



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

Doe+: solução mobile para doação para instituições de caridade

Trabalho de Conclusão de Curso

Luiz Carlos da Conceição Júnior



São Cristóvão – Sergipe 2022

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

Luiz Carlos da Conceição Júnior

Doe+: solução mobile para doação para instituições de caridade

Trabalho de Conclusão de Curso submetido ao Departamento de Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador(a): Rafael Oliveira Vasconcelos

São Cristóvão – Sergipe 2022

Agradecimentos

Aos meus pais, por me incentivarem a continuar quando eu quis desistir do curso, e por me apoiarem mesmo sem entender o que alguém dessa área faz.

Às minhas irmãs, por sempre me fazer acreditar que sou a pessoa mais inteligente do mundo quando apareciam com alguma dúvida extremamente simples que eu sei resolver.

Aos meus amigos, os que eu fiz durante toda a vida, que contribuíram para o meu crescimento como pessoa, e os que eu fiz na universidade, que abriram minha mente para muitas coisas.

A Yoná e Geovane, que sempre estiveram presentes para falar de qualquer coisa que não fosse o curso ou o trabalho, e assim me permitiram esfriar um pouco a cabeça.

A Everton, por me apoiar e estar sempre comigo quando eu precisei, por entender minhas limitações e acreditar em mim.

Aos meus professores, em especial Adicinéia, Gilton e Rogério, que foram os principais responsáveis por me fazer gostar ainda mais do curso.

Ao meu orientador, Rafael, pela paciência e por estar sempre disponível para tirar minhas dúvidas.

A todos vocês, obrigado!

Resumo

Este trabalho apresenta uma pesquisa acerca da computação social e dos aplicativos sociais voltados para a doação para instituições de caridade. O objetivo do trabalho é desenvolver um aplicativo para dispositivos móveis que possibilite a busca por organizações de caridade e a doação para as mesmas. É feito um comparativo entre a solução desenvolvida e as existentes. Todo o processo englobado pela Engenharia de Software utilizado no desenvolvimento da aplicação é descrito, bem como as tecnologias utilizadas. Conclui-se que os aplicativos sociais têm grande importância na sociedade, ao permitir que as pessoas possam compartilhar bens e ajudar os mais necessitados, e a solução desenvolvida tem o diferencial de permitir a busca por instituições de caridade por meio de geolocalização, contando com um filtro por tipos, o que auxilia a divulgar o trabalho de ONGs que atuam na região do usuário.

Palavras-chave: Computação Social, Aplicativo Social, Doação, Aplicação Móvel, Geolocalização.

Abstract

This work presents a research on social computing and social applications aimed at donating to charities. The objective of the work is to develop an application for mobile devices that allows the search for charitable organizations and the donation to them. A comparison is made between the developed solution and the existing ones. The entire process encompassed by Software Engineering used in the development of the application is described, as well as the technologies applied. It is concluded that social applications have great importance in society, as they allow people to share goods and help people in need, also the solution developed has the differential of allowing the search for charities through geolocation, with a filter by types, which helps to publicize the work of NGOs that work on user's region.

Keywords: Social Computing, Social Application, Donation, Mobile Application, Geolocation.

Lista de quadros

Quadro 1 – [RF001] permitir cadastro de usuário	31
Quadro 2 - [RF002] permitir busca por instituições de caridade próximas	32
Quadro 3 - [RF003] filtrar tipos de instituições	32
Quadro 4 - [RF004] exibir dados das instituições	32
Quadro 5 - [RF005] permitir cadastro de instituição	33
Quadro 6 - [RF006] permitir cópia de dados	33
Quadro 7 - [RF007] permitir busca e doação sem cadastro	34
Quadro 8 - [RF008] exibir informações básicas ao selecionar uma instituição	34
Quadro 9 - [NFUS001] listar por localização	35
Quadro 10 - [NFUS002] consistência na interface do usuário	35
Quadro 11 - [NFPO001] versões multiplataforma	36
Quadro 12 - [NFIM001] linguagem de programação	36
Quadro 13 - [NFIN001] banco de dados	37

