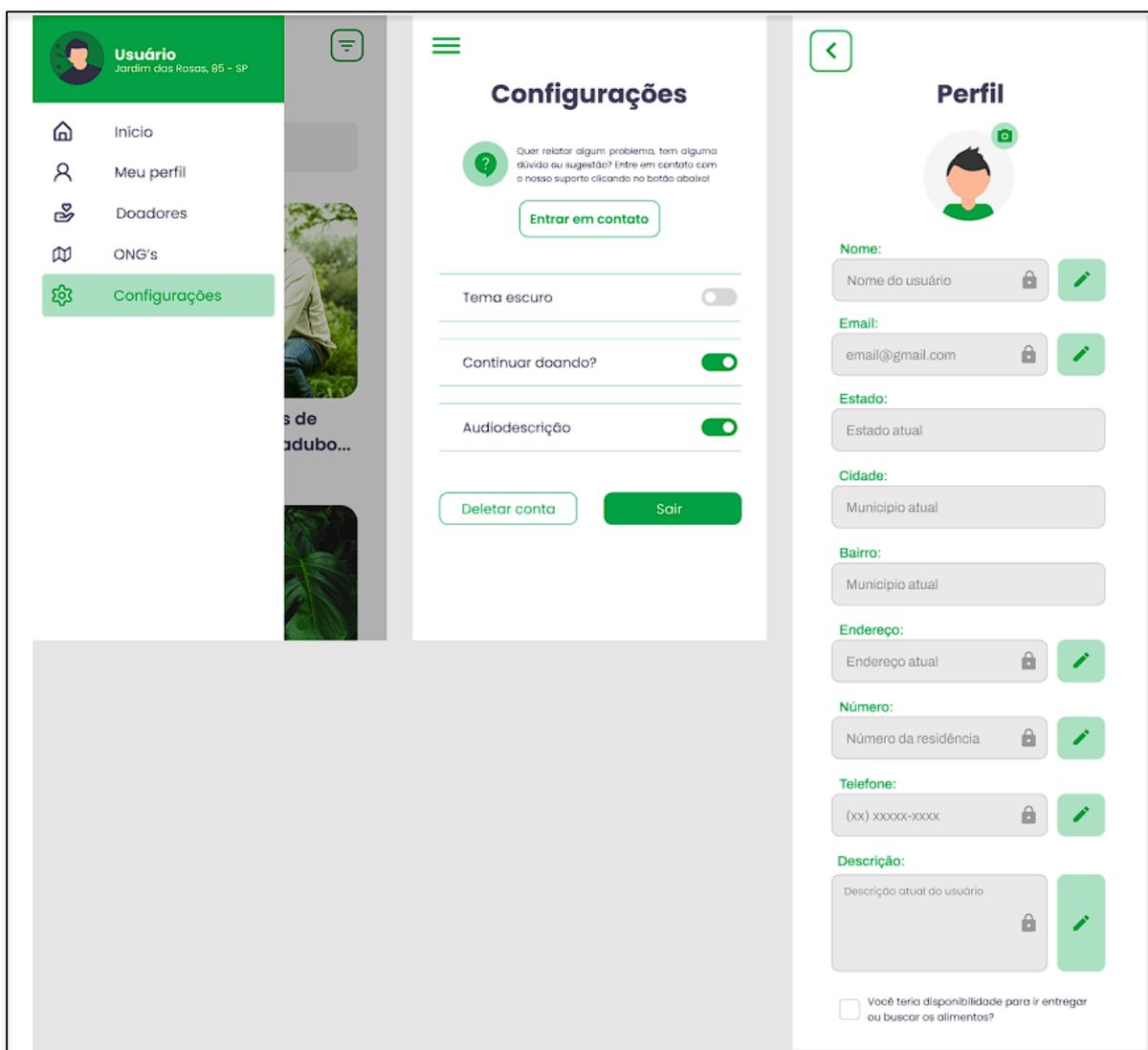
Fonte: Autoria própria, 2022

2.7.7 Tela Configurações/Edição de perfil do aplicativo

Quando o usuário for em "Configurações" ele será direcionado à uma tela na qual poderá clicar em "Entrar em contato", ele irá para o formulário do site GreenPoint para tirar dúvidas, relatar problemas etc... Além disso, o usuário poderá ativar ou desativar o modo escuro do aplicativo, se quer ou não continuar doando (caso seja doador) e ativar ou desativar áudio-descrição. Também terá a opção de sair de sua conta ou deletá-la.

Se o usuário for em "Meu perfil", terá a opção de editar todos seus dados informados anteriormente em seu cadastro.

Figura 19 – Configurações/Edição de Perfil do Aplicativo



Fonte: Autoria própria, 2022

2.7.8 Código da tela Cadastro do aplicativo

Figura 20 – Código da Tela de Cadastro do Aplicativo (1)

```

1  function RegisterStageOne({
2    name,
3    setName,
4    email,
5    setEmail,
6    cpf,
7    setCpf,
8    phoneNumber,
9    setPhoneNumber,
10   profileImage,
11   password,
12   setPassword,
13   confirmPassword,
14   setConfirmPassword,
15   nextStage,
16   navigation,
17   audioDescription
18 }){
19
20   return (
21     <View style={{width: '80%', marginLeft: 'auto', marginRight: 'auto'}}
22       <!-- use profile photo -->
23       <View style={{marginLeft: 'auto', marginRight: 'auto', marginVertical: 10}}>
24         <View style={{backgroundColor: '#AEC2C2', width: 45, height: 45,
25           justifyContent: 'center', alignItems: 'center', borderRadius: 100,
26           top: 30, left: 90, zIndex: 5, overflow: 'hidden'}}>
27           <TouchableNativeFeedback style={{width: 45, height: 45,
28             justifyContent: 'center', alignItems: 'center'}}
29             onPress={() => navigation.navigate("CamScreen", { startingPoint: "RegisterScreen" })}>
30             <Icon name="camera-plus" size={20} color="#03AB45"/>
31           </TouchableNativeFeedback>
32         </View>
33         <Image style={{profileImage == "" ? {} : {borderRadius: 100, width: 136, height: 128}}
34           source={profileImage == "" ? require("../img/ProfileImage.png") : {uri: profileImage}}>
35         </Image>
36       </View>
37       <!--inputs-->
38       <View>
39         <Input audioDescription={audioDescription} label="Nome" value={name} onChangeText={setName} placeholder="Insira seu nome completo"/>
40         <Input audioDescription={audioDescription} label="Email" value={email} onChangeText={setEmail} placeholder="Insira seu Email"/>
41         <View style={{width: '100%', height: 90, marginTop: 10, marginBottom: 10, marginRight: 'auto', marginLeft: 'auto'}}>
42           <Text style={{fontFamily: 'Poppins_600SemiBold', fontSize: 18, marginLeft: 10, color: '#03AB45'}}>CPF:
43           <!*!<Input audioDescription={audioDescription} <icon onPress={() => Speak("CPF")}
44             name="volume-high" size={18} color="#03AB45"/>
45           </Text>
46           <TextInputMask style={{borderWidth: 1, borderColor: '#BFBFBF', borderRadius: 10, padding: 15,
47             height: 40, fontFamily: 'Poppins_400Regular', fontSize: 16}}
48             type="cpf"
49             onChangeText={setCpf} value={cpf} placeholder="Ex: xxx.xxx.xxx-xx"/>
50           </View>
51         </View>
52         <View style={{width: '100%', height: 90, marginTop: 10, marginBottom: 10, marginRight: 'auto', marginLeft: 'auto'}}>
53           <Text style={{fontFamily: 'Poppins_600SemiBold', fontSize: 18, marginLeft: 10, color: '#03AB45'}}>Telefone:
54           <!*!<Input audioDescription={audioDescription} <icon onPress={() => Speak("Telefone")}
55             name="volume-high" size={18} color="#03AB45"/>
56           </Text>
57           <TextInputMask style={{borderWidth: 1, borderColor: '#BFBFBF', borderRadius: 10, padding: 15,
58             height: 40, fontFamily: 'Poppins_400Regular', fontSize: 16}}
59             type="tel-phone" options={{maskType: '98L', withDDD: true}}
60             onChangeText={setPhoneNumber} value={phoneNumber} placeholder="Ex: (xx) xxxxx-xxxx"/>
61           </View>
62         <Input audioDescription={audioDescription} secureText={true} label="Senha" value={password} onChangeText={setPassword} placeholder="Insira sua senha"/>
63         <Input audioDescription={audioDescription} secureText={true} label="Confirma sua senha" value={confirmPassword} onChangeText={setConfirmPassword} placeholder="Insira sua senha novamente"/>
64       </View>
65       <TouchableNativeFeedback onPress={nextStage}>
66         <View style={styles.nextButton}>
67           <Icon name="chevron-right" size={35} color="#03AB45"/>
68         </View>
69       </TouchableNativeFeedback>
70     </View>
71   )
72 }
73
74 }

```

Fonte: Autoria própria, 2022

Figura 21 – Código da Tela de Cadastro do Aplicativo (2)

```

1  function RegisterStageTwo({
2    address,
3    setAddress,
4    city,
5    setCity,
6    isDonor,
7    setIsDonor,
8    acceptedTerms,
9    setAcceptedTerms,
10   nextStage,
11   state,
12   setState,
13   postalCode,
14   setPostalCode,
15   neighborhood,
16   setNeighborhood,
17   houseNumber,
18   setHouseNumber,
19   audioDescription
20 }) {
21   return(
22     <View style={{marginTop: 48, width: "88%", marginLeft: "auto", marginRight: "auto"}}>
23
24     <Input audioDescription={audioDescription} label="CEP:" value={postalCode} onChangeText={setPostalCode} placeholder="Insira seu CEP"/>
25     <Input audioDescription={audioDescription} label="Estado:" value={state} onChangeText={setState} placeholder="Insira seu estado" editable={false}/>
26     <Input audioDescription={audioDescription} label="Cidade:" value={city} onChangeText={setCity} placeholder="Insira sua cidade" editable={false}/>
27     <Input audioDescription={audioDescription} label="Bairro:" value={neighborhood} onChangeText={setNeighborhood} placeholder="Insira seu bairro" editable={false}/>
28     <Input audioDescription={audioDescription} label="Endereço:" value={address} onChangeText={setAddress} placeholder="Ex: Rua das rosas" editable={false}/>
29     <Input audioDescription={audioDescription} label="Número:" value={houseNumber} onChangeText={setHouseNumber} placeholder="Insira o número da casa"/>
30
31     <View style={{marginTop: 26}}>
32       <Text style={styles.modalText}>0 que você deseja?
33       <{"\t"}audioDescription && <Icon onPress={() => Speak("0 que você deseja?")}
34         name="volume-high" size={18} color="#03AB45"/>
35     </Text>
36     <RadioButton.Group
37       onValueChange={userType => userType === "Donor" ? setIsDonor(true) : setIsDonor(false)}
38       value={isDonor === true ? "Donor" : "Receptor"}>
39       <View style={{flexDirection: 'row', alignItems: 'center'}}>
40         <RadioButton color="#03AB45" value="Donor" />
41         <Text style={styles.modalText, {marginLeft: 3}}>>Doar
42           {"\t"}audioDescription && <Icon onPress={() => Speak("Doar")}
43             name="volume-high" size={18} color="#03AB45"/>
44         </Text>
45       </View>
46       <View style={{flexDirection: 'row', alignItems: 'center'}}>
47         <RadioButton color="#03AB45" value="Receptor" />
48         <Text style={styles.modalText, {marginLeft: 4}}>>Receber doações
49           {"\t"}audioDescription && <Icon onPress={() => Speak("Receber doações")}
50             name="volume-high" size={18} color="#03AB45"/>
51         </Text>
52       </View>
53     </RadioButton.Group>
54
55     <View style={{flexDirection: 'row', marginTop: 10, marginBottom: 10}}>
56     <Checkbox status={acceptedTerms ? 'checked' : 'unchecked'}
57       onPress={() => setAcceptedTerms(acceptedTerms)} color="#03AB45"/>
58     <Text style={styles.modalText}>Declaro que li os <Text onPress={() => Linking.openURL('https://green-point.netlify.app')}
59       style={color: "#03AB45"}>Termos de Uso</Text>, e estou ciente.
60     {"\t"}audioDescription && <Icon onPress={() => Speak("Declaro que li os termos de uso no texto verde e estou ciente deles")}
61       name="volume-high" size={18} color="#03AB45"/>
62     </Text>
63   </View>
64 </View>
65
66 <TouchableNativeFeedback onPress={nextStage}>
67 <View style={styles.nextButton}>
68 <Icon name="chevron-right" size={35} color="#03AB45"/>
69 </View>
70 </TouchableNativeFeedback>
71 </View>
72 )
73 }

```

Fonte: Autoria própria, 2022

Figura 22 – Código da Tela de Cadastro do Aplicativo

```

1 function RegisterStageThree({
2   isDonor,
3   isAdult,
4   setIsAdult,
5   oavailability,
6   setAvailability,
7   income,
8   setIncome,
9   dependents,
10  setDependents,
11  registerFirebase,
12  audioDescription
13 }) {
14   return (
15     <View style={{width: "88%", marginLeft: "auto", marginRight: "auto", marginTop: 20}}>
16
17     <View style={{marginTop: 40}}>
18       <text style={{styles.modalText}}>Você se disponibiliza a levar alimentos próximo ao seu local?
19       <["\t"]audioDescription && <Icon onPress={() => Speak("Você se disponibiliza a levar alimentos próximo ao seu local?")}
20       name="volume-high" size={18} color="#83A845"/>
21     </Text>
22     <RadioButton.Group
23       onChange={userType => userType = "Donor" ? setAvailability(true) : setAvailability(false)}
24       value={availability === true ? "Donor" : "Receptor"}>
25       <View style={{flexDirection: "row", alignItems: "center"}}>
26         <RadioButton color="#83A845" value="Donor" />
27         <text style={{styles.modalText, marginLeft: 20}}>Sim
28         <["\t"]audioDescription && <Icon onPress={() => Speak("Sim")}
29         name="volume-high" size={18} color="#83A845"/>
30       </Text>
31     </View>
32     <View style={{flexDirection: "row", alignItems: "center"}}>
33     <RadioButton color="#83A845" value="Receptor" />
34     <text style={{styles.modalText, marginLeft: 20}}>Não
35     <["\t"]audioDescription && <Icon onPress={() => Speak("Não")}
36     name="volume-high" size={18} color="#83A845"/>
37   </Text>
38 </View>
39 </RadioButton.Group>
40 </View>
41
42 <View style={{marginTop: 40}}>
43 <text style={{styles.modalText}}>Você tem mais de 18 anos?<["\t"]audioDescription && <Icon onPress={() => Speak("Você tem mais de 18 anos?")}
44 name="volume-high" size={18} color="#83A845"/></Text>
45 <RadioButton.Group
46   onChange={userType => userType = "Donor" ? setIsAdult(true) : setIsAdult(false)}
47   value={isAdult === true ? "Donor" : "Receptor"}>
48   <View style={{flexDirection: "row", alignItems: "center"}}>
49     <RadioButton color="#83A845" value="Donor" />
50     <text style={{styles.modalText, marginLeft: 20}}>Sim
51     <["\t"]audioDescription && <Icon onPress={() => Speak("Sim")}
52     name="volume-high" size={18} color="#83A845"/>
53   </Text>
54 </View>
55 <View style={{flexDirection: "row", alignItems: "center"}}>
56 <RadioButton color="#83A845" value="Receptor" />
57 <text style={{styles.modalText, marginLeft: 20}}>Não
58 <["\t"]audioDescription && <Icon onPress={() => Speak("Não")}
59 name="volume-high" size={18} color="#83A845"/>
60 </Text>
61 </View>
62 </RadioButton.Group>
63 </View>
64
65 <text style={{styles.modalText, marginLeft: 20}}>Tem dúvidas? Entre no nosso<text style={{color: "#83A845"}}> site.</Text>
66 <["\t"]audioDescription && <Icon onPress={() => Speak("Tem dúvidas? Entre no nosso site clicando no texto verde")}
67 name="volume-high" size={18} color="#83A845"/>
68 </Text>
69
70 <text style={{styles.modalText, marginLeft: 10, marginBottom: 10}}>Seus dados estão protegidos. Visite nossa <text style={{color: "#83A845"}}> Política de Privacidade.</Text>
71 <["\t"]audioDescription && <Icon onPress={() => Speak("Seus dados estão protegidos. Visite nossa Política de Privacidade clicando no texto verde")}
72 name="volume-high" size={18} color="#83A845"/>
73 </Text>
74
75 <ButtonMain label="Cadastrar-se" onPress={registerFirebase}/>
76 </View>
77
78
79
80 <View style={{width: "88%", marginLeft: "auto", marginRight: "auto", marginTop: 40}}>
81 <Input audioDescription={audioDescription} label="Renda Familiar Mensal" value={income} onChangeText={setIncome} placeholder="Insira a renda total da sua família"/>
82 <Input audioDescription={audioDescription} label="Número de dependentes dessa renda" value={dependents} onChangeText={setDependents} placeholder="Insira o número de dependentes" />
83
84 <View style={{flexDirection: "row", marginTop: 40, marginBottom: 20}}>
85 <Checkbox status={availability ? "checked" : "unchecked"}
86   onPress={() => setAvailability(availability)} color="#83A845"/>
87
88 <text style={{styles.modalText}}>Eu me disponibilizo a buscar alimentos próximos ao meu local
89 <["\t"]audioDescription && <Icon onPress={() => Speak("Eu me disponibilizo a buscar alimentos próximos ao meu local")}
90 name="volume-high" size={18} color="#83A845"/>
91 </Text>
92 </View>
93
94 <text style={{styles.modalText, marginBottom: 30}}>Seus dados estão protegidos. Visite nossa <text style={{color: "#83A845"}}> Política de Privacidade.</Text>
95 <["\t"]audioDescription && <Icon onPress={() => Speak("Seus dados estão protegidos. Visite nossa Política de Privacidade clicando no texto verde")}
96 name="volume-high" size={18} color="#83A845"/>
97 </Text>
98
99 <ButtonMain label="Cadastrar-se" onPress={registerFirebase}/>
100 </View>
101 }
102 }

```

Fonte: Autoria própria, 2022

Figura 23 – Código da Tela de Cadastro do Aplicativo

```

1  const addUserDatabase = (user) => {
2      const database = firebase.firestore();
3      console.log(audioDescription);
4
5      database.collection('users').doc(user.uid).set({
6          userType: isDonor ? "donors" : "receivers",
7      }).then(() => {
8          if(isDonor)
9              database.collection("donors").doc(user.uid).set({
10                 id: user.uid,
11                 fullName: user.displayName,
12                 email: user.email,
13                 profileImage: user.photoURL,
14                 phoneNumber: phoneNumber,
15                 availability: availability,
16                 cpf: cpf,
17                 postalCode: postalCode,
18                 neighborhood: neighborhood,
19                 address: address + ", " + houseNumber,
20                 state: state,
21                 city: city,
22                 isActiveUser: true,
23                 audioDescription: audioDescription,
24                 darkTheme: false,
25                 description: "Sem descrição"
26             }).then(() => {
27                 setIsLoading(false);
28                 navigation.reset({
29                     index: 0,
30                     routes: [{name: 'LoginScreen', params: {audioDescriptionIsActive: audioDescription}}],
31                 });
32             }).catch((error) => {
33                 let errorCode = error.code;
34                 let errorMessage = error.message;
35                 console.log(errorMessage);
36                 alert("Não foi possível efetuar o registro !!");
37             })
38         } else {
39             database.collection("receivers").doc(user.uid).set({
40                 id: user.uid,
41                 fullName: user.displayName,
42                 email: user.email,
43                 profileImage: user.photoURL,
44                 phoneNumber: phoneNumber,
45                 availability: availability,
46                 dependents: dependents,
47                 income: income,
48                 cpf: cpf,
49                 postalCode: postalCode,
50                 neighborhood: neighborhood,
51                 address: address + ", " + houseNumber,
52                 state: state,
53                 city: city,
54                 isActiveUser: true,
55                 audioDescription: audioDescription,
56                 darkTheme: false,
57                 description: "Sem descrição"
58             }).then(() => {
59                 setIsLoading(false);
60                 navigation.navigate('LoginScreen', {audioDescriptionIsActive: audioDescription});
61             }).catch((error) => {
62                 let errorCode = error.code;
63                 let errorMessage = error.message;
64                 console.log(errorMessage);
65                 alert("Não foi possível efetuar o registro !!");
66             })
67         }
68     }).catch((error) => {
69         let errorCode = error.code;
70         let errorMessage = error.message;
71         console.log(errorMessage);
72         alert("Não foi possível efetuar o registro !!");
73     })
74 }

```

Fonte: Autoria própria, 2022

2.7.9 Código da tela Login do aplicativo

Figura 24 – Código da Tela de Login do Aplicativo

```

1  export default function Login({navigation, route}) {
2    const [email, setEmail] = useState("");
3    const [password, setPassword] = useState("");
4
5    const loginFirebase = () => {
6      firebase.auth().setPersistence(firebase.auth.Auth.Persistence.LOCAL)
7      .then(() => {
8        firebase.auth().signInWithEmailAndPassword(email.trim(), password.trim())
9          .then((userCredential) => {
10             var user = userCredential.user;
11
12             navigation.navigate("AppDrawer");
13           })
14           .catch((error) => {
15             var errorCode = error.code;
16             var errorMessage = error.message;
17             alert("Login e/ou senha incorretos !!");
18           });
19     });
20   }
21
22   return (
23     <View style={styles.container}>
24       <ScrollView>
25         <StatusBar style="dark-content" translucent backgroundColor="transparent" />
26         <Back onPress={() => navigation.goBack()} />
27         <View style={{width: "80%", marginLeft: "auto", marginRight: "auto"}}>
28           <Text style={styles.text, color: "#3385ff"}><em>Sem vindo(a) de volta!</em>
29           <{!route || route.params.audioDescriptionIsActive && <Icon
30             onPress={() => Speech("Sem vindo de volta! Essa é a tela de login, para entrar, digite seu email e senha e clique no botão abaixo, caso tenha perdido sua senha e queira redefini-la, clique no texto verde")
31             name="volume-high" size={28} color="#838045" />
32           </Text>
33
34           <Image style={{marginTop: 40, marginBottom: 20}} source={require("../img/LoginImage.png")} />
35
36           <Input label="Email:" value={email} onChangeText={text => setEmail(text)}
37             placeholder="Insira seu Email" />
38
39           <Input secureText={true} label="Senha:" value={password} onChangeText={text => setPassword(text)}
40             placeholder="Insira sua senha" />
41
42           <Text onPress={() => navigation.navigate("ResetPasswordScreen", {audioDescriptionIsActive: route.params.audioDescriptionIsActive})}
43             style={styles.forgetText}>Esqueceu sua senha?</Text>
44
45           <View style={{marginBottom: 30}}>
46             <ButtonMain label="Entrar" onPress={loginFirebase} />
47           </View>
48         </View>
49       </ScrollView>
50     </View>
51   );
52 }
53
54 }

```

Fonte: Autoria própria, 2022

2.7.10 Código da tela Notícias do aplicativo

Figura 25 – Código da Tela de Notícias do Aplicativo

```

1  return(
2
3      <Provider>
4
5          <Portal>
6
7              <view style={{style.container, backgroundColor: theme.backgroundColor}}>
8                  <StatusBar style={theme.statusBar} translucent backgroundColor="transparent"/>
9                  <view style={{flexDirection: 'row'}}>
10                     <TouchableOpacity onPress={() => navigation.toggleDrawer()}>
11                         <FilterBottom onPress={() => navigation.toggleDrawer()}>
12                     </view>
13
14                     <Text style={{style.title, color: theme.secondaryColor}}>
15                         Notícias{"\n"}
16                     </Text>
17
18                     <TouchableOpacity onPress={() => navigation.toggleDrawer()}>
19                         <Text style={{fontStyle: 'italic', color: theme.secondaryColor}}>
20                             Para saber mais sobre uma notícia, pressione sobre ela e segure{"\n"}
21                         </Text>
22                     </TouchableOpacity>
23
24                     <view style={{alignItems: 'center', marginBottom: 10}}>
25                         <Search backgroundColor={theme.secondaryBackgroundColor} color={theme.secondaryColor}>
26                             <input type="text" value={searchInput} onChange={setSearchInput}/>
27                         </view>
28
29                     <FlatList data={results.sort((a, b) => compareDates(a.consultationDate, b.consultationDate))} keyExtractor={(item, index) => index.toString()} renderItem={({item} =>
30
31                         <TouchableOpacity onPress={() => navigation.navigate('NewsDetails', {newsId: item.id})}>
32                             <view style={{style.newsContainer}}>
33                                 <Image
34                                     style={{style.image}}
35                                     source={item.image}
36                                     alt={item.title}
37                                 </Image>
38
39                                 <Text style={{style.newsTitle, color: theme.secondaryColor}}>{item.title}</Text>
40                                 <Text style={{style.newsLink}}>{item.link}</Text>
41                             </view>
42                         </TouchableOpacity>
43                     </FlatList>
44
45                     <FilterModal>
46
47                         <Modal contentContainerStyle={{style.modal, backgroundColor: theme.backgroundColor}} isVisible={isVisible} onRequestClose={() => filter()}>
48                             <Text style={{style.modalTitle, color: theme.secondaryColor}}>
49                                 Escolha o tema das notícias exibidas {"\n"}
50                             </Text>
51
52                             <TouchableOpacity onPress={() => toggleFilter('Agricoltura')}>
53                                 <Text style={{style.filterText, color: theme.secondaryColor}}>
54                                     Agricoltura{"\n"}
55                                 </TouchableOpacity>
56
57                             <TouchableOpacity onPress={() => toggleFilter('Alimentação')}>
58                                 <Text style={{style.filterText, color: theme.secondaryColor}}>
59                                     Alimentação{"\n"}
60                                 </TouchableOpacity>
61
62                             <TouchableOpacity onPress={() => toggleFilter('Sustentabilidade')}>
63                                 <Text style={{style.filterText, color: theme.secondaryColor}}>
64                                     Sustentabilidade{"\n"}
65                                 </TouchableOpacity>
66
67                             <TouchableOpacity style={{height: 30, width: '100%', borderRadius: 5, overflow: 'hidden', margin: 5}}>
68                                 <Text style={{color: theme.secondaryColor}}>
69                                     Fechar {"\n"}
70                             </TouchableOpacity>
71
72                         </Modal>
73
74                     </Provider>
75
76                 </Portal>
77
78             </Provider>
79
80         </return>
81     )
82 }

```

Fonte: Autoria própria, 2022

2.7.11 Código da tela ONGs do aplicativo

Figura 26 – Código da Tela de ONGs do Aplicativo

```

1  return(
2
3      <Provider>
4      <Portal>
5          <View style={{styles.container, {backgroundColor: theme.backgroundColor}}>
6              <StatusBar style={theme.statusBar} translucent backgroundColor="transparent" />
7              <View style={{flexDirection: 'row'}}>
8                  <MenuDrawer onPress={() => navigation.toggleDrawer()} />
9                  <FilterButton onPress={() => setFilterVisible(true)} />
10             </View>
11
12             <View style={{marginLeft: 'auto', marginRight: 'auto', width: '80%', alignItems: 'center'}}>
13                 <Text style={{styles.textTitle, {color: theme.secondaryColor}}>ONG's
14                     {"\t"}{userData.audioDescription} && <Icon2
15                         onPress={() => Speak("Tela de ONG's")} name="volume-high" size={28} color="#83A045" /></Text>
16                 <Text style={styles.textSubtitle}>Locais e sites de ONG's que aceitam doações{"\t"}{userData.audioDescription} && <Icon2
17                     onPress={() => Speak("Locais e sites de ONG's que aceitam doações")} name="volume-high" size={22} color="#83A045" /></Text>
18             </View>
19
20             <FlatList
21                 data={filteredPlacesAndSites}
22                 keyExtractor={({item, index}) => index.toString()}
23                 renderItem={({item, index}) =>
24                     <View style={{alignItems: 'center'}}>
25                         <View style={{width: '85%', borderColor: theme.borderColor, backgroundColor: theme.boxBackgroundColor, borderWidth: 1, borderRadius: 18,
26                             justifyContent: 'center', marginTop: 18, marginBottom: 18, overflow: 'hidden'}}>
27                             <View style={{marginLeft: 'auto', right: 15, zIndex: 5, position: 'absolute', top: 15}}>
28                                 <Menu contentStyle={{backgroundColor: theme.backgroundColor}} visible={menu == index ? true : false} onDismiss={() => setMenu(0)}
29                                     onChange={icon2 name="dots-vertical" onPress={() => setMenu(index)}
30                                     size={25} color="#7A7A7A" />
31
32                                 <Menu.Item leadingIcon="phone"
33                                     onPress={() => Linking.openURL('tel:${item.phoneNumber}')}
34                                     title="Ligar" />
35
36                                 <Menu.Item leadingIcon="share-variant"
37                                     onPress={() => onShare(item.address, item.title, item.link, item.phoneNumber)}
38                                     title="Compartilhar" />
39
40                                 <Menu.Item leadingIcon="alert" onPress={() => Linking.openURL('https://forms.gle/E73jfmCJbwqx5gJ8')} title="Reportar" />
41                             </Menu>
42                         </View>
43
44                         <TouchableNativeFeedback onPress={() => openModal(item)}>
45                             <View style={{flexDirection: 'row', padding: 20}}>
46                                 <Image
47                                     style={styles.imagePlace}
48                                     source={item.address ? require('../img/Location-FlatList.png') : require('../img/Site.png')}
49                                 />
50                                 <View style={{width: '60%'}}>
51                                     <Text numberOfLines={1} style={{styles.textTitleFlatList, {color: theme.secondaryColor}}>{item.title}</Text>
52                                     <Text style={{fontFamily: 'Poppins_400Regular', fontSize: 12, color: '#545454'}}>Contato: {item.phoneNumber}</Text>
53                                 <View style={{flexDirection: 'row', marginTop: 8, alignItems: 'center', item.state == null ? {display: 'none'} : {}}>
54                                     <Icon name="map" size={19} color="#83A045" />
55                                     <Text numberOfLines={1} style={styles.subTextFlatList}> {item.state ? item.state : ""}</Text>
56                                 </View>
57                                 <View style={{flexDirection: 'row', marginTop: 8, item.link == null ? {display: 'none'} : {}}>
58                                     <Icon name="Link-2" size={19} color="#83A045" />
59                                     <Text numberOfLines={1} style={styles.subTextFlatList}> {item.link ? item.link : ""}</Text>
60                                 </View>
61                             </View>
62                         </TouchableNativeFeedback>
63                     </View>
64                 </FlatList>
65             </Provider>
66         </Portal>
67     </View>
68 )

```

Fonte: Autoria própria, 2022

Figura 27 – Código da Tela de ONGs do Aplicativo

```

1  {/**Modal filter*/}
2  <Modal onDismiss={() => setFilterVisible(false)} visible={filterVisible} contentContainerStyle={{styles.modalStyle, {backgroundColor: theme.backgroundColor}}}
3  <Text style={styles.filterTextBold, {color: theme.secondaryColor}}>Escolha como quer fazer sua doação</Text>
4  {userData.audioDescription &&
5  <Icon2 onPress={() => Speak("Escolha como quer fazer sua doação")} name="volume-high" size={20} color="#03A845"/>
6  </Text>
7  <View style={{flexDirection: 'row', marginTop: 10, marginBottom: 10, alignItems: 'center'}}>
8  <Checkbox status={filterSites ? 'checked' : 'unchecked'} color="#03A845"
9  onPress={() => setFilterSites(filterSites)}/>
10 <Text style={styles.modalText, {color: theme.secondaryColor}}>Sites
11 {`${userData.audioDescription} &&
12 <Icon2 onPress={() => Speak("Através de sites")} name="volume-high" size={20} color="#03A845"/>
13 </Text>
14 </View>
15 </View>
16 <View style={{flexDirection: 'row', marginTop: 10, marginBottom: 10, alignItems: 'center'}}>
17 <Checkbox status={filterPlaces ? 'checked' : 'unchecked'} color="#03A845"
18 onPress={() => setFilterPlaces(filterPlaces)}/>
19 <Text style={styles.modalText, {color: theme.secondaryColor}}>Locais
20 {`${userData.audioDescription} &&
21 <Icon2 onPress={() => Speak("Através de locais")} name="volume-high" size={20} color="#03A845"/>
22 </Text>
23 </View>
24 </View>
25 <View style={filterPlaces ? {display: 'flex'} : {display: 'none'}}>
26 <ComboBox value={filterState} onPress={(item) => setFilterState(item)}/>
27 </View>
28 </View>
29 <View style={{height: 35, width: "45%", borderRadius: 5, overflow: 'hidden', marginLeft: "auto", marginTop: 30}}>
30 <TouchableNativeFeedback onPress={() => filter()}>
31 <View style={{backgroundColor: "#03A845", justifyContent: 'center', alignItems: 'center',
32 height: '100%', borderRadius: 5}}>
33 <Text style={styles.modalTextButtons, {color: "white"}}>Aplicar</Text>
34 </View>
35 </TouchableNativeFeedback>
36 </View>
37 </Modal>

```

Fonte: Autoria própria, 2022

Figura 28 – Código da Tela de ONGs do Aplicativo

```

1  {/**Modal places and sites*/}
2  <Modal contentContainerStyle={styles.modalStyle, {backgroundColor: theme.backgroundColor}} visible={placesAndSitesModal} onDismiss={() => setPlacesAndSitesModal(false)}
3  <View style={{alignItems: 'center', width: "100%"}}>
4  <Image style={{width: 100, height: 80, marginBottom: 10}}
5  source={dataModal.address ? require("../img/location-flatlist.png") : require("../img/site.png")}/>
6  <Text style={styles.filterTextBold, {fontSize: 16, textAlign: 'center', color: theme.secondaryColor}}>{dataModal.title}</Text>
7  <Text style={{fontFamily: 'Poppins_400Regular', fontSize: 16, color: '#545454', marginBottom: 20}}>Contato: {dataModal.phoneNumber}</Text>
8  <View style={styles.dataModalStyle, dataModal.state === null ? {display: 'none'} : {}}>
9  <Icon name="map" size={20} color="#03A845"/>
10 <Text numberOfLines={1} style={styles.subTextFlatList, {width: 'auto', fontSize: 15, marginLeft: 5}}>
11 {dataModal.state ? dataModal.state : ""}</Text>
12 </View>
13 <View style={styles.dataModalStyle, dataModal.address === null ? {display: 'none'} : {alignItems: 'flex-start'}}>
14 <Icon name="map-pin" size={20} color="#03A845"/>
15 <Text style={styles.subTextFlatList, {width: 'auto', fontSize: 12, marginLeft: 5, textAlign: 'center'}}>
16 {dataModal.address ? dataModal.address : ""}</Text>
17 </View>
18 <View style={styles.dataModalStyle, dataModal.link === null ? {display: 'none'} : {}}>
19 <Icon name="link-2" size={20} color="#03A845"/>
20 <Text onPress={() => Linking.openURL(dataModal.link)}
21 numberOfLines={1} style={styles.subTextFlatList, {width: 'auto', fontSize: 15, marginLeft: 5}}>
22 {dataModal.link ? dataModal.link : ""}</Text>
23 </View>
24 </View>
25 </View>
26 {/**Modal buttons*/}
27 <View style={{flexDirection: 'row', width: "100%", marginTop: 45}}>
28 <View style={{height: 35, width: "45%", borderRadius: 5, overflow: 'hidden'}}>
29 <TouchableNativeFeedback onPress={() => setPlacesAndSitesModal(false)}>
30 <View style={{backgroundColor: "transparent", justifyContent: 'center', alignItems: 'center',
31 height: '100%', border: 1px solid "#03A845", borderRadius: 5}}>
32 <Text style={styles.modalTextButtons}>Cancelar</Text>
33 </View>
34 </TouchableNativeFeedback>
35 </View>
36 </View>
37 <View style={{height: 35, width: "45%", borderRadius: 5, overflow: 'hidden', marginLeft: "auto"}}>
38 <TouchableNativeFeedback onPress={() => Linking.openURL('tel:${dataModal.phoneNumber}')}>
39 <View style={{backgroundColor: "#03A845", justifyContent: 'center', alignItems: 'center',
40 height: '100%', borderRadius: 5}}>
41 <Text style={styles.modalTextButtons, {color: "white"}}>Ligar</Text>
42 </View>
43 </TouchableNativeFeedback>
44 </View>
45 </View>
46 </Modal>
47 </View>
48 </Provider>
49 </Portal>
50 </Provider>
51 </>
52 )