Lista de figuras

Figura 1 - Aviso de fim do suporte ao deploy gratuito pelo Heroku.....	21
Figura 2 - Telas do Charity Miles.....	25
Figura 3 - Telas do Solidarius	26
Figura 4 - Tela inicial do Para Quem Doar	27
Figura 5 - Diagrama de Casos de Uso	38
Figura 6 - Modelo de Processos de Negócio	39
Figura 7 - Hierarquia de Telas.....	40
Figura 8 - Criar novo app no Heroku	51
Figura 9 - Passo a passo para deploy via Contâiner	52
Figura 10 - Tela Inicial.....	53
Figura 11 - Tela de Carregamento.....	54
Figura 12 - Mapa	54
Figura 13 - Filtro	55
Figura 14 - Informações da ONG	55
Figura 15 - Informações da ONG	56
Figura 16 - Escolha de Cadastro.....	56
Figura 17 - Cadastro de Usuário.....	57
Figura 18 - Cadastro de Instituição.....	57
Figura 19 - Cadastro de Instituição (continuação).....	58
Figura 20 - Cadastro de Instituição (imagens selecionadas)	58
Figura 21 - Tela de Login.....	59
Figura 22 - Tela Inicial (logado)	59
Figura 23 - Tela de Favoritas.....	60
Figura 24 - Tela de Favoritas (sem ONGs favoritadas).....	60
Figura 25 - Tela de avaliação	61

Lista de códigos-fonte

Código 1 – Classe ApplicationUser	41
Código 2 – Arquivo de Contexto da classe ApplicationUser	42
Código 3 – Arquivo de Contexto da classe Fotos	42
Código 4 – Obtenção da latitude e longitude pela Google Maps API	43
Código 5 – Método para obter posição do marcador	44

Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i>
APP	Aplicativo
BPMN	<i>Business Process Model and Notation</i>
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CRUD	<i>Create, Read, Update, Delete</i>
GPS	<i>Global Positioning System</i>
IDE	<i>Integrated Development Environment</i>
IOS	<i>iPhone Operating System</i>
IoT	<i>Internet of Things</i>
NFEF	Requisito Não Funcional de Eficiência
NFIM	Requisito Não Funcional de Implementação
NFIN	Requisito Não Funcional de Interoperabilidade
NFPO	Requisito Não Funcional de Portabilidade
NFUS	Requisito Não Funcional de Usabilidade
ONG	Organização Não Governamental
P2P	<i>Peer-to-Peer</i>
PaaS	<i>Platform as a Service</i>
RF	Requisito Funcional
RNF	Requisito Não Funcional
SDK	<i>Software Development Kit</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	<i>Structured Query Language</i>
TCC	Trabalho de Conclusão de Curso
TI	Tecnologia da Informação
UML	<i>Unified Modeling Language</i>

Sumário

Introdução	12
1.1 Objetivos	13
1.1.1 Objetivo geral	13
1.1.2 Objetivos específicos	13
1.2 Estrutura do documento	13
Fundamentação teórica	15
2.1 Computação social	15
2.2 Aplicativos móveis	16
2.3 Geolocalização	16
2.4 ONGs	17
2.5 Caridade	17
Materiais e métodos	19
3.1 Materiais	19
3.2 Métodos	21
Trabalhos relacionados	23
4.1 Revisão bibliográfica	23
4.2 Soluções encontradas	24
4.2.1 Charity Miles	24
4.1.2 Solidarius	25
4.1.3 Dalai Lama Gebetsmühle	26
4.1.4 Tem Açúcar?	26
4.1.5 Para Quem Doar	27
4.3 Comparativo com o Doe+	27
Desenvolvimento	30
5.1 Engenharia de Software	30
5.2 Requisitos do projeto	31
5.2.1 Requisitos Funcionais (RF)	31
5.2.2 Requisitos Não Funcionais (RNF)	35
5.3 Modelagem do Doe+	37
5.3.1 Diagrama de Caso de Uso	37
5.3.2 Modelo de processo de negócio	38