```

Fonte: Autoria própria, 2022

2.7.12 Código da tela Doadores/Ajude Alguém do aplicativo

Figura 29 – Código da Tela de Doadores/Ajude Alguém do Aplicativo

```

1  return (
2
3      <Provider>
4      <Portal>
5
6          <View style={styles.container, {backgroundColor: theme.backgroundColor}}>
7              <StatusBar style={theme.statusBar} translucent backgroundColor="transparent"/>
8              <View style={{flexDirection: 'row'}}>
9                  <MenuDrawer onPress={() => navigation.toggleDrawer()}>
10                     <FilterButton onPress={() => setVisibleFilter(true)}/>
11                 </View>
12
13                 <View style={styles.content}>
14                     <Text style={styles.textTitle, {color: theme.secondaryColor}}>{userData.userType === "donors" ? "Ajude alguém" : "Doadores"}
15                     <Text style={styles.textDescription, {color: theme.secondaryColor}}>{userData.audioDescription}
16                     <Icon name="volume-high" size={28} color="#83A845"/>
17                     </Text>
18                     <Text style={styles.textSubtitle}>
19                         {userData.userType === "donors" ? "Pessoas que estão precisando de doações" : "Doadores que estão disponíveis para fazer doações"}
20                     <Text style={styles.textDescription, {color: theme.secondaryColor}}>{userData.audioDescription}
21                     <Icon name="volume-high" size={22} color="#83A845"/>
22                     </Text>
23                 </View>
24
25                 <FlatList data={filteredUsers} keyExtractor={({item, index}) => index.toString()} renderItem={({item, index}) =>
26                     <View style={{width: '98%', marginTop: 10, marginBottom: 10,
27                         marginLeft: 'auto', marginRight: 'auto', overflow: 'hidden', borderRadius: 10}}>
28                         <View style={{marginLeft: 'auto', top: 20, right: 20, zIndex: 5, position: 'absolute'}}>
29                             <Menu contentsStyle={{backgroundColor: theme.backgroundColor}} visible={menu === index ? true : false} onDismiss={() => setMenu(-1)}
30                                 anchor=<Icon name="dots-vertical" onPress={() => setMenu(index)}
31                                     size={25} color="#7A7A7A"/>
32
33                                 <Menu.Item leadingIcon="phone"
34                                     onPress={() => Linking.openURL('tel:${item.phoneNumber}')}>
35                                     title="Ligar" />
36
37                                 <Menu.Item leadingIcon="share-variant"
38                                     onPress={() => onShare(item.phoneNumber, item.fullName, item.description)}>
39                                     title="Compartilhar" />
40
41                                 <Menu.Item leadingIcon="alert" onPress={() => Linking.openURL('https://forms.gle/ZkueDWEtdoSKDRc6')} title="Reportar" />
42                             </Menu>
43                         </View>
44
45                         <TouchableNativeFeedback onPress={() => {setModalItem(item); userData.audioDescription && userAudioDescription(item)}}>
46                             <View style={{borderColor: theme.borderColor, borderColor: theme.backgroundColor, backgroundColor: theme.backgroundColor, borderWidth: 1,
47                                 borderRadius: 10, padding: 20, flexDirection: 'row'}}>
48                                 <Image style={{height: 100, width: 100, borderRadius: 100, marginTop: 'auto', marginBottom: 'auto'}} source={{uri: item.profileImage}}/>
49                                 <View style={{flexDirection: 'column', marginLeft: 20, width: '55%', marginTop: 'auto', marginBottom: 'auto'}}>
50                                     <Text numberOfLines={2} style={styles.nameUser, {color: theme.secondaryColor}}>{item.fullName}</Text>
51                                     <Text numberOfLines={1} style={styles.contactUser}>Contato: {item.phoneNumber}</Text>
52
53                                     <View style={{flexDirection: 'row', justifyContent: 'center', alignItems: 'center'}}>
54                                         <Icon name={userData.userType === "donors" ? "truck-delivery-outline" : "truck-fast-outline"} size={30}
55                                             color={item.availability ? "#83A845" : "#FF6161"} />
56                                         <Text style={styles.donate, {color: theme.secondaryColor}}>{item.availability} {item.availability ? "Está disponível para"
57                                             : "Não está disponível para"} {userData.userType === "donors" ? "buscar" : "entregar"}</Text>
58                                     </View>
59                                 </View>
60                             </View>
61                         </TouchableNativeFeedback>
62
63                     </View>
64                 </FlatList>
65             </View>
66         </Provider>
67     </Portal>
68 )

```

Fonte: Autoria própria, 2022

Figura 30 – Código da Tela de Doadores/Ajude Alguém do Aplicativo

```

1  {/User modal/}
2  <Modal contentContainerStyle={styles.modalStyle, {backgroundColor: theme.backgroundColor}} visible={modalItem ? true : false} onDismiss={() => setModalItem("")}>
3  <Image style={{height: 130, width: 134, borderRadius: 100, marginTop: 20, marginLeft: "auto", marginRight: "auto"}} source={{uri: modalItem.profileImage}}/>
4  <Text style={styles.nameUser, {textAlign: 'center', color: theme.secondaryColor}}>{modalItem.fullName}</Text>
5  <Text style={styles.modalDescription}>{modalItem.description}</Text>
6
7  <View style={{flexDirection: 'row', alignItems: 'center'}}>
8  <Icon name={userData.userType === "donors" ? "truck-delivery-outline" : "truck-fast-outline"}
9  size={20} color={modalItem.availability ? "#838045" : "#FF6161"}/>
10 <Text style={styles.donate, {fontWeight: 'bold', color: "#FF6161"}}>
11 {modalItem.availability ? "Esta disponível para"
12 : "Não esta disponível para entregar"} {userData.userType === "donors" ? "buscar" : "entregar"}
13 </Text>
14 </View>
15
16 <View style={{flexDirection: 'row', marginTop: 8, alignItems: 'center', marginLeft: 4}}>
17 <Icon name="map" size={21} color="#838045"/>
18 <Text numberOfLines={1} style={styles.subTextFlatList, {width: '85%', fontSize: 12, marginLeft: 10}}>
19 {modalItem.city} - {modalItem.state}
20 </Text>
21 </View>
22
23 <View style={{flexDirection: 'row', marginTop: 18, alignItems: 'center', marginLeft: 4}}>
24 <Icon name="map-pin" size={21} color="#838045"/>
25 <Text numberOfLines={2} style={styles.subTextFlatList, {width: '85%', fontSize: 12, marginLeft: 10}}>
26 {modalItem.address} - {modalItem.neighborhood} {modalItem.state}
27 </Text>
28 </View>
29
30 <Text style={styles.contactUser, {fontSize: 14, marginTop: 10}}>Contato: (* * + {modalItem.phoneNumber}</Text>
31
32 {/Modal buttons/}
33 <View style={{flexDirection: 'row', width: '100%', marginTop: 20}}>
34 <View style={{height: 35, width: "45%", borderRadius: 5, overflow: 'hidden'}}>
35 <TouchableNativeFeedback onPress={() => Linking.openURL('tel://'+modalItem.phoneNumber')}>
36 <Text style={{backgroundColor: "transparent", justifyContent: 'center', alignItems: 'center',
37 height: '100%', borderRadius: 5, border: '1px solid #838045', borderStyle: 'solid'}}>
38 <Text style={styles.modalTextButtons}>Ligar</Text>
39 </View>
40 </TouchableNativeFeedback>
41 </View>
42
43 <View style={{height: 35, width: "45%", borderRadius: 5, overflow: 'hidden', marginLeft: "auto"}}>
44 <TouchableNativeFeedback onPress={() => Linking.openURL('http://api.whatsapp.com/send?phone=55'+modalItem.phoneNumber')}>
45 <Text style={{backgroundColor: "#838045", color: "white", justifyContent: 'center', alignItems: 'center',
46 height: '100%', borderRadius: 5}}>
47 <Text style={styles.modalTextButtons, {color: "white"}}>Mensagem</Text>
48 </View>
49 </TouchableNativeFeedback>
50 </View>
51 </View>
52 </Modal>

```

Fonte: Autoria própria, 2022

Figura 31 – Código da Tela de Doadores/Ajude Alguém do Aplicativo

```

1  {/Filter - Modals/}
2  <Modal visible={visibleFilter} contentContainerStyle={styles.modalStyle, {backgroundColor: theme.backgroundColor}} onDismiss={() => filter()}>
3  <View style={{width: "100%"}}>
4  <Text style={styles.nameUser, {fontSize: 20, color: theme.secondaryColor}}>Opções de filtro</Text>
5  <Text style={styles.nameUser, {fontSize: 20, color: theme.secondaryColor}}>{userData.audioDescription} &&
6  <Icon onPress={() => Speak("Opções de filtro")} name="volume-high" size={20} color="#838045"/>
7  </Text>
8
9  <View style={{flexDirection: 'row', marginTop: 10, marginBottom: 10, alignItems: 'center'}}>
10 <Checkbox status={hasAvailability ? 'checked' : 'unchecked'} color="#838045"
11 onPress={() => setHasAvailability(hasAvailability)}>
12 <Text style={styles.modalText, {color: theme.secondaryColor}}>Apenas pessoas com disponibilidade de {userData.userType === "donors" ? "entregar" : "buscar"}
13 </Text>
14 <Text style={styles.modalText, {color: theme.secondaryColor}}>{userData.audioDescription} &&
15 <Icon onPress={() => Speak("Apenas pessoas com disponibilidade de {userData.userType === "donors" ? "entregar" : "buscar"}")
16 name="volume-high" size={20} color="#838045"/>
17 </Text>
18 </View>
19
20 <List.Accordion
21 style={{expanded ? {backgroundColor: theme.thirdColor} : {backgroundColor: theme.backgroundColor}}
22 title={locationFilter}
23 titleStyle={{expanded ? {color: "#838045"} : {color: theme.secondaryColor}}
24 expanded={expanded}
25 onPress={() => {setExpanded(expanded); userData.audioDescription && 'expanded &&
26 Speak("Aqui você poderá escolher quais usuários serão mostrados, para saber mais sobre a opção aperte e segure-a")
27 left={props => <List.Icon color={expanded ? "#838045" : theme.secondaryColor} icon="folder" />}
28 >
29 <List.Item titleStyle={{color: theme.secondaryColor}}
30 onLongPress={() => userData.audioDescription && Speak("Mostrar todos")} title="Mostrar todos"
31 onPress={() => setLocationFilter("Mostrar todos")}/>
32 <List.Item titleStyle={{color: theme.secondaryColor}}
33 onLongPress={() => userData.audioDescription && Speak("Mostrar somente usuários do seu bairro")}
34 title="Mesmo bairro" onPress={() => setLocationFilter("Mesmo bairro")}/>
35 <List.Item titleStyle={{color: theme.secondaryColor}}
36 onLongPress={() => userData.audioDescription && Speak("Mostrar somente usuários da sua cidade")}
37 title="Mesma cidade" onPress={() => setLocationFilter("Mesma cidade")}/>
38 <List.Item titleStyle={{color: theme.secondaryColor}}
39 onLongPress={() => userData.audioDescription && Speak("Mostrar somente usuários do seu estado")}
40 title="Mesmo estado" onPress={() => setLocationFilter("Mesmo estado")}/>
41 </List.Accordion>
42
43 <View style={{height: 35, width: "45%", borderRadius: 5, overflow: 'hidden', marginLeft: "auto", marginTop: 20}}>
44 <TouchableNativeFeedback onPress={() => filter()}>
45 <View style={{backgroundColor: "#838045", justifyContent: 'center', alignItems: 'center',
46 height: '100%', borderRadius: 5}}>
47 <Text style={styles.modalTextButtons, {color: "white"}}>Aplicar</Text>
48 </View>
49 </TouchableNativeFeedback>
50 </View>
51 </Modal>
52 </View>
53 </Portal>
54 </Provider>
55 </>
56 )

```

Fonte: Autoria própria, 2022

2.7.13 Código da tela Configurações do aplicativo

Figura 32 – Código da Tela de Configurações do Aplicativo

```

1  return(
2      <Provider>
3      <Portal>
4      <View style={{styles.container, {backgroundColor: theme.backgroundColor}}>
5      <StatusBar style={theme.statusBar} translucent backgroundColor="transparent"/>
6      <TouchableOpacity onPress={() => navigation.toggleDrawer()}>
7
8      <Text style={{styles.textTitle, {color: theme.secondaryColor}}>Configurações{"\t"}{userData.audioDescription} && <Icon2
9      onPress={() => Speak("Tela de configurações")}
10     name="volume-high" size={20} color="#03A045"/>
11     </Text>
12
13     <View>
14     <View style={{width: "100%"}}>
15     <View style={{width: '85%', flexDirection: 'row', alignItems: 'center', marginTop: 30, marginLeft: 'auto', marginRight: 'auto'}}>
16     <View style={styles.donateView}>
17     <Icon name="contact-support" size={35} color="#03A045"/>
18     </View>
19     <Text style={{styles.textSingle, {color: theme.secondaryColor}}>Quer relatar algum problema, tem alguma dúvida ou sugestão?
20     Entre em contato com o nosso suporte clicando no botão abaixo!{"\t"}{userData.audioDescription} && <Icon2
21     onPress={() => Speak("Quer relatar algum problema, tem alguma dúvida ou sugestão? Entre em contato com o nosso suporte clicando no botão abaixo!")>
22     name="volume-high" size={20} color="#03A045"/>
23     </Text>
24     </View>
25
26     <View style={{width: '60%', height: 55, borderRadius: 10, overflow: 'hidden', marginTop: 20, marginLeft: 'auto', marginRight: 'auto'}}>
27     <TouchableNativeFeedback onPress={() => Linking.openURL("https://forms.gle/HN5EzDdY7ahMGTRz6")}>
28     <View style={styles.background_button}>
29     <Text style={{styles.textbutton, {color: '#03A045'}}>Entrar em contato</Text>
30     </View>
31     </TouchableNativeFeedback>
32     </View>
33 </View>
34 </View>
35
36 <View style={{width: '80%', marginTop: 40, marginLeft: 'auto', marginRight: 'auto'}}>
37 <View style={styles.options}>
38 <Text style={{styles.optionText, {color: theme.secondaryColor}}>Tema escuro{"\t"}{userData.audioDescription} && <Icon2
39 onPress={() => Speak("Tema escuro")}
40 name="volume-high" size={20} color="#03A045"/>
41 </Text>
42 <View style={{marginLeft: "auto", marginTop: 4}}>
43 <ToggleSwitch
44 id={donTheme}
45 onColor="#03A045"
46 offColor="#D7D7D7"
47 size="small"
48 onToggle={() => toggleThemeSwitch()}
49 />
50 </View>
51 </View>
52
53 <View style={styles.options}>
54 <Text style={{styles.optionText, {color: theme.secondaryColor}}>Continuar {userData.userType == "donors" ? "doando" : "como receptor"}{"\t"}{userData.audioDescription} && <Icon2
55 onPress={() => Speak("Continuar ${userData.userType == "donors" ? "doando" : "como receptor"}")}
56 name="volume-high" size={20} color="#03A045"/>
57 </Text>
58 <View style={{marginLeft: "auto", marginTop: 4}}>
59 <ToggleSwitch
60 id={isActiveUser}
61 onColor="#03A045"
62 offColor="#D7D7D7"
63 size="small"
64 onToggle={() => toggleActiveUserSwitch()}
65 />
66 </View>
67 </View>
68 </View>

```

Fonte: Autoria própria, 2022

2.7.14 Código de controle da navegação e dos sliders do aplicativo

Figura 33 – Código de Controle da Navegação e dos Sliders do Aplicativo

```

1
2
3 <View style={styles.options}>
4   <Text style={{styles.optionText, {color: theme.secondaryColor}}}>Audiodescrição{"\t"}{userData.audioDescription} && <Icon2
5     onPress={() => Speak("Audiodescrição")}
6     name="volume-high" size={26} color="#83A845"/>
7   </Text>
8   <View style={{marginLeft: "auto", marginTop: 4}}>
9     <ToggleSwitch
10      isOn={audioDescription}
11      onColor="#83A845"
12      offColor="#070707"
13      size="small"
14      onToggle={() => toggleAudioDescriptionSwitch()}
15    />
16  </View>
17 </View>
18
19 <View style={{flexDirection: 'row', width: '100%', marginTop: 25}}>
20   <View style={{height: 40, width: "45%", borderRadius: 5, overflow: 'hidden'}}>
21     <TouchableNativeFeedback onPress={() => setModalIsVisible(true)}>
22       <View style={{backgroundColor: "transparent", justifyContent: 'center', alignItems: 'center',
23         height: '100%', borderWidth: 1, borderColor: "#83A845", borderRadius: 6}}>
24         <Text style={styles.buttonsText}>Deletar conta</Text>
25       </View>
26     </TouchableNativeFeedback>
27   </View>
28
29   <View style={{height: 40, width: "45%", borderRadius: 5, overflow: 'hidden', marginLeft: "auto"}}>
30     <TouchableNativeFeedback onPress={() => logOutAccount()}>
31       <View style={{backgroundColor: "#83A845", justifyContent: 'center', alignItems: 'center',
32         height: '100%', borderRadius: 6}}>
33         <Text style={styles.buttonsText, {color: "white"}}>Sair</Text>
34       </View>
35     </TouchableNativeFeedback>
36   </View>
37 </View>
38
39 </View>
40
41 <Modal visible={modalIsVisible} contentContainerStyle={styles.modalStyle, {backgroundColor: theme.backgroundColor}}
42   onDismiss={() => {setModalIsVisible(false); alert("Sua conta não foi deletada");}}>
43   <Text style={styles.text, {fontSize: 18, marginBottom: 20, width: "80%", color: theme.secondaryColor}}>Para deletar sua conta,
44     insira sua senha abaixo!!</Text>
45
46   <View style={{width: "80%", marginLeft: "auto", marginRight: "auto"}}>
47     <Input background={theme.inputBackgroundColor} color={theme.textColor} borderColor={theme.borderColor}
48       label="Senha:" placeholder="Insira sua senha" secureText={true} value={password} onChangeText={setPassword}/>
49   </View>
50
51   <View style={{width: "80%", marginLeft: 'auto',
52     marginRight: 'auto', marginBottom: 30}}>
53     <ButtonMain label="Deletar conta" onPress={deleteAccount}/>
54   </View>
55 </Modal>
56 </Portal>
57 </Provider>
58 )

```

Fonte: Autoria própria, 2022

Figura 34 – Código de Controle da Navegação e dos Sliders do Aplicativo

```

1 function AppDrawer() {
2   const [screenName, setScreenName] = useState("");
3   const { getUserData } = useContext(UserContext);
4   const { theme } = useContext(ThemeContext);
5
6   useEffect(() => {
7     getUserData();
8   }, [])
9
10  firebase_auth().onAuthStateChanged((user) => {
11    const database = firebase.firestore();
12
13    if (user) {
14      database.collection("users").doc(user.uid).get().then((doc) => {
15        if (doc.data())
16          setScreenName(doc.data().userType === "donors" ? "Ajude alguém" : "Doadores");
17      });
18    }
19  });
20
21  return (
22    <Drawer.Navigator drawerContent={props => <CustomDrawer {...props} />} initialRouteName="HomeScreen"
23    screenOptions={{headerShown: false, drawerStyle: {backgroundColor: theme.backgroundColor}}} />
24    <Drawer.Screen name="HomeScreen" component={Home}
25    options={{drawerLabelStyle: {fontFamily: "Poppins_400Regular"}, drawerActiveBackgroundColor: theme.thirdColor, drawerActiveTintColor: "#83A845",
26    drawerLabel: "Início", drawerInactiveTintColor: theme.secondaryColor, drawerIcon: ({focused}) => <Icon2 name="home" size={25} color={focused ? "#83A845" : theme.secondaryColor} />}} />
27    <Drawer.Screen name="ProfileScreen" component={Profile}
28    options={{drawerLabelStyle: {fontFamily: "Poppins_400Regular"}, drawerActiveBackgroundColor: theme.thirdColor, drawerActiveTintColor: "#83A845",
29    drawerLabel: "Meu perfil", drawerInactiveTintColor: theme.secondaryColor, drawerIcon: ({focused}) => <Icon2 name="user" size={25} color={focused ? "#83A845" : theme.secondaryColor} />}} />
30    <Drawer.Screen name="FarmersAndCompaniesScreen" component={FarmersAndCompanies}
31    options={{drawerLabelStyle: {fontFamily: "Poppins_400Regular"}, drawerActiveBackgroundColor: theme.thirdColor, drawerInactiveTintColor: theme.secondaryColor, drawerActiveTintColor: "#83A845",
32    drawerLabel: screenName, drawerIcon: ({focused}) => <Icon3 name={screenName === "Doadores" ? "hand-holding-heart" : "hands-helping"}
33    size={22} color={focused ? "#83A845" : theme.secondaryColor} />}} />
34    <Drawer.Screen name="PlacesAndSitesScreen" component={PlacesAndSites}
35    options={{drawerLabelStyle: {fontFamily: "Poppins_400Regular"}, drawerActiveBackgroundColor: theme.thirdColor, drawerInactiveTintColor: theme.secondaryColor, drawerActiveTintColor: "#83A845",
36    drawerLabel: "LNC's", drawerIcon: ({focused}) => <Icon2 name="map" size={25} color={focused ? "#83A845" : theme.secondaryColor} />}} />
37    <Drawer.Screen name="Configurations" component={Configuration}
38    options={{drawerLabelStyle: {fontFamily: "Poppins_400Regular"}, drawerActiveBackgroundColor: theme.thirdColor, drawerInactiveTintColor: theme.secondaryColor, drawerActiveTintColor: "#83A845",
39    drawerLabel: "Configurações", drawerIcon: ({focused}) => <Icon2 name="settings" size={25} color={focused ? "#83A845" : theme.secondaryColor} />}} />
40  </Drawer.Navigator>
41  )
42  }

```

Fonte: Autoria própria, 2022

Figura 35 – Código de Controle da Navegação e dos Sliders do Aplicativo

```

1  export default function App() {
2    let [fontLoaded] = useFonts({
3      Poppins_600SemiBold, Poppins_700Bold, Poppins_400Regular
4    });
5
6    const { userData } = useContext(UserContext);
7    const [isLoading, setIsLoading] = useState(true);
8    const [user, setUser] = useState();
9    const [viewed, setViewed] = useState(false);
10   const [dotColor, setDotColor] = useState("white");
11   const [introductionWasViewed, setIntroductionViewed] = useState(false);
12   const [audioDescriptionIsActive, setAudioDescriptionIsActive] = useState(true);
13
14   firebase.auth().onAuthStateChanged((user) => {
15     if(user) {
16       setViewed(true);
17       setUser(user);
18     }
19
20     setIsLoading(false);
21   });
22
23   const renderNextButton = () => {
24     return (
25       <View style={[styles.background_next, {borderColor: dotColor}]}>
26         <Icon name="chevron-right" size={40} color={dotColor}/>
27       </View>
28     );
29   }
30
31   const renderDoneButton = () => {
32     return (
33       <View style={[styles.background_next, {borderColor: dotColor, width: 100}]}>
34         <Text style={[styles.text_slides, {marginTop: 5, fontSize: 15, width: 'auto'}]}>Vamos lá</Text>
35       </View>
36     );
37   }
38
39   renderItem = ({item}) => {
40     return (
41       <View style={{backgroundColor: item.backgroundColor, height: height, justifyContent: 'center'}}>
42         <StatusBar barStyle="dark-content" translucent backgroundColor="transparent"/>
43         <View style={{marginBottom: 70, alignItems: 'center'}}>
44           <Text style={[styles.title_slides, item.key === "two" ? {color: "#3F3D56"} : {}]}>
45             {item.title}{"\t"}{audioDescriptionIsActive} && <Icon onPress={() => Speak(item.title)} name="volume-high" size={28}
46             color={item.key === "two" ? "#3F3D56" : 'white'}/>
47           </Text>
48
49           <Image style={[styles.image_slides, item.key === "two" ?
50             {height: 252, width: 350} :
51             item.key === "three" ? {height: 300, width: 300} : {marginVertical: 70}]} source={item.image}/>
52
53           <Text style={[styles.text_slides, item.key === "two" ? {color: "#3F3D56", fontSize: 20} : {}]}>
54             {item.text}{"\t"}{audioDescriptionIsActive} && <Icon onPress={() => Speak(item.text)} name="volume-high" size={28}
55             color={item.key === "two" ? "#3F3D56" : 'white'}/>
56           </Text>
57
58           {item.key === "three" &&
59             <ToggleSwitch
60               trackOnStyle={{marginTop: 20}}
61               trackOffStyle={{marginTop: 20}}
62               isOn={audioDescriptionIsActive}
63               onColor="#00712F"
64               offColor="#D7D7D7"
65               size="small"
66               onToggle={() => setAudioDescriptionIsActive(!audioDescriptionIsActive)}
67             />
68         </View>
69       </View>
70     );
71   };
72 }
73 }

```

Fonte:

Autoria própria, 2022

Figura 36 – Código da Tela de Configurações do Aplicativo

```

1
2
3 <View style={styles.options}>
4   <Text style={[styles.optionText, {color: theme.secondaryColor}]}>Audiodescrição{"\t"}{userData.audioDescription} && <Icon2
5     onPress={() => Speak("Audiodescrição")}
6     name="volume-high" size={20} color="#03A045"/>
7   </Text>
8   <View style={{marginLeft: "auto", marginTop: 4}}>
9     <ToggleSwitch
10      isOn={audioDescription}
11      onColor="#03A045"
12      offColor="#070707"
13      size="small"
14      onToggle={() => toggleAudioDescriptionSwitch()}
15    />
16  </View>
17 </View>
18
19 <View style={{flexDirection: 'row', width: '100%', marginTop: 25}}>
20   <View style={{height: 40, width: "45%", borderRadius: 5, overflow: 'hidden'}}>
21     <TouchableNativeFeedback onPress={() => setModalIsVisible(true)}>
22       <View style={{backgroundColor: "transparent", justifyContent: 'center', alignItems: 'center',
23         height: '100%', borderWidth: 1, borderColor: "#03A045", borderRadius: 6}}>
24         <Text style={styles.buttonsText}>Deletar conta</Text>
25       </View>
26     </TouchableNativeFeedback>
27   </View>
28   <View style={{height: 40, width: "45%", borderRadius: 5, overflow: 'hidden', marginLeft: "auto"}}>
29     <TouchableNativeFeedback onPress={() => logOutAccount()}>
30       <View style={{backgroundColor: "#03A045", justifyContent: 'center', alignItems: 'center',
31         height: '100%', borderRadius: 6}}>
32         <Text style={[styles.buttonsText, {color: "white"}]}>Sair</Text>
33       </View>
34     </TouchableNativeFeedback>
35   </View>
36 </View>
37 </View>
38
39 </View>
40
41 <Modal visible={modalIsVisible} contentContainerStyle={[styles.modalStyle, {backgroundColor: theme.backgroundColor}]}
42   onDismiss={() => {setModalIsVisible(false); alert("Sua conta não foi deletada");}}>
43   <Text style={styles.text, {fontSize: 18, marginBottom: 20, width: "80%", color: theme.secondaryColor}}>Para deletar sua conta,
44     insira sua senha abaixo!!</Text>
45
46   <View style={{width: "80%", marginLeft: "auto", marginRight: "auto"}}>
47     <Input background={theme.inputBackgroundColor} color={theme.textColor} borderColor={theme.borderColor}
48       label="Senha:" placeholder="Insira sua senha" secureText={true} value={password} onChangeText={setPassword}/>
49   </View>
50
51   <View style={{width: "80%", marginLeft: 'auto',
52     marginRight: 'auto', marginBottom: 30}}>
53     <ButtonMain label="Deletar conta" onPress={deleteAccount}/>
54   </View>
55 </Modal>
56 </Portal>
57 </Provider>
58 )

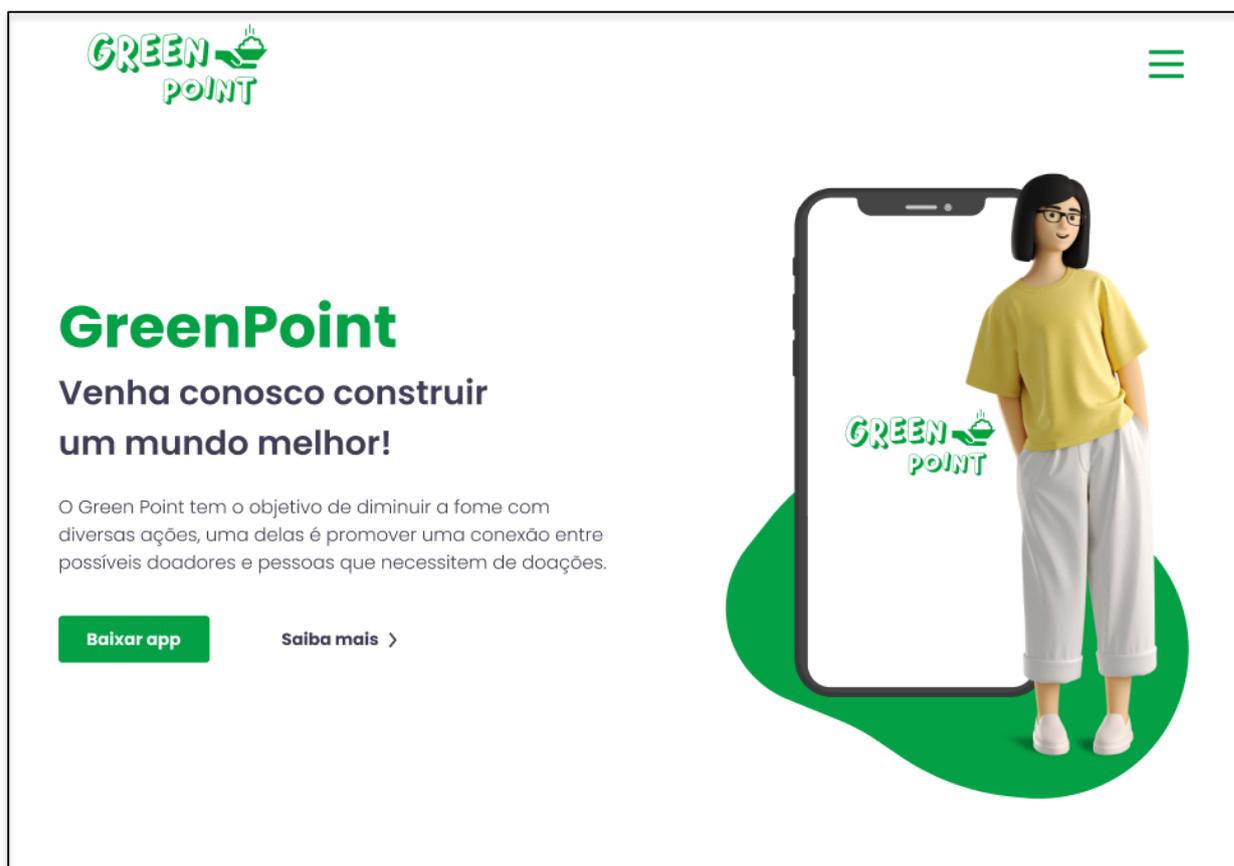
```

Fonte: Autoria própria, 2022

2.7.15 Página Inicial do site

A página inicial do site possui o "Menu", o botão de baixar o aplicativo para usar no celular e a opção "Saber mais", caso o usuário se interessar pelo projeto e queira ter mais informações sobre.

Figura 37 – Página Inicial do Site

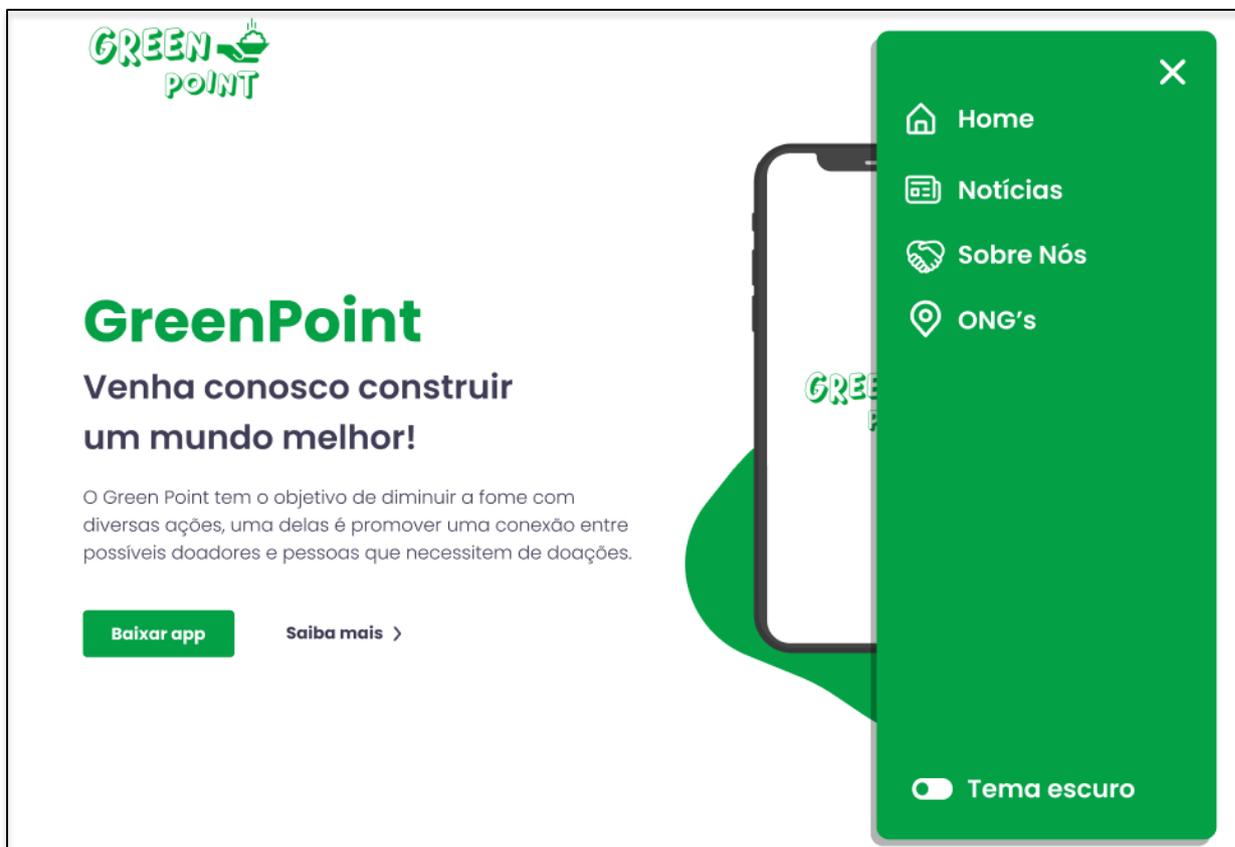


Fonte: Autoria própria, 2022

2.7.16 Menu do site

O "Menu" do site mostra as opções que o usuário tem para navegar pelo site: "Home" (página inicial), "Notícias", "Sobre Nós" e "ONGs", além disso, há a possibilidade de ativar ou desativar o modo escuro.

Figura 38 – Menu do Site

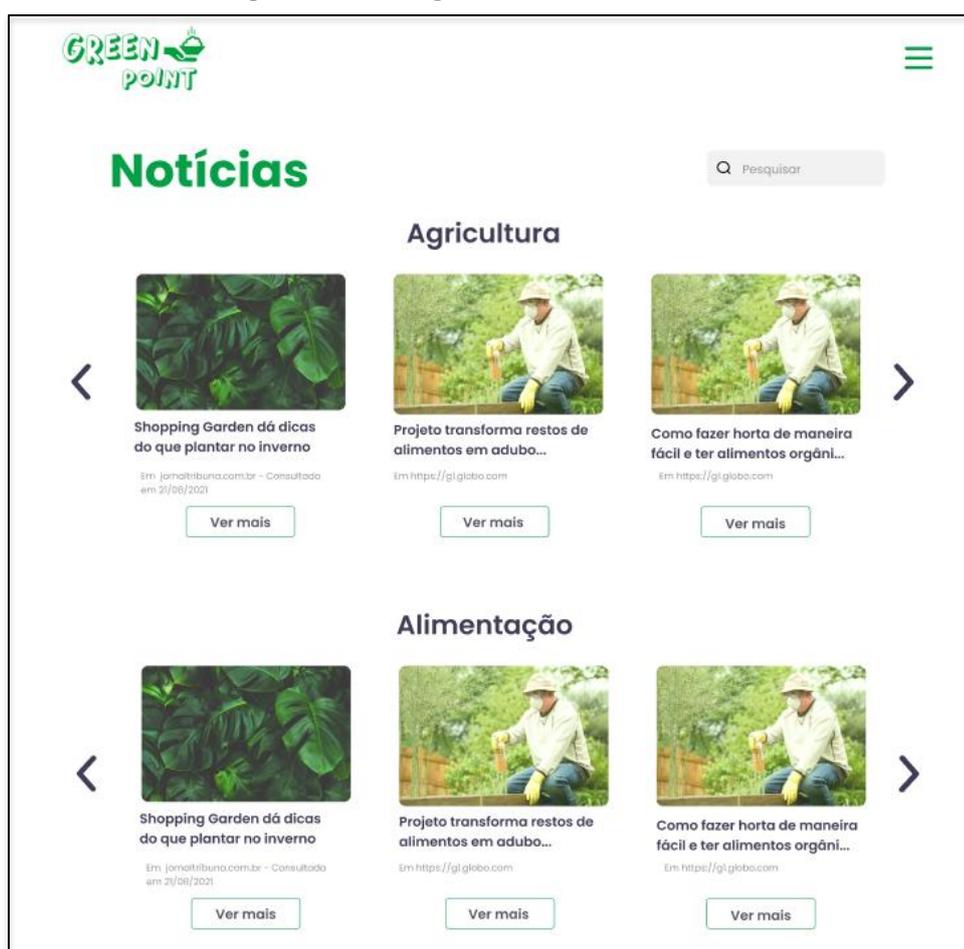


Fonte: Autoria própria, 2022

2.7.17 Página Notícias do site

A primeira página que o usuário será direcionado quando entrar no site, é a página de notícias. Nela haverá notícias sobre agricultura, alimentação e sustentabilidade. Na barra de pesquisa poderão ser escritas palavras-chave para achar determinadas notícias e, também, terá um filtro, no qual a pessoa poderá escolher os tipos de notícias que quer ver. No canto superior direito terá o "Menu" para navegar em outras páginas do site.

Figura 39 – Página de Notícias do Site



Sustentabilidade



Shopping Garden dá dicas do que plantar no inverno

Em jornaltribuna.com.br - Consultado em 2/06/2022

[Ver mais](#)



Projeto transforma restos de alimentos em adubo...

Em <https://g1.globo.com>

[Ver mais](#)



Como fazer horta de maneira fácil e ter alimentos orgâni...

Em <https://g1.globo.com>

[Ver mais](#)

Sobre Nós

O GreenPoint é um projeto sem fins lucrativos, que visa reduzir o número de pessoas em situação de vulnerabilidade alimentar. O nosso objetivo é divulgar e facilitar a comunicação entre possíveis doadores e pessoas que necessitam de doações, assim, contribuindo para a redução da fome.