5.3.3 Hierarquia de telas da aplicação	39
5.4 Codificação	40
5.4.1 API	40
5.4.2 Mobile	42
5.4.3 Repositórios.....	44
Conclusão	45
Referências	47
Apêndices	50
Apêndice I – Manual de Deploy da API.....	51
Apêndice II – Manual de Utilização do Doe+	53

1

Introdução

A caridade é um serviço vital na ajuda de populações marginalizadas e em situação de vulnerabilidade socioeconômica. A doação online surge como um meio conveniente para os doadores e os beneficiados, trazendo comodidade e praticidade na hora de realizar essa ação (HOU et al., 2021)

De acordo com Wu & Zhu (2020), o incentivo da participação pública na doação para instituições de caridade se tornou um problema social urgente durante a pandemia da COVID-19. Essa demanda tornou essencial o desenvolvimento de ferramentas que facilitem e auxiliem a doação.

Segundo Moresi et al. (2017), os aplicativos sociais se caracterizam como ferramentas da tecnologia social, e seu objetivo é trazer uma opção alternativa ao desenvolvimento tecnológico focado na produção, consumo e vigilância do indivíduo. Já Franzoni & Silva (2016) definem inovação social como uma solução aos problemas crônicos que não podem ser resolvidos por meio da lógica da sociedade, como escassez de recursos e problemas socioambientais.

É importante não confundir aplicativos sociais com redes sociais. Os aplicativos sociais também podem ser referidos como “aplicativos sérios”, que se refere a todos os tipos de aplicativos que não são desenvolvidos com interesse em entretenimento, e podem dar atenção a assuntos como injustiça social e ativismo (JENSEN, 2011).

A chamada “crise de confiança” é um dos principais problemas quando se vai desenvolver aplicativos sociais. Os usuários precisam confiar nos sistemas que estão usando. Um doador precisa saber que seu dinheiro vai chegar no beneficiado, e o beneficiado precisa saber que vai receber a doação. A tecnologia Blockchain tem grande importância nessa questão, com a propagação das redes Peer-to-Peer (P2P), o que garante a credibilidade do processo (WU; ZHU, 2020).

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo geral deste trabalho é desenvolver uma aplicação móvel que facilite a busca por instituições de caridade com base em geolocalização.

1.1.2 Objetivos específicos

- Realizar uma revisão da literatura sobre tecnologias sociais;
- Analisar trabalhos anteriores relacionados ao tema;
- Realizar um comparativo entre a solução desenvolvida e as existentes;
- Descrever o processo de desenvolvimento da API e da aplicação móvel;
- Disponibilizar a API;
- Discutir os resultados obtidos.

1.2 Estrutura do documento

O texto deste trabalho está organizado em mais quatro Seções além desta introdução. Os tópicos a seguir descrevem o conteúdo dessas Seções.

- Seção 2 – Fundamentação teórica: descreve todo o referencial teórico necessário para a compreensão e a elaboração do trabalho, definindo termos como Computação Social, Geolocalização, ONGs e Caridade.
- Seção 3 – Metodologia: descreve os materiais e métodos utilizados na pesquisa.
- Seção 4 – Trabalhos relacionados: descreve detalhadamente as soluções encontradas no mercado e mostra uma comparação entre estas e a solução desenvolvida.
- Seção 5 – Desenvolvimento: traz toda a parte de engenharia de software aplicada ao projeto, como levantamento de requisitos funcionais e não- funcionais, modelagem de diagramas, protótipo de telas e detalhes sobre a codificação.

- Seção 6 – Conclusão: traz os resultados obtidos pela pesquisa e os trabalhos a serem desenvolvidos no futuro.

2

Fundamentação teórica

Nesta Seção, são apresentados os assuntos que servem de base para o trabalho. A Seção 2.1 define e mostra a importância da Computação Social; a Seção 2.2 traz o conceito de aplicativos móveis; na Seção 2.3, é estudado o termo Geolocalização; a Seção 2.4 mostra os significados do termo ONG e a Seção 2.5 faz um breve estudo sobre o termo caridade e sua importância.