Contate-nos

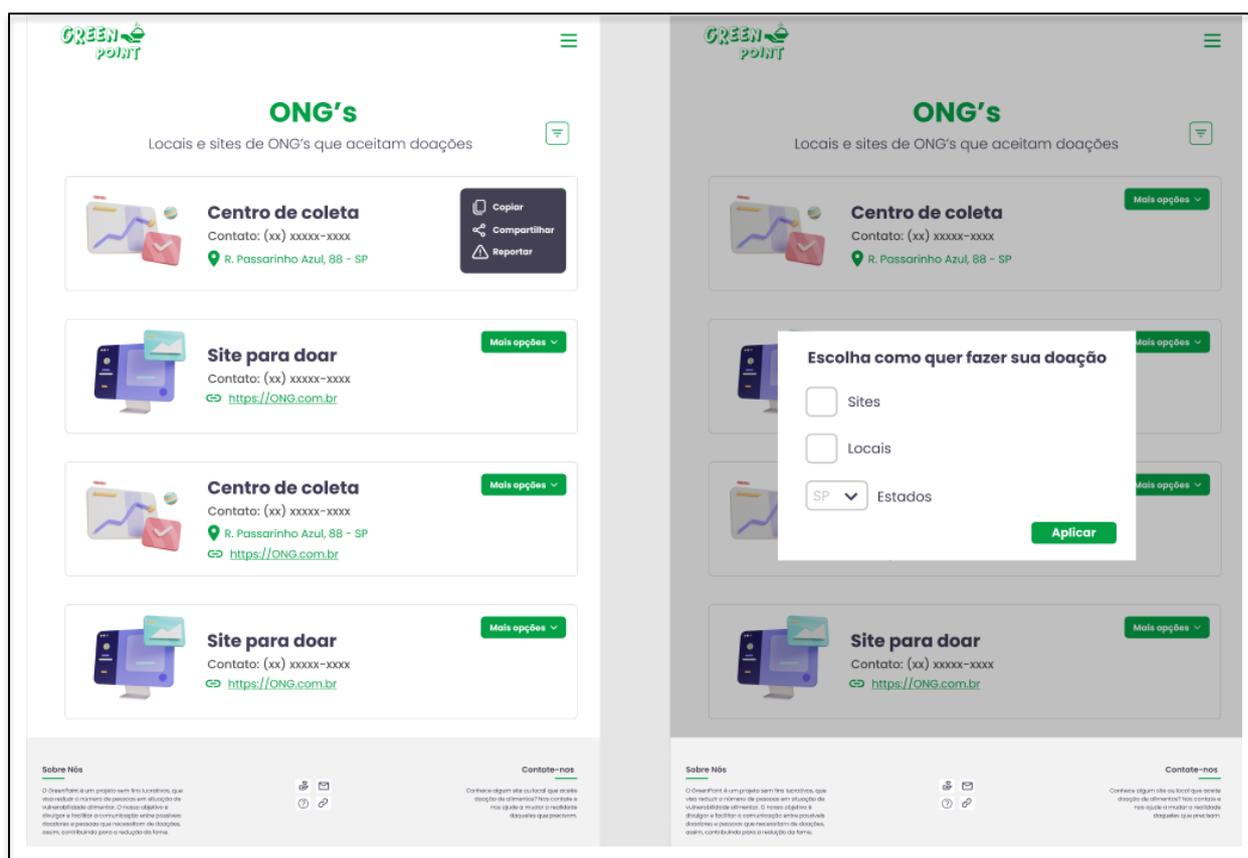
Conhece algum site ou local que aceite doação de alimentos? Nos contate e nos ajude a mudar a realidade daqueles que precisam.

Fonte: Autoria própria, 2022

2.7.18 Página ONGs do site

Na página de ONGs haverá vários perfis de ONGs físicas ou que possuem um site (os ícones de cada um serão diferenciados), seus nomes e endereços ou links. Haverá um filtro para que o usuário encontre somente ONGs físicas (poderá ser filtrado por estado) ou sites. Será possível clicar nos três pontos do perfil e o usuário poderá copiar o link/endereço, compartilhar o perfil ou reportar.

Figura 40 – Página de ONGs do Site



Fonte: Autoria própria, 2022

2.7.19 Página Sobre Nós do site

A página Sobre Nós, será possível ver sobre o que é o projeto GreenPoint, seu objetivo e, caso o usuário queira, poderá contribuir com o projeto ao clicar no botão "Contribuir". No fim da página há os perfis dos desenvolvedores e é possível entrar em contato com algum deles clicando em sua foto ou nome.

Figura 41 – Página Sobre Nós do Site



Time que torna possível a luta contra a fome



Alice Chinoque



Carolina Fernandes



Diego de Sousa



Isabella Gomes



Nicollas Isaac



Sarah Macedo

Acesse nossos [termos de uso](#).

Sobre Nós

O GreenPoint é um projeto sem fins lucrativos, que visa reduzir o número de pessoas em situação de vulnerabilidade alimentar. O nosso objetivo é divulgar e facilitar a comunicação entre possíveis doadores e pessoas que necessitam de doações, assim, contribuindo para a redução da fome.



Contate-nos

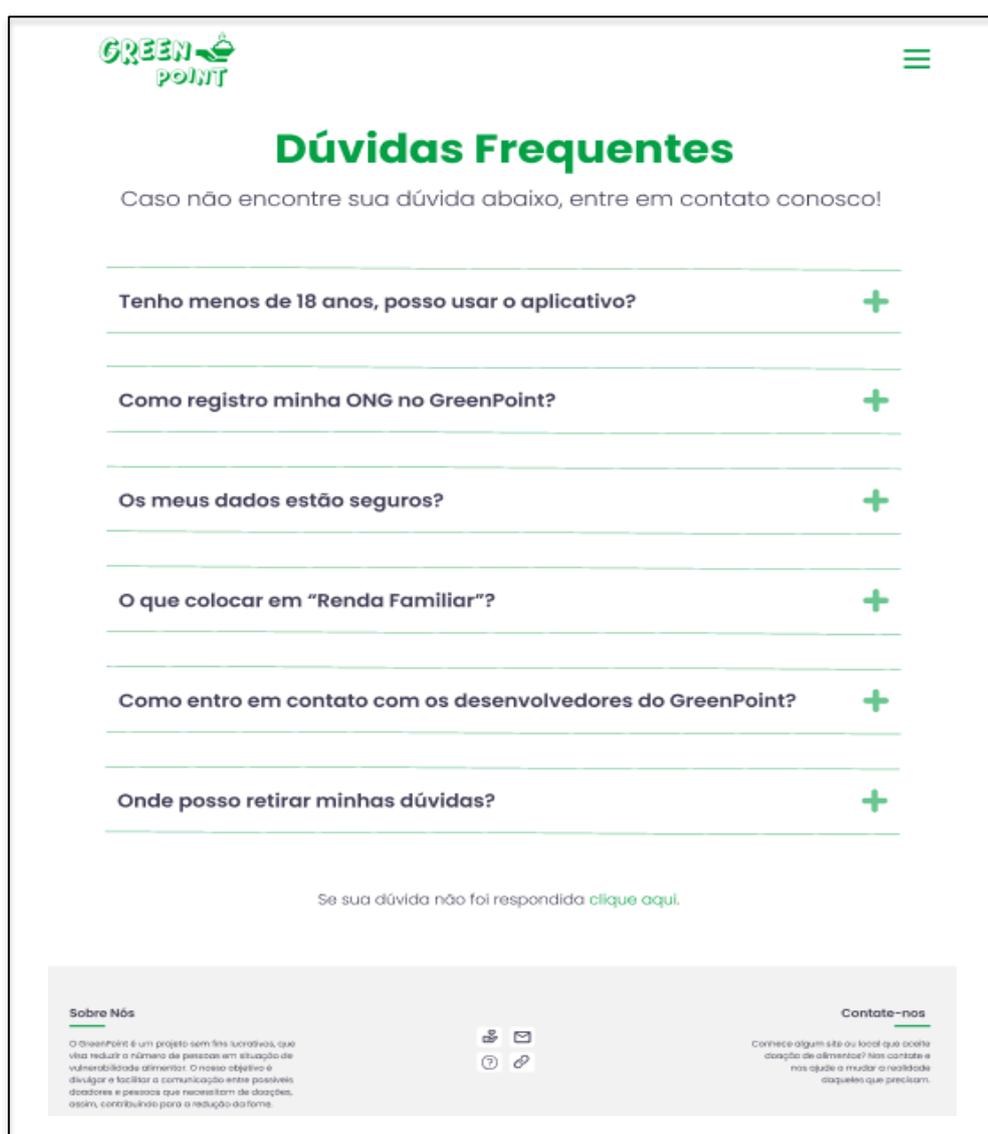
Conhece algum site ou local que aceite doação de alimentos? Nos contate e nos ajude a mudar a realidade daqueles que precisam.

Fonte: Autoria própria, 2022

2.7.20 Página Dúvidas Frequentes do site

Na página de dúvidas, há as possíveis dúvidas que o usuário pode ter ao utilizar o site ou aplicativo. Ao clicar no sinal "+" (mais) aparecerá as respostas das perguntas feitas. Se a dúvida não for respondida, o usuário poderá ir em "clique aqui" e, então, será direcionado ao formulário do site, no qual poderá fazer perguntas mais específicas sobre algo que queira.

Figura 42 – Página de Dúvidas Frequentes do Site



Fonte: Autoria própria, 2022

2.7.21 Página Formulário de contato do site

A página do formulário do site é utilizada pelo usuário para tirar dúvidas ou fazer perguntas mais específicas que não foram respondidas anteriormente. O usuário terá que preencher os campos de nome, e-mail e, então, digitar sua mensagem para poder enviá-la.

Figura 43 – Página do Formulário de Contato do Site

GREEN POINT

Formulário

Relate uma dúvida, sugestão, um problema ou erro do sistema.

Nome

insira seu nome

E-mail

insira seu e-mail

Mensagem

Escreva sua dúvida, problema, erro ou sugestão

Enviar

Sobre Nós

O GreenPoint é um projeto sem fins lucrativos, que visa reduzir o número de pessoas em situação de vulnerabilidade alimentar. O nosso objetivo é divulgar e facilitar a comunicação entre possíveis doadores e pessoas que necessitam de doações, assim, contribuindo para a redução da fome.

Contate-nos

Conhece algum site ou local que aceite doação de alimentos? Nos contate e nos ajude a mudar a realidade daqueles que precisam.

Fonte: Autoria própria, 2022

2.7.22 Página Contribua do site

A página de contribuição aparecerá quando o usuário tiver clicado no botão "Contribuir" da página Sobre Nós. Nessa página, será possível saber mais sobre como contribuir com o projeto e terá a chave PIX para que o usuário contribua com um valor que preferir para ajudar o projeto GreenPoint a continuar ajudando pessoas.

Figura 44 – Página Contribua do Site



Nosso PIX

Doe por nossa chave PIX: greenpoint-org@gmail.com



Caso deseje fazer uma doação financeira, clique no botão abaixo e envie o valor através do PIX.

[Copie nossa chave PIX](#)

Acesse nossas [diretrizes](#).

Sobre Nós

O GreenPoint é um projeto sem fins lucrativos, que visa reduzir o número de pessoas em situação de vulnerabilidade alimentar. O nosso objetivo é divulgar e facilitar a comunicação entre possíveis doadores e pessoas que necessitam de doações, assim, contribuindo para a redução da fome.

Contate-nos

Conhece algum site ou local que aceite doação de alimentos? Nos contate e nos ajude a mudar a realidade daqueles que precisam.



Fonte: Autoria própria, 2022

2.7.23 Página Formulário de registro do site

A outra página de formulário do site é destinada à ONGs e/ou empresas que queiram ter parceria com o projeto GreenPoint. Nela haverá os campos de nome, e-mail, CNPJ, endereço ou site de contato e a mensagem que a instituição deseja enviar.

Figura 45 – Página do Formulário de Registro do Site



The image shows a web form for registration on the GreenPoint website. At the top left is the GreenPoint logo, and at the top right is a hamburger menu icon. The main heading is 'Formulário' in green, followed by the sub-heading 'Registre sua empresa/ONG e seja nosso parceiro!'. The form consists of several input fields: 'Nome da empresa ou ONG' with a placeholder 'insira seu nome'; 'E-mail' with a placeholder 'insira seu e-mail'; 'CNPJ' with a placeholder 'insira seu CNPJ'; 'Endereço ou site de contato' with a placeholder 'insira seu endereço ou site'; and a larger text area for 'Mensagem' with a placeholder 'Escreva sobre as ideias e ações realizadas pela sua empresa/ONG'. A green 'Enviar' button is positioned below the message field. At the bottom, there is a footer section with 'Sobre Nós' and 'Contate-nos' links, social media icons, and contact information.

GREEN POINT

Formulário

Registre sua empresa/ONG e seja nosso parceiro!

Nome da empresa ou ONG

E-mail

CNPJ

Endereço ou site de contato

Mensagem

Enviar

Sobre Nós

O GreenPoint é um projeto sem fins lucrativos, que visa reduzir o número de pessoas em situação de vulnerabilidade alimentar. O nosso objetivo é divulgar e facilitar a comunicação entre possíveis doadores e pessoas que necessitam de doações, assim, contribuindo para a redução da fome.

Contate-nos

Conhece algum site ou local que aceite doação de alimentos? Nos contate e nos ajude a mudar a realidade daqueles que precisam.

Fonte: Autoria própria, 2022

2.7.24 Página Termos de uso

A página de termos de uso apresenta as responsabilidades que os usuários devem ter ao utilizar o aplicativo/site GreenPoint. Nela, há regras para utilização do software, proibições, proteção de dados etc., para não haver mal entendidos diante dos direitos e deveres do usuário.

Figura 46 – Página de Termos de Uso (1)



GREEN POINT

Termos de Uso

Sobre nossos Serviços

- **Princípios de segurança e privacidade.** Desde que fundamos o GreenPoint, nós desenvolvemos nossos Serviços com base em sólidos princípios de segurança e privacidade.
- **Conectar você a outras pessoas.** Possibilitamos e sempre tentamos aprimorar maneiras para que você se comunique com outros usuários do GreenPoint, entrando em contato com a pessoa desejada, quando você desejar.
- **Comunicação com Empresas.** Nós possibilitamos e sempre tentamos aprimorar maneiras para que usuários, empresas e outras organizações se comuniquem entre si por meio dos nossos Serviços para fazer atualizações e marketing.
- **Proteção, segurança e integridade.** Trabalhamos para garantir a proteção, segurança e integridade dos nossos Serviços, inclusive para lidar adequadamente com pessoas abusivas e violações às nossas normas. Nos dedicamos para proibir o uso indevido de nossos Serviços, como condutas nocivas, as violações de nossos Termos e políticas e abordamos situações em que podemos ajudar a oferecer suporte ou proteger nossa comunidade. Se soubermos de pessoas ou atividades assim, tomaremos as medidas necessárias, seja removendo tais pessoas ou atividades, ou entrando em contato com as autoridades de aplicação da lei.

Fonte: Autoria própria, 2022

Figura 47 – Página de Termos de Uso (2)

Política de Privacidade e dados do usuário

O GreenPoint se preocupa com sua privacidade. A Política de Privacidade do GreenPoint descreve nossas práticas relativas a dados (e também mensagens), inclusive os tipos de informações que recebemos e coletamos, como elas são usadas e divulgadas e seus direitos associados ao tratamento de suas informações.

Nossos Termos e Políticas. Nossos Serviços devem ser usados de acordo com nossos Termos e políticas. Se você violar nossos Termos ou políticas, podemos tomar medidas em relação a sua conta, como desativá-la ou suspendê-la. Se fizermos isso, você concorda em não criar outra conta sem nossa permissão.

Uso lícito e aceitável. Nossos Serviços devem ser acessados e utilizados somente para fins lícitos, autorizados e aceitáveis. Você não usará (ou ajudará outras pessoas a usar) nossos Serviços: (a) de forma a violar, apropriar-se indevidamente ou infringir direitos do GreenPoint, dos nossos usuários ou de terceiros, inclusive direitos de privacidade, de publicidade, de propriedade intelectual ou outros direitos de propriedade; (b) de forma ilícita, obscena, difamatória, ameaçadora, intimidadora, assediante, odiosa, ofensiva em termos raciais ou étnicos, ou que instigue ou encoraje condutas que sejam ilícitas ou inadequadas, como a incitação a crimes violentos, a exploração de crianças ou outras pessoas, a ação de colocá-las em perigo, ou a coordenação de danos reais; (c) envolvendo declarações falsas, incorretas ou enganosas; (d) para se passar por outra pessoa; (e) para enviar comunicações ilícitas ou não permitidas, como mensagens em massa, mensagens automáticas, ligações automáticas e afins; ou (f) de forma a envolver o uso não pessoal dos nossos Serviços, a menos que esteja autorizado por nós.

Fonte: Autoria própria, 2022

Figura 48 – Página de Termos de Uso (3)

Prejuízo ao GreenPoint ou a nossos usuários. Você não está autorizado, diretamente, indiretamente, por meios automatizados ou quaisquer outros a acessar, usar, copiar, adaptar, modificar, elaborar trabalhos derivados, distribuir, licenciar, sublicenciar, transferir, executar ou de qualquer forma explorar (ou prestar auxílio para que alguém o faça) nossos Serviços de maneira não permitida ou autorizada, ou de forma a prejudicar ou onerar a nós, nossos Serviços, sistemas, usuários ou terceiros, inclusive, seja diretamente ou mediante automação: (a) fazer engenharia reversa, alterar, modificar, criar trabalhos derivados, descompilar ou extrair códigos dos nossos Serviços; (b) enviar, armazenar ou transmitir vírus ou outros códigos nocivos usando nossos Serviços; (c) obter ou tentar obter acesso não autorizado a nossos Serviços ou sistemas; (d) interferir ou interromper a segurança, a proteção, a confidencialidade, a integridade, a disponibilidade ou o desempenho de nossos Serviços; (e) criar contas por nossos Serviços usando meios não autorizados ou automatizados; (f) coletar informações de ou sobre nossos usuários de maneira não autorizada; (g) vender, revender, alugar ou cobrar por nossos Serviços ou por dados obtidos a partir dos nossos Serviços ou de nós de forma não autorizada; ou (h) distribuir ou disponibilizar nossos Serviços em rede para ser usado por vários dispositivos ao mesmo tempo, exceto conforme autorizado em ferramentas expressamente fornecidas por meio de nossos Serviços; (i) criar um software ou APIs que desempenham a mesma função que nossos Serviços e oferecê-los a terceiros de maneira não autorizada; ou (j) usar indevidamente canais de denúncia, como enviar denúncias ou contestações fraudulentas ou infundadas.

Manutenção da segurança de sua conta. Você é responsável por manter seu dispositivo e sua conta do GreenPoint protegidos e seguros e deve nos informar imediatamente quando houver uso não autorizado ou violação de segurança em sua conta ou em nossos Serviços.

Caso tiver dúvidas [clique aqui](#).

Sobre Nós

O GreenPoint é um projeto sem fins lucrativos, que visa reduzir o número de pessoas em situação de vulnerabilidade alimentar. Nosso objetivo é divulgar e facilitar a comunicação entre possíveis doadores e pessoas que necessitam de doações, assim, contribuindo para a redução da fome.



Contate-nos

Conhece algum site ou local que aceite doação de alimentos? Nos contate e nos ajude a mudar a realidade daqueles que precisam.

2.7.25 Código da página Inicial do site

Figura 49 – Código da Página Inicial do Site

```

1 <main>
2   <div class="ScreenMain">
3     <div class="TextInfo">
4       <h1 class="animate__animated animate__bounce">GreenPoint</h1>
5       <h2>Venha conosco construir um mundo melhor!</h2>
6       <p>O Green Point tem o objetivo de diminuir a fome com diversas ações,
7         uma delas é promover uma conexão entre possíveis doadores e pessoas que necessitem de doações.</p>
8     <div class="buttons">
9       <button>Baixar App</button>
10      <a href="/aboutus/">Saiba mais
11        <span class="iconify" data-icon="eva:arrow-ios-forward-fill" data-height="22"></span>
12      </a>
13    </div>
14  </div>
15  <div class="ImgHome">
16    
17  </div>
18 </div>
19 </main>

```

Fonte: Autoria própria, 2022

2.7.26 Código da página Notícias do site

Figura 50 – Código da Página de Notícias do Site

```

1 <section id="searchContainer" class="search">
2   <div id="container_search" class="container_carrocel">
3     <div id="items_carrocel_model_search" class="items_carrocel items_carrocel_search">
4       <div id="container_ItemsSearch" class="container_ItemsCarrocel container_ItemsCarrocel_search">
5         
6         <h2>Shopping Garden dá dicas do que plantar no inverno</h2>
7         <p>Em jornaltribuna.com.br - Consultado em 21/08/2021</p>
8         <a>Ver mais</a>
9       </div>
10    </div>
11  </div>
12 </section>

```

Fonte: Autoria própria, 2022

2.7.27 Código da página ONGs do site

Figura 51 – Código da Página de ONGs do Site

```
1  function switchVerMais(numItem) {
2      var ver = document.querySelectorAll('.ver');
3
4      if(lastNumItem != null && numItem != lastNumItem) {
5          ver[lastNumItem].classList.remove("verIsOpen");
6          ver[lastNumItem].classList.add("verIsClose");
7
8          ver[numItem].classList.add("verIsOpen");
9          ver[numItem].classList.remove("verIsClose");
10         verIsOpen = true;
11         lastNumItem = numItem;
12         return;
13     }
14
15     if(!verIsOpen)
16     {
17         ver[numItem].classList.add("verIsOpen");
18         ver[numItem].classList.remove("verIsClose");
19         verIsOpen = true;
20         lastNumItem = numItem;
21         return;
22     }
23
24     if(verIsOpen) {
25         ver[numItem].classList.remove("verIsOpen");
26         ver[numItem].classList.add("verIsClose");
27         verIsOpen = false;
28         return;
29     }
30 }
```

Fonte: Autoria própria, 2022

2.7.28 Código da página Sobre Nós do site

Figura 52 – Código da Página Sobre Nós do Site

```

1 <section id="Team">
2   <h1>Time que torna a luta contra a fome possível</h1>
3   <div class="team1">
4     <div class="dev-person">
5       <a href="https://www.linkedin.com/in/alice-chinoque-377b661b6/" target="_blank">
6       <p>Alice Chinoque</p>
7     </a>
8   </div>
9   <div class="dev-person">
10    <a href="https://www.linkedin.com/in/carolina-fernandes-23b717254/">
11    <p>Carolina Fernandes</p>
12  </a>
13 </div>
14 <div class="dev-person">
15   <a href="https://github.com/DiegoSousa-11" target="_blank">
16   <p>Diego Sousa</p>
17 </a>
18 </div>
19 </div>
20 <div class="team2">
21   <div class="dev-person">
22     <a href="https://www.linkedin.com/in/isabella-gomes-a7665a237/" target="_blank">
23     <p>Isabella Gomes</p>
24   </a>
25 </div>
26 <div class="dev-person">
27   <a href="https://www.github.com/nicollasisaac" target="_blank">
28   <p>Nicollas Isaac</p>
29 </a>
30 </div>
31 <div class="dev-person">
32   <a href="https://www.linkedin.com/in/sarah-macedo-895661237/" target="_blank">
33   <p>Sarah Macedo</p>
34 </a>
35 </div>
36 </div>
37 </section>

```

Fonte: Autoria própria, 2022

2.7.29 Código da página Dúvidas Frequentes do site

Figura 53 – Código da Página de Dúvidas Frequentes do Site

```

1 questions.map((itemQuestion, index) => {
2   let itemsQuestions = document.querySelector(".questionItem").cloneNode(true);
3
4   itemsQuestions.id = "questionItem";
5   itemsQuestions.querySelector(".TextQandA p").innerHTML = itemQuestion.question;
6   itemsQuestions.setAttribute("onclick", `showAnswer(${index})`);
7
8   document.querySelector("#Questions").append(itemsQuestions);
9 })
10
11 document.getElementById("questionItem").parentNode.removeChild(document.getElementById("questionItem"));
12

```

Fonte: Autoria própria, 2022

2.7.30 Código da página Contribua do site

Figura 54 – Código da Página Contribua do Site

```
1 function copy(){  
2     navigator.clipboard.writeText("org.greenpoint@gmail.com")  
3     alert("Chave PIX copiada")  
4 }
```

Fonte: Autoria própria, 2022

2.7.31 Código da página Termos de uso do site

Figura 55 – Código da Página de Termos de Uso do Site

```
1 #termos{  
2     width: 90%;  
3     margin: auto;  
4     text-align: justify;  
5     padding-bottom: 2.5rem;  
6 }  
7  
8 .aba h2{  
9     color: var(--green);  
10    margin-top: 2rem;  
11 }  
12  
13 .aba span{  
14    font-weight: bold;  
15    color: var(--dark-blue);  
16 }  
17
```

Fonte: Autoria própria, 2022

2.7.32 Código da página Formulário de contato do site

Figura 56 – Código da Página do Formulário de Contato do Site

```

1  var Name = document.getElementById(Nome)
2
3  function msg(){
4      alert("Sua mensagem enviada com sucesso, " + Name);
5  }

```

Fonte: Autoria própria, 2022

2.7.33 Código da página Formulário de registro do site

Figura 57 – Código da Página do Formulário de Registro do Site

```

1  <form action="https://formsubmit.co/org.greenpoint@gmail.com" method="POST">
2      <div class="name">
3          <p>Nome da empresa ou ONG</p>
4          <input type="text" name="Nome" placeholder="Insira seu nome" required>
5      </div>
6      <div class="email">
7          <p>Email</p>
8          <input type="email" name="Email" placeholder="Insira seu e-mail" required>
9      </div>
10     <div class="cnpj">
11         <p>CNPJ</p>
12         <input type="number" name="Cnpj" placeholder="Insira o CNPJ" required>
13     </div>
14     <div class="contact">
15         <p>Endereço ou site de contato</p>
16         <input type="text" name="Contact" placeholder="Insira seu endereço ou site de contato" required>
17     </div>
18     <div class="send">
19         <p>Mensagem</p>
20         <textarea name="message" id="message" rows="7" cols="15" placeholder="Conte um pouco sobre o seu projeto e sobre sua empresa/ong" required></textarea>
21     </div>
22     <div class="btn">
23         <input type="submit" value="Enviar" onClick="msg()">
24     </div>
25     <input type="hidden" name="_captcha" value="false">
26     <input type="hidden" name="_next" value="http://127.0.0.1:5500/RegisterForm/index.html">
27 </form>

```

Fonte: Autoria própria, 2022

2.8 Rotinas de testes do aplicativo

2.8.1 Testes realizados

Os testes feitos após a produção do aplicativo foram o teste funcional, que testa os requisitos funcionais, as funções e os casos de uso, o teste de integração que valida a integração, de diferentes módulos em conjunto e identifica os bugs e problemas relacionados a eles e, por fim, o teste de ponta a ponta que garante que o comportamento completo do sistema esteja de acordo com o planejado e simula a atividade do usuário com o programa final em um ambiente parecido com o de produção.

2.8.2 Tabelas de testes da tela de Sliders

Contador:	001
Localização:	Tela “ <i>Sliders</i> ”.
Criticidade:	Média.
Objetivo do Teste:	Verificar o comportamento ao realizar a mudança entre as telas.
Caso de teste:	Testar os botões de mudança de tela.
Pré-Condição:	1. Acessar o aplicativo pela primeira vez.
Procedimento:	1. Acessar o aplicativo; 2. Apertar no botão para que a passagem de telas ocorra; 3. Realizar o deslizamento entre as telas.
Resultado Esperado:	1. O sistema deverá realizar a passagem dos <i>sliders</i> .
Resultado Validado:	O <i>software</i> funcionou como esperado.

Contador:	002
Localização:	Tela " <i>Sliders</i> ".
Criticidade:	Média.
Objetivo do Teste:	Verificar o comportamento ao habilitar/desabilitar a audiodescrição.
Caso de teste:	Testar o botão que muda a situação da audiodescrição.
Pré-Condição:	1. Acessar a última tela de <i>sliders</i> .
Procedimento:	1. Acessar o aplicativo; 2. Deslizar os sliders até a última tela; 3. Desativar a audiodescrição.
Resultado Esperado:	1. O sistema deverá desabilitar a audiodescrição.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

2.8.3 Tabela de testes da tela Inicial de Login/Cadastro

Contador:	001
Localização:	Tela inicial "Login/Cadastro".
Criticidade:	Alta.
Objetivo do Teste:	Verificar o comportamento ao apertar nos botões e analisar se eles redirecionam o usuário para suas respectivas telas.
Caso de teste:	Testar se os botões de <i>login</i> /cadastro cumprem as funções esperadas.
Pré-Condição:	1. Acessar a tela inicial.
Procedimento:	1. Acessar a tela de <i>login</i> /cadastro; 2. Apertar no botão de <i>login</i> ; 3. Apertar no botão de cadastro.
Resultado Esperado:	1. O sistema deverá redirecionar o usuário para a tela correta.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

2.8.4 Tabelas de testes da tela Login

Contador:	001
Localização:	Tela " <i>Login</i> ".
Criticidade:	Alta.
Objetivo do Teste:	Verificar se o <i>software</i> realiza o <i>login</i> corretamente.
Caso de teste:	Testar se o usuário consegue realizar o <i>login</i> .
Pré-Condição:	1. Acessar a tela de <i>login</i> .
Procedimento:	1. Acessar a tela de <i>login</i> ; 2. Preencher os campos com <i>email</i> e senha válidos; 3. Apertar no botão "Entrar".
Resultado Esperado:	1. O sistema deverá efetuar o <i>login</i> do usuário.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	002
Localização:	Tela “Login”.
Criticidade:	Alta.
Objetivo do Teste:	Verificar se o <i>software</i> realiza o <i>login</i> corretamente.
Caso de teste:	Testar se o usuário consegue realizar o <i>login</i> com <i>email</i> e senha inválidos.
Pré-Condição:	1. Acessar a tela de <i>login</i> .
Procedimento:	1. Acessar a tela de <i>login</i> ; 2. Preencher os campos com <i>email</i> e senha inválidos; 3. Apertar no botão “Entrar”.
Resultado Esperado:	1. O sistema não deverá efetuar o <i>login</i> do usuário.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	003
Localização:	Tela “ <i>Login</i> ”.
Criticidade:	Alta.
Objetivo do Teste:	Verificar se o <i>software</i> fornece a possibilidade de recuperação de senha corretamente.
Caso de teste:	Testar se o usuário recebe um email para recuperar a sua senha.
Pré-Condição:	1. Acessar a tela de <i>login</i> .
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela de <i>login</i>; 2. Apertar a opção “Esqueceu sua senha?”; 3. Inserir o <i>email</i> utilizado no cadastro da conta; 4. Pressionar o botão “Próximo”.
Resultado Esperado:	1. O sistema deverá enviar um email contendo as instruções para o usuário realizar a recuperação da senha.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

2.8.5 Tabelas de testes da tela Cadastro

Contador:	001
Localização:	Tela “Cadastro”.
Criticidade:	Baixa.
Objetivo do Teste:	Verificar se o <i>software</i> insere a foto do usuário corretamente.
Caso de teste:	Testar se a foto do usuário é inserida de modo correto.
Pré-Condição:	1. Acessar a tela de cadastro.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela de cadastro; 2. Apertar no ícone “+” para inserir uma foto; 3. Inserir uma foto escolhida no armazenamento; 4. Efetuar o cadastro e acessar o aplicativo.
Resultado Esperado:	1. O sistema deverá inserir a foto do usuário de maneira correta.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	002
Localização:	Tela “Cadastro”.
Criticidade:	Alta.
Objetivo do Teste:	Verificar o comportamento do <i>software</i> ao preencher os campos com dados inválidos.
Caso de teste:	Testar o programa ao inserir dados inválidos.
Pré-Condição:	1. Acessar a tela de cadastro.
Procedimento:	1. Acessar a tela de cadastro; 2. Inserir informações inválidas nos campos de cadastro; 3. Clicar no botão de ir para a próxima etapa.
Resultado Esperado:	1. O sistema deverá informar um erro referente ao campo que foi preenchido incorretamente para o usuário.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	003
Localização:	Tela “Cadastro”.
Criticidade:	Baixa.
Objetivo do Teste:	Verificar se o <i>software</i> realiza o auto preenchimento dos campos de endereço a partir do CEP.
Caso de teste:	Testar a ferramenta de inserção automática dos campos de endereço.
Pré-Condição:	1. Acessar a tela de cadastro.
Procedimento:	1. Acessar a tela de cadastro; 2. Inserir um CEP válido com 8 caracteres no campo indicado; 3. Efetuar o cadastro do usuário.
Resultado Esperado:	1. O sistema deverá preencher automaticamente os campos “estado”, “cidade”, “bairro” e “endereço” a partir do CEP informado.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	004
Localização:	Tela “Cadastro”.
Criticidade:	Alta.
Objetivo do Teste:	Verificar se o <i>software</i> informa um erro ao usuário quando todos os campos não forem preenchidos.
Caso de teste:	Testar se um erro é informado quando o usuário deixa campos em branco no momento do cadastro.
Pré-Condição:	1. Acessar a tela de cadastro.
Procedimento:	1. Acessar a tela de cadastro; 2. Deixar campos em branco no momento cadastral; 3. Clicar no botão de ir para a próxima etapa ou “Cadastrar-se”.
Resultado Esperado:	1. O sistema deverá informar um erro ao usuário referente ao campo que foi deixado em branco.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	005
Localização:	Tela “Cadastro”.
Criticidade:	Alta.
Objetivo do Teste:	Verificar se o <i>software</i> impede que menores de dezoito anos se cadastrem no aplicativo.
Caso de teste:	Testar se um erro é informado quando um usuário com menos de dezoito anos tentar se cadastrar no programa.
Pré-Condição:	1. Acessar a tela de cadastro.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela de cadastro; 2. Preencher todos os campos solicitados; 3. Marcar como “não” a opção “Você tem mais de 18 anos?”; 4. Efetuar o cadastro do usuário.
Resultado Esperado:	1. O sistema deverá informar um erro ao usuário quando tentar se cadastrar.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	006
Localização:	Tela “Cadastro”.
Criticidade:	Alta.
Objetivo do Teste:	Verificar se o <i>software</i> impede que usuários que não aceitem os termos de uso se cadastrem no aplicativo.
Caso de teste:	Testar se um erro é informado ao usuário que não aceita os termos de uso no momento do cadastro.
Pré-Condição:	1. Acessar a tela de cadastro.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela de cadastro; 2. Preencher todos os campos solicitados; 3. Não marcar a opção “Declaro que li os termos de uso e estou ciente”; 4. Efetuar o cadastro do usuário.
Resultado Esperado:	1. O sistema deverá informar um erro ao usuário quando tentar se cadastrar.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

2.8.6 Tabelas de testes da tela Notícias

Contador:	001
Localização:	Tela “Notícias”.
Criticidade:	Média.
Objetivo do Teste:	Verificar se é possível buscar palavras-chave na barra de pesquisa.
Caso de teste:	Testar o funcionamento da barra de pesquisa.
Pré-Condição:	1. Estar na tela de notícias.
Procedimento:	1. Digitar a palavra escolhida; 2. Verificar se as notícias que contém essa palavra apareceram.
Resultado Esperado:	1. As notícias que têm a palavra chave em seu conteúdo aparecem em ordem cronológica (data de publicação) na tela.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	002
Localização:	Tela “Notícias”.
Criticidade:	Alta.
Objetivo do Teste:	Verificar se é possível acessar a notícia.
Caso de teste:	Testar se os usuários podem visualizar as notícias.
Pré-Condição:	1. As notícias estão listadas na tela.
Procedimento:	1. Acessar a tela de notícias; 2. Clicar em uma notícia desejada.
Resultado Esperado:	1. O sistema deverá possibilitar a visualização da notícia no navegador;
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	003
Localização:	Tela “Notícias”.
Criticidade:	Alta.
Objetivo do Teste:	Verificar se há a possibilidade de realizar a rolagem na tela.
Caso de teste:	Testar a ferramenta de rolagem de tela.
Pré-Condição:	1. As notícias estão listadas na tela.
Procedimento:	1. Acessar a tela de notícias; 2. Efetuar rolagem na tela.
Resultado Esperado:	1. O sistema deverá possibilitar a rolagem de notícias em toda a tela;
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	004
Localização:	Tela “Notícias”.
Criticidade:	Média.
Objetivo do Teste:	Verificar se os filtros escolhidos estão funcionando.
Caso de teste:	Testar a ferramenta de filtragem de notícias.
Pré-Condição:	1. Escolher filtro já pré-determinado.
Procedimento:	1. Acessar o ícone de filtragem; 2. Selecionar o filtro escolhido; 3. Confirmar a seleção.
Resultado Esperado:	1. O sistema deverá apresentar as notícias relacionadas ao filtro escolhido.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	005
Localização:	Tela “Notícias”.
Criticidade:	Baixa.
Objetivo do Teste:	Verificar o comportamento do software ao pesquisar espaços em branco na barra de pesquisa.
Caso de teste:	Testar a ferramenta de pesquisa com espaços em branco.
Pré-Condição:	1. Barra de pesquisa sem nenhum caractere ou espaços inseridos.
Procedimento:	1. Acessar a tela de notícias; 2. Preencher barra de pesquisa com alguns espaços em branco.
Resultado Esperado:	1. O sistema deverá continuar exibindo as notícias iniciais.
Resultado Validado:	O <i>software</i> não funcionou como esperado.