2.1 Computação social

O termo Computação Social descreve qualquer tipo de aplicação computacional em que um software serve como intermediário para uma relação social, como em interações via redes sociais ou serviços de apoio mútuo (SCHULER, 1994).

Chamoso et. al (2019) definem a Computação Social como a facilitação computacional para estudos e dinâmicas sociais, assim como o desenvolvimento de tecnologias que consideram o contexto social. A internet produziu um novo elemento social que permite a colaboração e o compartilhamento de interesses entre usuários, o que foi batizado pela empresa norte-americana *O'Reilly Media* como “Web 2.0”, e a partir deste conceito, surgiu o que hoje é conhecido como Computação Social.

De acordo com Erickson (2019), apesar dos estudos nessa área serem realizados desde a década de 60, a Computação Social surgiu entre as décadas de 90 e 2000. Nessa época, os sistemas digitais deixaram de ser utilizados apenas para atividades colaborativas, como conversas on-line, e passaram a processar todo o conteúdo gerado por meio de interações sociais e a retornar os resultados desse processamento.

Já segundo Wang et al (2007), a ideia de Computação Social existe desde a década

de 40, quando o estudioso Vannevar Bush concebeu um dispositivo de memória e comunicação chamado *memex*. Ele também propôs ideias de socialização e trabalho colaborativo via dispositivos.

A ideia de Computação Social também abre espaço para discutir sobre Computação Comunitária, que é definida como uma tecnologia que facilita a disseminação de informações, discussões e atividades conjuntas entre serviços públicos e privados de uma sociedade com seus indivíduos, preocupando-se com o desenvolvimento econômico e social da região em que é aplicada (CARROLL, 2001).

2.2 Aplicativos móveis

De acordo com Barra et al. (2017), aplicativos móveis são conceituados como um conjunto de ferramentas desenhado para realizar tarefas e trabalhos específicos. Esses aplicativos visam atender o acesso dos usuários à informação e ao conhecimento, sem restrição de tempo e espaço.

Os aplicativos móveis têm facilitado a vida das pessoas, ampliando o impacto da tecnologia na sociedade e permitindo a expansão da flexibilidade para vários tipos de relações, como um meio de automatização de diversas tarefas, até então realizadas manualmente, o que desperta o interesse de muitos para o desenvolvimento de aplicativos que solucionem problemas do cotidiano (MAIA & MARIN, 2021).

De acordo com Pressman & Maxim (2016), o projeto de aplicativos móveis abrange atividades técnicas e não técnicas: estabelecer a percepção e a aparência do aplicativo, criar o layout estético da interface do usuário, estabelecer o ritmo da interação do usuário, definir a estrutura arquitetural geral, desenvolver o conteúdo e a funcionalidade residentes na arquitetura e planejar a navegação que ocorre no aplicativo.

Esse tipo de projeto permite que seja criado um modelo que pode ser avaliado em termos de qualidade e aperfeiçoado antes de os códigos e conteúdos serem gerados. É no projeto que se estabelece a qualidade de um aplicativo móvel (PRESSMAN; MAXIM, 2016).

2.3 Geolocalização

A geolocalização ou georreferenciação, é um recurso utilizado para rastrear um

indivíduo ou objeto a partir de sua posição geográfica com base em coordenadas via satélite emitidas por sinais de internet, radiofrequência e GPS. É empregada a partir de um conjunto de *hardware*, *software* e dados geográficos como latitude e longitude (CUNHA, 2019).

De acordo com Leszczynski (2019), a geolocalização se caracteriza por um fenômeno sociotécnico que está associado à proliferação contemporânea de *Big Data* espacial e à consciência posicional de objetos inteligentes e dispositivos que participam da Internet das Coisas (IoT) por meio de tecnologias como *WiFi*, *Bluetooth* e redes móveis.

Os sistemas de geolocalização são baseados principalmente na tecnologia GPS, mas recentemente, com o desenvolvimento de sistemas de redes sem fio de dados e telefonia celular, tornou-se possível posicionar geograficamente um dispositivo desde que ocorra a conexão por meio de um desses sistemas (BRUNET; FREIRE, 2010).

2.4 ONGs

O termo “organização não governamental” tem caráter polissêmico, já que é apropriado para definir diferentes atores sociais. Nas últimas décadas, o termo ONG se difundiu e tem sido usado para denotar um conjunto heterogêneo de organizações (COSTA; SILVA; BONAN, 2011).

Segundo Landim (2002), a sigla ONG já se tornou uma palavra, e o conceito situa-se entre a caridade pessoalizada e a ação pública governamental, mas não confundindo-se com nenhuma delas. Uma forma de pensar seu significado é considerar a ONG enquanto categoria construída socialmente. O termo vem sendo utilizado no contexto brasileiro para designar um conjunto de organizações da sociedade civil com características específicas.

2.5 Caridade

Van der Linder (2011) descreve uma ação de caridade, como o desejo de doar tempo, dinheiro ou recursos para uma organização sem fins lucrativos.

De acordo com Hou et al. (2009), as razões pelas quais os indivíduos estão dispostos a doar para instituições sem fins lucrativos foram examinadas empiricamente de uma perspectiva ampla em áreas como sociologia, psicologia, antropologia e economia. Variáveis como idade, sexo, renda, classe social, normas sociais e grau de convicção religiosa mostraram um impacto

no comportamento de doação.

Um caminho promissor para as instituições de caridade aumentarem a eficiência da arrecadação de fundos com pouco ou nenhum custo adicional são as campanhas na esfera digital, uma plataforma de filantropia em rápido crescimento (ERCEG et al., 2018).

3

Materiais e métodos

Esta Seção apresenta os materiais e métodos utilizados para a realização deste trabalho. A Seção 3.1 mostra as tecnologias utilizadas no desenvolvimento da solução e a Seção 3.2 apresenta os métodos adotados no trabalho.

3.1 Materiais

A seguir, são discutidos os *frameworks*, ferramentas, linguagens de programação, banco de dados e especificações utilizadas para o desenvolvimento do software.

O aplicativo foi desenvolvido utilizando o *framework Flutter* através do ambiente de desenvolvimento integrado *Visual Studio Code* (VSCode), e testado no emulador de *Android* do *Android Studio* e em um dispositivo físico do modelo *Pocophone F1*, da marca *Xiaomi*. Foi utilizado o banco de dados *PostgreSQL* hospedado no *Heroku*. O aplicativo se comunica com uma API desenvolvida em Asp.NET Core 6.0, que também está hospedada no *Heroku* por meio de um *Container Docker*. As definições dos termos apresentados estão abaixo.

Flutter é um *framework* de código aberto da linguagem de programação *Dart* desenvolvido pelo Google. Com ele é possível desenvolver aplicativos híbridos, que funcionam tanto no sistema operacional *Android*, quanto no *iOS*, utilizando um único código. Além de desenvolvimento para dispositivos móveis, o *Flutter* também pode ser utilizado para desenvolvimento de páginas web (FLUTTER, 2022).

O *Visual Studio Code* é um ambiente de desenvolvimento integrado (IDE) de código aberto com suporte a diversas linguagens desenvolvido pela *Microsoft*. Sua interface simples e interativa permite uma rápida codificação e o software conta com ferramentas para

versionamento com Git, refatoração de código e complementação inteligente, além de dispor de uma grande quantidade de extensões que tornam o desenvolvimento mais eficiente (VISUAL STUDIO, 2022).

O Android Studio é um IDE voltado para o desenvolvimento Android desenvolvido pelo Google. O software conta com uma extensa biblioteca de emuladores de dispositivos móveis e que podem ser utilizados com outros IDEs, como o VSCode (ANDROID STUDIO, 2022).

PostgreSQL é um sistema gerenciador de banco de dados (SGBD) objeto relacional de código aberto. É um dos SGBDs mais avançados da atualidade, contando com recursos como consultas complexas, integridade transacional e linguagem procedural (WIKIPÉDIA, 2022).