Fonte: Autoria própria, 2022

2.8.7 Tabelas de testes da tela Perfil

Contador:	001
Localização:	Tela "Perfil".
Criticidade:	Média.
Objetivo do Teste:	Verificar se é possível que o usuário modifique todos os seus dados.
Caso de teste:	Testar se é possível que o usuário modifique seus dados cadastrais.
Pré-Condição:	1. Campos preenchidos com os dados informados pelo usuário durante o cadastro.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela de perfil; 2. Modificar os campos das informações do usuário; 3. Clicar no botão correspondente aos campos modificados para que as alterações sejam salvas.
Resultado Esperado:	1. O sistema deverá alterar os dados do usuário.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	002
Localização:	Tela "Perfil".
Criticidade:	Média.
Objetivo do Teste:	Verificar o comportamento do <i>software</i> ao preencher o campo "CEP" com um valor inválido.
Caso de teste:	Testar o programa ao inserir um CEP inválido.
Pré-Condição:	1. Campos preenchidos com os dados informados pelo usuário durante o cadastro.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela de perfil; 2. Inserir informações inválidas no campo "CEP"; 3. Clicar no botão correspondente ao campo modificado para que a alteração seja salva.
Resultado Esperado:	1. O sistema deverá informar um erro referente ao campo CEP, que foi preenchido incorretamente.
Resultado Validado:	O <i>software</i> não funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	003
Localização:	Tela "Perfil".
Criticidade:	Média.
Objetivo do Teste:	Verificar o comportamento do <i>software</i> ao inserir um campo em branco durante a alteração dos dados.
Caso de teste:	Testar o programa ao inserir um campo em branco no decorrer da mudança de dados do usuário.
Pré-Condição:	1. Dados informados corretamente pelo usuário durante o cadastro.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela de perfil; 2. Inserir informações em branco nos campos que contêm informações do usuário; 3. Clicar no botão correspondente aos campos modificados para que as alterações sejam salvas.
Resultado Esperado:	1. O sistema deverá informar um erro referente ao campo que foi preenchido em branco.
Resultado Validado:	O <i>software</i> não funcionou como esperado.

Fonte: Autoria própria, 2022

2.8.8 Tabelas de testes da tela Doadores

Contador:	001
Localização:	Tela “Doadores”.
Criticidade:	Média.
Objetivo do Teste:	Verificar se os botões referentes a cada doador realizam suas respectivas funções.
Caso de teste:	Testar se é possível ligar, compartilhar, reportar e enviar mensagens para o doador.
Pré-Condição:	1. Todos os doadores estão listados.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela de doadores; 2. Clicar no menu de algum usuário doador; 3. Clicar em cada uma das funções: ligar, compartilhar, reportar; 4. Clica fora do menu para fechá-lo; 5. Clicar em algum usuário e testar os botões de ligar e enviar mensagens através do <i>WhatsApp</i>.
Resultado Esperado:	1. O sistema deverá executar as respectivas tarefas com êxito.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	002
Localização:	Tela "Doadores".
Criticidade:	Baixa.
Objetivo do Teste:	Verificar se é possível realizar a filtragem de pessoas disponíveis para entregar alimentos.
Caso de teste:	Testar se é possível realizar a filtragem de pessoas de acordo com sua disponibilidade.
Pré-Condição:	1. Todos os doadores estão listados.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela de doadores; 2. Clicar no botão de filtrar; 3. Marcar a opção "Apenas pessoas com disponibilidade de entregar"; 4. Clicar no botão "Aplicar".
Resultado Esperado:	1. O sistema deverá filtrar as pessoas que estejam disponíveis para entregar alimentos.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	003
Localização:	Tela "Doadores".
Criticidade:	Baixa.
Objetivo do Teste:	Verificar se é possível realizar a filtragem de pessoas de acordo com suas regiões.
Caso de teste:	Testar se é possível realizar a filtragem de pessoas de acordo com sua localização.
Pré-Condição:	1. Todos os doadores estão listados.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela de doadores; 2. Clicar no botão de filtrar; 3. Escolher se a filtragem será de acordo com o mesmo estado, município ou bairro; 4. Clicar no botão "Aplicar".
Resultado Esperado:	1. O sistema deverá filtrar as pessoas que estejam na região selecionada.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

2.8.9 Tabelas de testes da tela Ajude Alguém

Contador:	001
Localização:	Tela “Ajude Alguém”.
Criticidade:	Média.
Objetivo do Teste:	Verificar se os botões referentes a cada receptor realizam suas respectivas funções.
Caso de teste:	Testar se é possível ligar, compartilhar, reportar e enviar mensagens para o receptor.
Pré-Condição:	1. Todos os receptores estão listados.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela “Ajude Alguém”; 2. Clicar no menu de algum usuário receptor; 3. Clicar em cada uma das funções: ligar, compartilhar, reportar; 4. Clica fora do menu para fechá-lo; 5. Clicar em algum usuário e testar os botões de ligar e enviar mensagens através do <i>WhatsApp</i>.
Resultado Esperado:	1. O sistema deverá executar as respectivas tarefas com êxito.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	002
Localização:	Tela "Ajude Alguém".
Criticidade:	Baixa.
Objetivo do Teste:	Verificar se é possível realizar a filtragem de pessoas disponíveis para buscar alimentos.
Caso de teste:	Testar se é possível realizar a filtragem de pessoas de acordo com sua disponibilidade.
Pré-Condição:	1. Todos os receptores estão listados.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela "Ajude Alguém"; 2. Clicar no botão de filtrar; 3. Marcar a opção "Apenas pessoas com disponibilidade de buscar"; 4. Clicar no botão "Aplicar".
Resultado Esperado:	1. O sistema deverá filtrar as pessoas que estejam disponíveis para buscar alimentos.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	003
Localização:	Tela “Ajude Alguém”.
Criticidade:	Baixa.
Objetivo do Teste:	Verificar se é possível realizar a filtragem de pessoas de acordo com suas regiões.
Caso de teste:	Testar se é possível realizar a filtragem de pessoas de acordo com sua localização.
Pré-Condição:	1. Todos os receptores estão listados.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela “Ajude Alguém”; 2. Clicar no botão de filtrar; 3. Escolher se a filtragem será de acordo com o mesmo estado, município ou bairro; 4. Clicar no botão "Aplicar".
Resultado Esperado:	1. O sistema deverá filtrar as pessoas que estejam na região selecionada.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

2.8.10 Tabelas de testes da tela ONGs

Contador:	001
Localização:	Tela “ONGs”.
Criticidade:	Média.
Objetivo do Teste:	Verificar se os botões referentes a cada ONG realizam suas respectivas funções.
Caso de teste:	Testar se é possível ligar, compartilhar e reportar a ONG.
Pré-Condição:	1. Todas as ONGs estão listadas.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela “ONGs”; 2. Clicar no menu de alguma ONG; 3. Clicar em cada uma das funções: ligar, compartilhar, reportar.
Resultado Esperado:	1. O sistema deverá executar as respectivas tarefas com êxito.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	002
Localização:	Tela “ONGs”.
Criticidade:	Média.
Objetivo do Teste:	Verificar se os botões referentes a cada ONG realizam suas respectivas funções no <i>modal</i> .
Caso de teste:	Testar se é possível ligar e abrir o site da ONG.
Pré-Condição:	1. Todas as ONGs estão listadas.
Procedimento:	<ol style="list-style-type: none">1. Acessar a tela “ONGs”;2. Clicar em alguma ONG;3. Clicar no botão ligar;4. Clicar no <i>link</i> de acesso da ONG.
Resultado Esperado:	<ol style="list-style-type: none">1. O sistema deverá ligar para a ONG;2. O sistema deverá abrir o site da ONG.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	003
Localização:	Tela “ONGs”.
Criticidade:	Média.
Objetivo do Teste:	Verificar se é possível aplicar as opções de filtro (filtrar pela localização da ONG e pela opção de apresentar site e/ou local de atuação fixo).
Caso de teste:	Testar se é possível aplicar as opções de filtro.
Pré-Condição:	1. Todas as ONGs estão listadas.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela “ONGs”; 2. Clicar na opção de filtro; 3. Selecionar diferentes combinações de opções de filtros.
Resultado Esperado:	1. O sistema deverá exibir ONGs de acordo com os filtros selecionados.
Resultado Validado:	O <i>software</i> não funcionou como esperado.

Fonte: Autoria própria, 2022

2.8.11 Tabelas de testes da tela Configurações

Contador:	001
Localização:	Tela “Configurações”.
Criticidade:	Baixa.
Objetivo do Teste:	Verificar se ocorre a mudança de tema.
Caso de teste:	Testar se a mudança do tema claro para escuro acontece.
Pré-Condição:	1. Tema claro definido como padrão.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela “Configurações”; 2. Clicar no botão <i>switch</i> da opção de tema; 3. Navegar entre as telas para verificar se o tema foi alterado; 4. Fechar o aplicativo e abri-lo novamente.
Resultado Esperado:	1. O sistema deverá realizar a alternância entre os temas e o armazenamento do tema escolhido.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	002
Localização:	Tela “Configurações”.
Criticidade:	Média.
Objetivo do Teste:	Verificar se a audiodescrição é habilitada/desabilitada.
Caso de teste:	Testar se a ativação/desativação da audiodescrição ocorre.
Pré-Condição:	1. Audiodescrição desativada/ativada de acordo com a escolha do usuário no cadastro.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela “Configurações”; 2. Clicar no botão switch da opção de audiodescrição; 3. Navegar entre as telas para verificar se a audiodescrição está funcionando.
Resultado Esperado:	1. O sistema deverá realizar a alternância entre os estados da audiodescrição.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	003
Localização:	Tela “Configurações”.
Criticidade:	Média.
Objetivo do Teste:	Verificar se a opção “Continuar como receptor/doador” é habilitada/desabilitada.
Caso de teste:	Testar se a ativação/desativação da opção “Continuar como receptor/doador” ocorre.
Pré-Condição:	1. Opção “Continuar como receptor/doador” desativada/ativada de acordo com a escolha do usuário.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela “Configurações”; 2. Clicar no botão <i>switch</i> da opção de “Continuar como receptor/doador”; 3. Efetuar <i>logout</i> do aplicativo; 4. Entrar com uma conta com um tipo de usuário (Doador ou Receptor) distinto da conta anteriormente conectada; 5. Abrir tela de Doadores/Ajude Alguém; 6. Verificar se a conta anteriormente conectada é listada de acordo com a opção inserida na opção “Continuar como receptor/doador”.
Resultado Esperado:	1. O sistema deverá realizar a alternância entre os estados da opção de “Continuar como receptor/doador” e fazer a listagem ou não do usuário de acordo com o estado (ligado/desligado) da opção escolhida.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	004
Localização:	Tela “Configurações”.
Criticidade:	Média.
Objetivo do Teste:	Verificar se é possível que o usuário saia de sua conta.
Caso de teste:	Testar se é possível que o usuário realize <i>logout</i> .
Pré-Condição:	1. Usuário conectado no aplicativo.
Procedimento:	1. Acessar a tela “Configurações”; 2. Clicar no botão “Sair”; 3. Clicar em “Sim” quando o <i>pop-up</i> de confirmação de <i>logout</i> for aberto.
Resultado Esperado:	1. O sistema deverá efetuar o <i>logout</i> .
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	005
Localização:	Tela “Configurações”.
Criticidade:	Média.
Objetivo do Teste:	Verificar se é possível que o usuário delete sua conta.
Caso de teste:	Testar se é possível que o usuário realize a exclusão permanente de sua conta.
Pré-Condição:	1. Usuário conectado no aplicativo.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela “Configurações”; 2. Clicar no botão “Deletar conta”; 3. Inserir a senha cadastrada no <i>modal</i> de exclusão de conta; 4. Clicar no botão “Deletar conta”.
Resultado Esperado:	1. O sistema deverá realizar a exclusão da conta do usuário cadastrado.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

2.9. Principais mudanças realizadas após a rotina de teste do aplicativo

Após os testes do aplicativo, foi possível encontrar alguns problemas que impediam a plena usabilidade do *software* como, por exemplo, um problema na filtragem de ONGs, de modo que o usuário não conseguia filtrar somente ONGs em locais físicos e ao executar essa função o aplicativo parava de funcionar. Além disso, foram encontrados problemas de erros de uso do *software* que não afetam a usabilidade do aplicativo, mas permitem o preenchimento de dados de maneira inválida como, por exemplo, alterar o CEP na tela "Meu Perfil" para um CEP inválido com menos de 8 caracteres ou preencher os campos da mesma tela com espaços em branco. Por fim, após os testes, os erros de filtragem, preenchimento inválido de dados da tela de perfil e busca com espaços em branco na tela de notícias foram identificados pela equipe de desenvolvimento do aplicativo e, então, corrigidos.

2.10 Rotinas de testes do site

2.10.1 Testes escolhidos

Os testes feitos após a produção do *website* foram os mesmos do aplicativo: teste funcional, teste de integração e teste de ponta a ponta.

2.10.2 Tabelas de teste da tela Inicial

Contador:	001
Localização:	Tela "Inicial".
Criticidade:	Média.
Objetivo do Teste:	Verificar o comportamento ao clicar no botão para realizar a instalação do aplicativo.
Caso de teste:	Testar o botão de <i>download</i> do aplicativo.
Pré-Condição:	1. Acessar o <i>site</i> pela primeira vez.
Procedimento:	1. Acessar o <i>site</i> ; 2. Clicar no botão que realiza a instalação do aplicativo.
Resultado Esperado:	1. O sistema deverá instalar o aplicativo automaticamente.
Resultado Validado:	O <i>software</i> não funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	002
Localização:	Tela “Inicial”.
Criticidade:	Média.
Objetivo do Teste:	Verificar se a mudança de tema ocorre.
Caso de teste:	Testar se a mudança do tema claro para escuro acontece
Pré-Condição:	1. Tema claro definido como padrão.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar o <i>site</i>; 2. Clicar no botão <i>switch</i> da opção de tema; 3. Navegar entre as telas para verificar se o tema foi alterado; 4. Fechar o <i>site</i> e abri-lo novamente.
Resultado Esperado:	1. O sistema deverá realizar a alternância entre os temas e o armazenamento do tema escolhido.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

2.10.3 Tabelas de testes da tela Notícias

Contador:	001
Localização:	Tela “Notícias”.
Criticidade:	Média.
Objetivo do Teste:	Verificar se é possível buscar palavras-chave na barra de pesquisa.
Caso de teste:	Testar o funcionamento da barra de pesquisa.
Pré-Condição:	1. Notícias listadas na tela.
Procedimento:	1. Digitar a palavra escolhida; 2. Verificar se as notícias que contém essa palavra apareceram.
Resultado Esperado:	1. As notícias que têm a palavra chave em seu conteúdo aparecem em ordem cronológica (data de publicação) na tela.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	002
Localização:	Tela "Notícias".
Criticidade:	Alta.
Objetivo do Teste:	Verificar se o carrossel (responsável por mudar as notícias que aparecem na tela ao clicar no botão) funciona.
Caso de teste:	Testar o funcionamento do carrossel.
Pré-Condição:	1. Notícias listadas na tela.
Procedimento:	1. Acessar a tela de notícias; 2. Clicar nas setas para que as notícias sejam alternadas.
Resultado Esperado:	1. As notícias deverão mudar quando a seta for clicada.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

Contador:	003
Localização:	Tela “Notícias”.
Criticidade:	Média.
Objetivo do Teste:	Verificar se o usuário é direcionado para o <i>site</i> da notícia escolhida ao clicar em “Ver mais”.
Caso de teste:	Testar o funcionamento dos botões de direcionamento para a notícia desejada.
Pré-Condição:	1. Notícias listadas na tela.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela de notícias; 2. Clicar no botão “Ver mais” referente a notícia desejada; 3. Verificar se a notícia aberta no navegador corresponde a escolhida.
Resultado Esperado:	1. Deverá ser aberta uma nova guia contendo a notícia desejada.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

2.10.4 Tabela de testes da tela Sobre Nós

Contador:	001
Localização:	Tela “Sobre Nós”.
Criticidade:	Baixa.
Objetivo do Teste:	Verificar se os <i>links</i> presentes na página funcionam.
Caso de teste:	Testar o funcionamento dos <i>links</i> da página.
Pré-Condição:	1.Botões presentes na tela.
Procedimento:	1. Acessar a tela "Sobre Nós"; 2. Clicar nos botões.
Resultado Esperado:	1.Ao clicar nos botões o usuário deverá ser redirecionado para a área desejada.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

2.10.5 Tabela de testes da tela Formulário de Dúvidas

Contador:	001
Localização:	Tela "Formulário de Dúvidas".
Criticidade:	Média.
Objetivo do Teste:	Verificar se os dados do formulário são enviados corretamente.
Caso de teste:	Testar o do botão “Enviar” do formulário.
Pré-Condição:	1. Inserir dados no formulário.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela de enviar dúvidas; 2. Inserir dados no formulário; 3. Clicar no botão “Enviar”. 4. Confirmar no <i>e-mail</i> se os dados foram enviados.
Resultado Esperado:	1. Os dados deverão ser enviados para o <i>e-mail</i> do projeto.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

2.10.6 Tabela de teste da tela contribua

Contador:	001
Localização:	Tela “Contribua”.
Criticidade:	Média.
Objetivo do Teste:	Verificar se o botão “Copiar Pix” funciona corretamente.
Caso de teste:	Testar o do botão “Copiar Pix” da página.
Pré-Condição:	1. Clicar no botão “Copiar Piix” .
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela “Contribua”; 2. Clicar no botão “Copiar Pix”; 3. Verificar se o e-mail do projeto está na área de transferência.
Resultado Esperado:	1.O <i>e-mail</i> do projeto deverá estar na área de transferência.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

2.10.7 Tabela de testes da tela Registro de ONGs e Empresas

Contador:	001
Localização:	Tela “Registro de ONGs e Empresas”.
Criticidade:	Alta.
Objetivo do Teste:	Verificar se os dados do formulário são enviados corretamente.
Caso de teste:	Testar o do botão “Enviar” do formulário.
Pré-Condição:	1. Inserir dados no formulário.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela de enviar dúvidas; 2. Inserir dados no formulário; 3. Clicar no botão “Enviar”. 4. Confirmar no e-mail se os dados foram enviados.
Resultado Esperado:	1. Os dados deverão ser enviados para o e-mail do projeto.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

2.10.8 Tabela de testes da tela ONGs

Contador:	001
Localização:	Tela “ONGs”.
Criticidade:	Alta.
Objetivo do Teste:	Verificar se é possível copiar, compartilhar e reportar os dados da ONG.
Caso de teste:	Testar os botões copiar, compartilhar e reportar.
Pré-Condição:	1. Estar na página “ONGs” e clicar em “Opções”.
Procedimento:	<ol style="list-style-type: none"> 1. Acessar a tela “ONGs”; 2. Clicar em “opções” da ONG desejada; 3. Clicar nos botões copiar, compartilhar e reportar”; 4. Confirmar se estão funcionando corretamente.
Resultado Esperado:	1. Os dados da ONG desejada deverão ser copiadas, compartilhadas e reportadas.
Resultado Validado:	O <i>software</i> funcionou como esperado.

Fonte: Autoria própria, 2022

2.11. Principais mudanças realizadas após a rotina de teste do site

Após a realização dos testes foi possível verificar um erro referente ao botão que realiza a instalação do aplicativo, pois ele não cumpria a sua função esperada, já que o intuito do site é divulgar o aplicativo, redirecionando o usuário para seu download. A equipe analisou as possíveis maneiras para resolver esse problema e, em conjunto, conseguiu achar a sua solução, fazendo com que o botão cumprisse sua tarefa.

2.12 Diferenciais do projeto

Em uma pesquisa realizada pelos integrantes, foram encontrados quatro projetos com propostas semelhantes: “Doação do Bem”, “Alguém Quer?”, “Comida Invisível” e “Banco de Alimentos”. Diante disso, os diferenciais implantados no aplicativo apresentado são:

- O projeto compõe um aplicativo e um *website*;
- A opção de ver notícias relacionadas à alimentação e agricultura em geral;
- A possibilidade de se cadastrar como um usuário doador (pessoa física) no aplicativo e ter conexão com pessoas e/ou instituições necessitadas, no qual, tanto o doador quanto o carecido poderão se encontrar pessoalmente para entregar ou receber a doação;
- O projeto contém um *website* com o propósito de divulgar o aplicativo, apresentar algumas funcionalidades como as opções de contato e propagar a intenção da proposta;
- Além disso, a instituição (ONG), caso necessite de doações, poderá se cadastrar e passar por uma análise pelo grupo desenvolvedor que comprovará sua legitimidade.

Entre o projeto apresentado e o aplicativo “Doação do Bem” as semelhanças encontradas são:

- Uma área para encontrar alguma instituição interessada em receber doações;
- Uma tela na qual é possível se cadastrar, ver informações da conta e alterar os dados cadastrados.

Entre o projeto apresentado e o aplicativo “Alguém Quer?” as semelhanças encontradas são:

- Uma tela para se cadastrar como um usuário doador, ver informações da conta e alterar os dados cadastrados;
- As pessoas que necessitam de doações entram em contato com o doador cadastrado.

Entre o projeto apresentado e o aplicativo “Comida Invisível” as semelhanças encontradas são:

- Um *website*, onde é possível ver informações sobre a ONG;
- Entrar em contato com instituições necessitadas de doações;
- Uma tela para se cadastrar, ver informações da conta e alterar os dados cadastrados.

Entre o projeto apresentado e o aplicativo “Banco de Alimentos” as semelhanças encontradas são:

- Um *website*, onde é possível ver informações sobre a ONG;
- Entrar em contato com a instituição necessitada;
- Uma tela para se cadastrar, ver informações da conta e alterar os dados cadastrados.

3 DISCUSSÃO DOS RESULTADOS

3.1 Resultados das pesquisas bibliográficas

A partir das pesquisas dos dados relacionados à insegurança alimentar no Brasil, o grupo refletiu sobre o impacto que todos os seres humanos têm na alimentação mundial, visto que o desperdício ainda é frequente e milhares de alimentos são descartados diariamente desnecessariamente. Os brasileiros sofrem muito com a vulnerabilidade alimentar, uma vez que é possível visualizar o país no mapa da fome novamente, então é necessária a conscientização das pessoas sobre a realidade em que vivemos, para que com a mobilização social, seja possível amenizar os impactos do desperdício na vida dos cidadãos.

3.2 Resultados das pesquisas de campo

Após a pesquisa realizada através do Google Forms, foi identificada a pertinência do projeto a partir de perguntas que visavam descobrir a opinião dos possíveis usuários em relação à ideia central. Ao analisar as respostas, foi perceptível a aceitação do “público” quanto ao projeto.

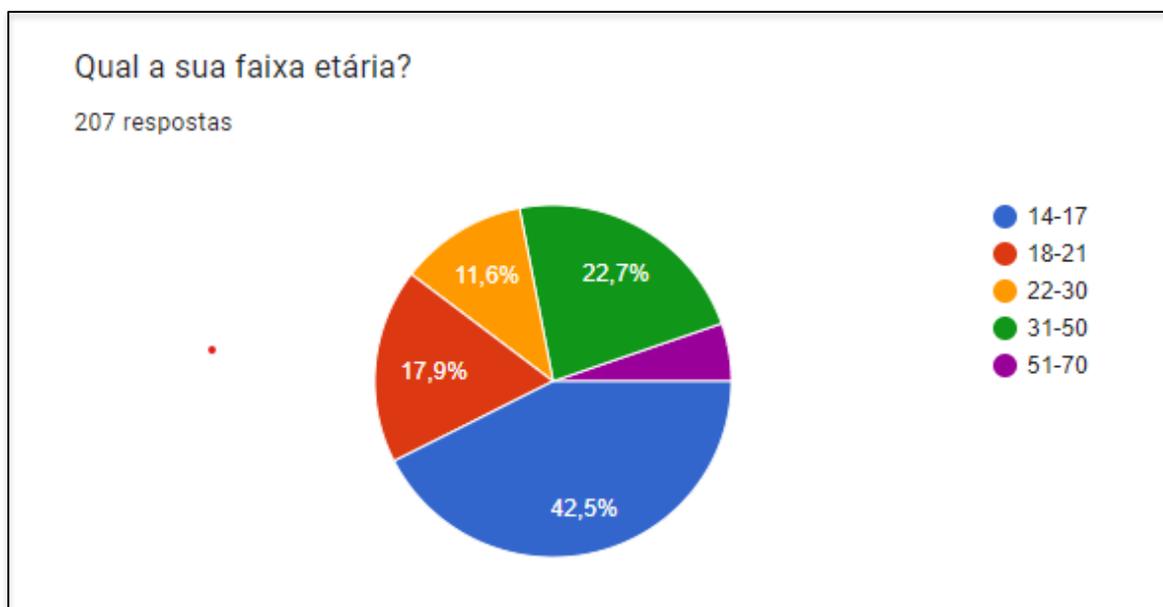
Figura 55 – Resultado do Formulário (1)



Fonte: Autoria Própria, 2022

Outra informação coletada foi a idade dos possíveis usuários, estabelecendo uma faixa etária, majoritariamente, de maiores de 18 anos.

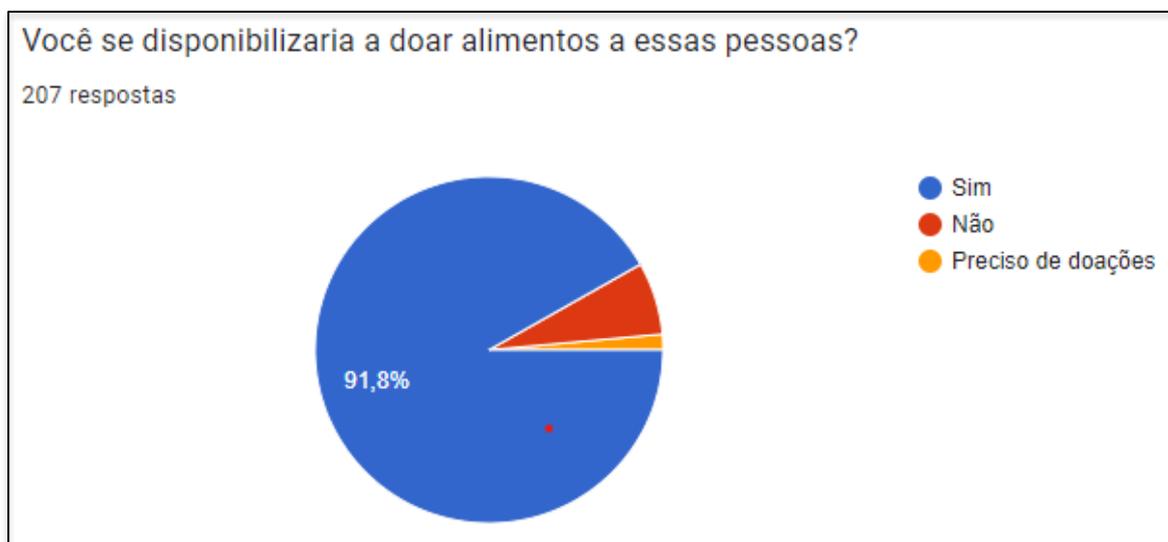
Figura 56 – Resultado do Formulário (2)



Fonte: Autoria Própria, 2022

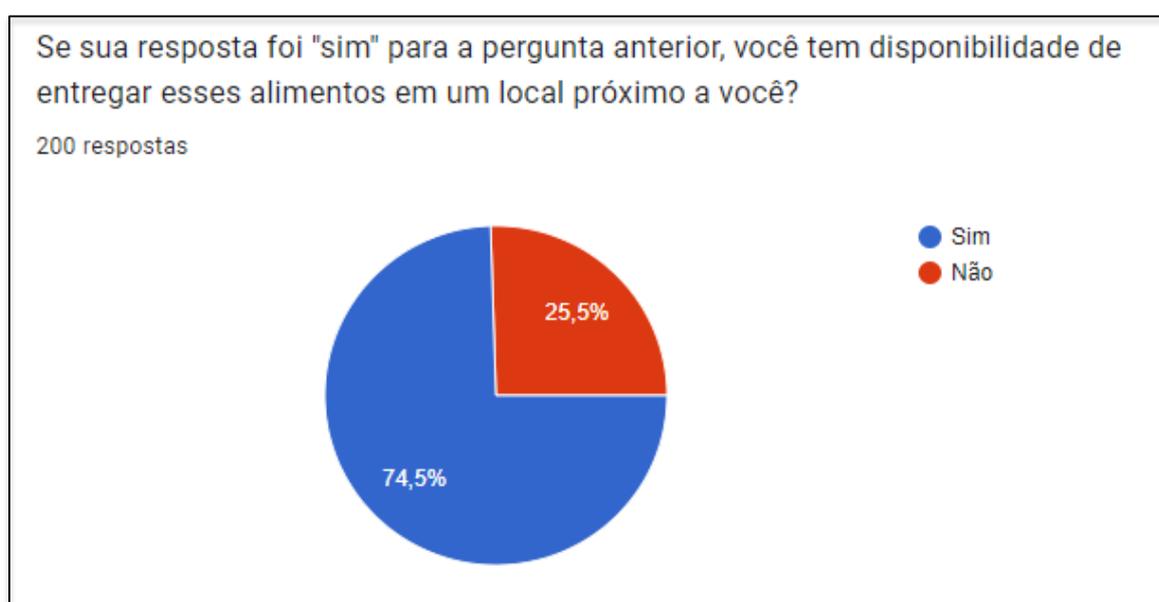
A partir do formulário, o grupo pôde estabelecer uma concepção de que a maioria dos doadores se disponibilizaria a entregar os alimentos, assim como a maioria dos receptores buscaria as doações.

Figura 57 – Resultado do Formulário (3)



Fonte: Autoria Própria, 2022

Figura 58 – Resultado do Formulário (4)



Fonte: Autoria Própria, 2022

3.3 Resultados da pesquisa na microrregião

Após as pesquisas de campo, o grupo decidiu contatar o Centro de Referência e Assistência Social de Taboão da Serra, a fim de estabelecer a microrregião a ser atendida pelo projeto. Ao analisar os dados recebidos, foi definido que os bairros mais carentes de Taboão da Serra se concentram na área atendida pelo CRAS Trianon, portanto, foram os bairros escolhidos como foco do aplicativo apresentado.

4 CONSIDERAÇÕES FINAIS

O projeto GreenPoint coopera no combate dos índices da fome e incentiva o desenvolvimento sustentável no município de Taboão da Serra, em específico os bairros Jd. Freitas Júnior, Jd. Record, Jd. Roberto, Vila Mafalda e Sítio das Madres. Com o aplicativo sendo colocado em uso, busca-se auxiliar as pessoas em situação de vulnerabilidade social, além de conscientizar a população acerca do consumo adequado dos alimentos e da agricultura sustentável. Incentivar a diminuição do desperdício e/ou descarte de alimentos excedentes por parte de pessoas que podem doar para instituições ou pessoas comuns e a comunicação entre os usuários do aplicativo será uma forma ágil de realizar tais doações.

Por outro lado, o projeto promove o exercício da cidadania e colaboração no município, uma vez que é dever moral de todo cidadão a contribuição àqueles que estão em situação de carência social. A partir das doações, muitas pessoas receberão alimentos e não estarão sujeitas às consequências da fome, como doenças relacionadas à falta de uma alimentação digna, desnutrição, anemia, entre outras.

Portanto, o projeto atende aos objetivos propostos e possui grande relevância no âmbito social, realizando as atividades predefinidas. Os resultados da implementação do aplicativo juntamente ao *website* serão de grande importância ao combater o avanço do problema da fome na cidade de Taboão da Serra.

REFERÊNCIAS

ALVARENGA, Darlan. **Brasil tem a 4ª maior taxa de desemprego do mundo, aponta ranking com 44 países.** [S. l.], 22 nov. 2021. Disponível em: <<https://g1.globo.com/economia/noticia/2021/11/22/brasil-tem-a-4a-maior-taxa-de-desemprego-do-mundo-aponta-ranking-com-44-paises.ghtml>>. Acesso em: 16 de maio de 2022.

CASTRO, Diego. **Tipos de testes de software: diferenças e exemplos.** Porto Alegre -RS, 26 de maio de 2021. Disponível em: <https://targettrust.com.br/blog/10-principais-tipos-de-teste-de-software/>. Acesso em: 19 set. 2022.

CATARINA, Robson. **Conheça os 10 principais Tipos de Testes de Software.** Porto Alegre - RS, 13 jan. 2021. Disponível em: <https://targettrust.com.br/blog/10-principais-tipos-de-teste-de-software/>. Acesso em: 19 set. 2022.

DE OLIVEIRA, Abrahão; REINO, Guilbert. **Cidade de SP tem aumento de 30% no nº de microempreendedores individuais ativos; saiba como fazer a formalização.** São Paulo, 16 set. 2021. Disponível em: <<https://g1.globo.com/sp/sao-paulo/noticia/2021/09/16/cidade-de-sp-tem-aumento-de-30percent-no-numero-de-microempreendedores-individuais-ativos-saiba-como-fazer-a-formalizacao.ghtml>>. Acesso em: 20 mai. 2022.

FERNANDES, Diego. **Expo: o que é, para que serve e quando utilizar?.** Blog RocketSeat, 27 mar. 2018. Disponível em: <https://blog.rocketseat.com.br/expo-react-native/>. Acesso em: 19 set. 2022.

FREITAS, Eduardo. **Saúde e Bem-estar.** [S. l.], entre 2015 e 2022. Disponível em: <https://brasil.uol.com.br/brasil/fome-no-brasil.htm>>. Acesso em: 29 abr. 2022.

GUEDES, Marylene. **O que vem a ser o ciclo de vida?.** *In:* GUEDES, Marylene. **Ciclo de vida do software: por que é importante saber?.** TreinaWeb, 16 mar. 2018. Disponível em: [https://www.treinaweb.com.br/blog/ciclo-de-vida-software-por-que-e-importante-saber#:~:text=O%20ciclo%20de%20vida%20de%20um%20software%20%C3%A9%20uma%20estrutura,%2C%20lan%20%C3%A7ado%2C%20aprimorado%20e%20finalizado](https://www.treinaweb.com.br/blog/ciclo-de-vida-software-por-que-e-importante-saber#:~:text=O%20ciclo%20de%20vida%20de%20um%20software%20%C3%A9%20uma%20estrutura,%2C%20lan%20%C3%A7ado%2C%20aprimorado%20e%20finalizado.). Acesso em: 5 set. 2022.

IBGE. **Mapa de pobreza e desigualdade.** [S. l.], 2003. Disponível em: <<https://cidades.ibge.gov.br/brasil/sp/taboad-da-serra/pesquisa/36/0?indicador=30246&tipo=ranking>>. Acesso em: 16 de maio de 2022.

IBGE. **Aumento da obesidade entre adultos, SP.** [S. l.], 21 out. 2020. Disponível em: <<https://www.gov.br/pt-br/noticias/saude-e-vigilancia-sanitaria/2020/10/pesquisa-do-ibge-mostra-aumento-da-obesidade-entre-adultos>>. Acesso em: 20 maio 2022.

IBM. **Diagramas de Classes**. [S. l.], 3 de maio de 2021. Disponível em: <https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=structure-class-diagrams>. Acesso em: 6 set. 2022.

INVISÍVEL, **Comida. Termo de Assunção de Responsabilidade**. [S. l.], 10 nov. 2020. Disponível em: <https://app.comidainvisivel.com.br/#/termo-assuncao>. Acesso em: 3 jun. 2022.

LENZANA, Ricardo. **O Modelo de Negócios e o Diagrama Canvas**. Blog Academia Perspectiva, 12 abr. 2021. Disponível em: <https://blog.academiaperspectiva.com/o-modelo-de-negocios-e-o-diagrama-canvas/#:~:text=O%20objetivo%20do%20Diagrama%20Canvas,uma%20Proposta%20de%20Valor%20%C3%BAnica>. Acesso em: 5 set. 2022.

MAISMEI. **MEI Microempreendedor Individual em Taboão da Serra, SP**. [S. l.], 2019. Disponível em: <https://www.maismei.com.br/mei-microempreendedor-individual-taboao-da-serra-sp>. Acesso em: 29 abr. 2022.

ONG BANCO DE ALIMENTOS. **O Brasil Tem 33,1 Milhões De Pessoas Que Passam Fome**. [S. l.], 9 jun. 2022. Disponível em: <https://bancodealimentos.org.br/o-brasil-tem-331-milhoes-de-pessoas-que-passam-fome/>. Acesso em: 10 jun. 2022.

RICARD, Paul. **Segundo a OMS, 2,3 bilhões de pessoas no mundo estão com sobrepeso ou obesidade**. [S. l.], 16 dez. 2019. Disponível em: <https://oglobo.globo.com/saude/segundo-oms-23-bilhoes-de-pessoas-no-mundo-estao-com-sobrepeso-ou-obesidade-1-24141513>. Acesso em: 29 abr. 2022.

SANTANA, Joel. MER e DER: **Modelagem de Bancos de Dados**. [S. l.], 2014. Disponível em: <https://www.devmedia.com.br/mer-e-der-modelagem-de-bancos-de-dados/14332>. Acesso em: 6 set. 2022.

SILVA, Guilherme Henrique de Lira. **Tipos de testes: quais os principais e por que utilizá-los?**. São Paulo, 23 dez. 2021. Disponível em: <https://www.alura.com.br/artigos/tipos-de-testes-principais-por-que-utiliza-los?>. Acesso em: 19 set. 2022.

TRTP. MER e DER: **Modelagem de Bancos de Dados**. Paraná, 2014. Disponível em: https://www.trt9.jus.br/pds/pdst9/guidances/concepts/use_case_model_CD178AF9.html. Acesso em: 6 set. 2022.

VALLADÃO, Marcelo. **Banco de Alimentos de Taboão atende 16 mil famílias por mês**. Taboão da Serra - SP 24 jul. 2011. Disponível em: <https://www.otaboanense.com.br/banco-de-alimentos-de-taboao-atende-16-mil-familias-por-mes/>. Acesso em: 16 de maio de 2022.

PODER300. **Mais de 20 milhões passam fome no Brasil e favelas dobraram em 10 anos**. Distrito Federal, 13 out. 2021. Disponível em: <https://www.poder360.com.br/brasil/mais-de-20-milhoes-passam-fome-no-brasil-e-favelas-dobraram-em-10-anos/>. Acesso em: 20 maio 2022.