Heroku é uma plataforma de nuvem como um serviço (PaaS) que suporta várias linguagens de programação. A plataforma permite hospedagem, configuração, testagem e publicação de projetos virtuais na nuvem. Também é possível configurar bancos de dados remotos pelo Heroku. O suporte ao PostgreSQL pela plataforma é gratuito até o dia 28 de novembro de 2022, de acordo com um aviso enviado aos usuários pelo próprio Heroku, mostrado na **Figura 1** (HEROKU, 2022).

Figura 1 - Aviso de fim do suporte ao deploy gratuito pelo Heroku



Dear Customer,

Thank you for being a Heroku user. You are receiving this email because our records show you are currently using our Heroku free plans. If you have already upgraded your services, please disregard this message.

As we [previously communicated](#), starting November 28th, 2022, free Heroku Dynos, free Heroku Postgres, and free Heroku Data for Redis® will no longer be available. You must take action to continue to operate your applications on our service and prevent the deletion of your free databases.

Beginning on November 28th, 2022, the following changes will occur for existing free Heroku product plans:

- Free Heroku dynos will be converted to Eco dynos and spun down. Any Heroku Scheduler jobs configured to use free dynos will fail. You must [reconfigure](#) any existing Scheduler jobs that use free dynos to use another [dyno type](#).
- For customers paying by credit or debit card, existing free Heroku Postgres and free Heroku Data for Redis® databases will be queued for deletion.
- For Enterprise accounts, beginning on November 28th, free Heroku Postgres and free Heroku Data for Redis® databases will automatically be upgraded to Mini data plans.

To prevent the disruption to your apps on free dynos and the deletion of free databases, you must upgrade to paid plans before November 28th, 2022. Subscribing to Eco automatically converts free dynos for all your personal apps to Eco, along with any Scheduler jobs that were using free dynos. [Visit the Knowledge Base for instructions](#).

For customers paying via a credit or debit card, you will begin accruing charges on December 1st, 2022 for the new Eco dyno and Mini data plans, so there's no reason to wait to upgrade. For Enterprise users you will be billed per the terms of your contract.

Fonte: Heroku, 2022

Asp.NET Core é um framework da linguagem de programação C# desenvolvido pela Microsoft. É uma tecnologia de código aberto multiplataforma que permite criar serviços para web, além de aplicativos de Internet das Coisas (MICROSOFT, 2022).

Docker é uma plataforma de código aberto que permite e facilita a criação de ambientes virtuais isolados. Com ele é possível “empacotar” uma aplicação dentro de um Container, tornando-a portátil para qualquer servidor que tenha suporte ao Docker (DOCKER, 2022).

Como o Heroku não tem suporte direto para Asp.NET, a API desenvolvida para o Doe+ foi empacotada em um Container Docker, e este foi hospedado no Heroku.

3.2 Métodos

Para o presente trabalho, foi feita uma pesquisa na literatura para encontrar projetos similares e auxiliar no levantamento de requisitos, modelagem de diagramas e criação das telas do aplicativo.

O software foi desenvolvido utilizando a metodologia ágil SCRUM, que é um método usado para orientar as atividades de desenvolvimento dentro de um processo que incorpora atividades como análise, projeto, evolução e entrega (PRESSMAN; MAXIM, 2016).

4

Trabalhos Relacionados

Esta Seção apresenta a revisão da literatura realizada para este trabalho; uma descrição mais detalhada sobre as soluções encontradas no mercado que possuem funcionalidades semelhantes ao Doe+; e um comparativo entre a solução desenvolvida e as existentes.

4.1 Revisão bibliográfica

A pesquisa para as referências bibliográficas foi realizada na base de dados Periódicos Capes. As expressões utilizadas foram “social computing” e derivados, que retornaram 303.899 resultados, e “donation application” e derivados, que retornaram 61.265 resultados. Foram buscados resultados em inglês e português, e aplicados os filtros de qualidade, ano de publicação e ocorrência dos termos. Nesta Seção, são apresentados os principais resultados encontrados.

O trabalho de Wang et al. (2007) mostra que a Computação Social é caracterizada como um novo paradigma do desenvolvimento tecnológico, e que se tornou um tema central em vários campos da tecnologia da informação e comunicação. É um campo de estudo que pode ser utilizado não só por pesquisadores, mas também por desenvolvedores e fabricantes de jogos e software.

Choi & Kim (2016) fizeram um estudo acerca da doação para instituições de caridade via aplicativos móveis. A maior distinção entre aplicativos móveis de doação e outros tipos de doações é que os usuários podem indiretamente doar dinheiro para caridade sempre que usarem as funções de aplicativos móveis de doação, como ver anúncios ou responder a

pesquisas. Alguns aplicativos também cobram um valor simbólico para sua instalação e o valor arrecadado é doado para a caridade.

A pandemia da COVID-19 trouxe a necessidade de realizar atividades mantendo o distanciamento social. Kim & Kim (2022) estudam o crescimento das doações online na Coreia do Sul, principalmente por meio de dispositivos móveis, inclusive no contexto da pandemia. De 2013 a 2019, o número de doações online na Coreia do Sul subiu de 3% do total de doações pessoais, para 23,2%. Os autores apontam os benefícios das doações online como os principais motivos pela escolha dos usuários. Estes preferem doações online para evitar locomoção e enfrentamento de filas ou grandes aglomerações. As redes sociais também têm um papel importante nesses números, já que alguns usuários optam por métodos participativos em redes sociais para realizar suas doações.

O advento da blockchain também contribuiu para o aumento das doações online. Muitos doadores que compraram as principais criptomoedas quando estas ainda custavam pouco e obtiveram um lucro generoso, ou que ainda tentam obter receita por meio da tecnologia, optam por usar seus lucros para realizar doações (KIM & KIM, 2022).

4.2 Soluções encontradas

A busca por soluções semelhantes no mercado foi realizada em lojas de aplicativos como a Google Play Store, para Android, e a Apple Store, para iOS. Também foram realizadas buscas na própria ferramenta de pesquisa do Google, para encontrar possíveis soluções que não estejam disponíveis para dispositivos móveis.

Por fim, foram realizadas pesquisas na base de dados Periódicos Capes, para encontrar soluções que foram documentadas em trabalhos acadêmicos. Foi utilizada a *string* de busca “donation application” e derivados, que retornaram 61.265 resultados. Foram buscados resultados em inglês e português, e aplicados os filtros de qualidade, ano de publicação e ocorrência dos termos. Os resultados mais relevantes encontrados em todas as buscas são apresentados nessa Seção.

4.2.1 Charity Miles

O aplicativo Charity Miles ([Charity Miles](#)), mostrado na **Figura 2**, é uma solução gratuita que incentiva os usuários a praticar ciclismo, caminhadas ou corridas, e converte as

milhas percorridas em dinheiro para instituições de caridade. Graças à sua conveniência, facilidade de uso e ajuda mútua, aplicativos sociais como o Charity Miles estão se tornando cada vez mais populares (CHOI; KIM, 2016).

Figura 2 - Telas do Charity Miles



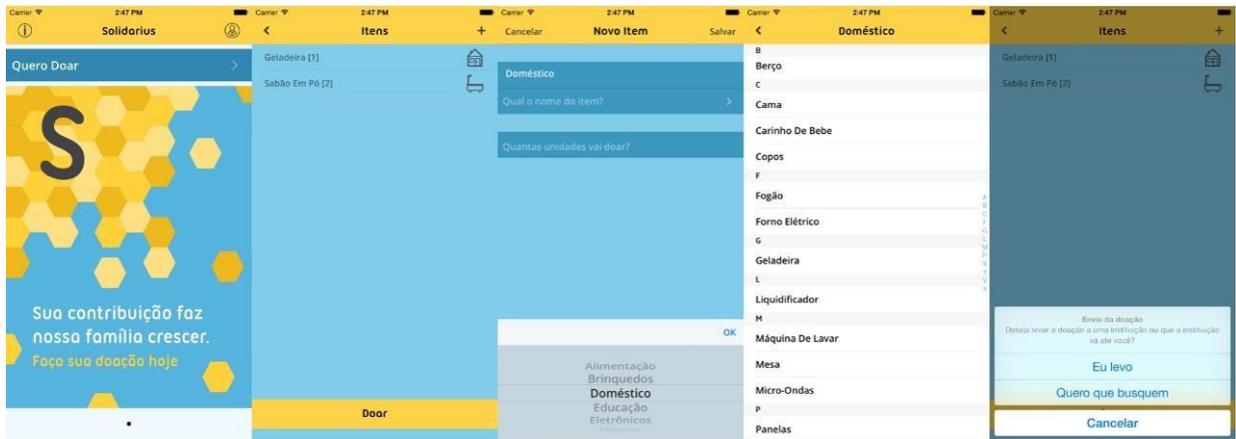
Fonte: Charity Miles, 2022

A doação via dispositivos móveis permite uma ação direta e sem complicações, com o usuário pesquisando o tipo de doação que quer fazer e uma comunicação sendo estabelecida entre doador e receptor. Comparado com sistemas de doação comuns ou sites da web, a doação móvel reduz as barreiras e, portanto, facilita a inclusão social, independentemente da idade, sexo e limitações físicas (CHOI; KIM, 2016).

4.1.2 Solidarius

Moresi et. al (2017) realizaram um estudo sobre o impacto da computação social e uma análise da motivação para o uso do aplicativo Solidarius ([Solidarius by BEPiD - Universidade Catolica de Brasilia](#)), mostrado na **Figura 3**, uma solução para dispositivos móveis desenvolvida na Universidade Católica de Brasília, com o objetivo de promover doações para instituições de caridade. Foram realizadas entrevistas com voluntários que testaram o aplicativo, e concluiu-se que a solução traz benefícios e desafios em sua utilização. O principal desafio é a capacitação dos voluntários para atuar na administração do aplicativo, especialmente colaboradores da terceira idade.

Figura 3 - Telas do Solidarius



Fonte: Solidarius, 2022

A partir dos prós e contras relatados nas entrevistas, a equipe responsável pela pesquisa criou uma tabela que relaciona os meios de doação tradicionais e a doação pelo aplicativo Solidarius. As vantagens do aplicativo incluem: anonimato, processo rápido e direto de doação, a não exigência de locomoção e, conseqüentemente, gastos logísticos (MORESI et. al, 2017).

4.1.3 Dalai Lama Gebetsmühle

O primeiro aplicativo social com o objetivo de realizar doações para instituições de caridade foi desenvolvido em 2009 na Alemanha. Após uma visita do Dalai Lama ao país, um *app*, chamado “Dalai Lama Gebetsmühle”, com frases aleatórias do líder religioso, foi lançado na Apple Store no valor de 79 centavos de dólar. Quatro dias após seu lançamento, tornou-se o terceiro *app* mais baixado da sua categoria, resultado alcançado devido à importância da visita do líder para os meios políticos e religiosos do país. A receita gerada foi doada para orfanatos em Uganda e no Tibete (JENSEN, 2011).

4.1.4 Tem Açúcar?

No trabalho de Brito & Goia (2020), é feito um estudo do aplicativo “Tem Açúcar?”, lançado no Brasil em 2014. A solução tem como objetivo promover o consumo consciente, e permite a troca de objetos entre vizinhos de grandes cidades. É a plataforma de compartilhamento de bens mais difundida do Brasil, com mais de 180 mil usuários.

4.1.5 Para Quem Doar

O grupo Globo criou uma iniciativa chamada Para Quem Doar ([Para Quem Doar](#)), mostrado na **Figura 4**, que tem o objetivo de divulgar organizações de caridade de todo o Brasil e auxiliar nas doações para essas instituições. O programa está disponível apenas pelo site, e aceita indicações de novas organizações pelos usuários, mas por ter pouca divulgação, tem pouquíssimas instituições cadastradas (GLOBO, 2022).

Figura 4 - Tela inicial do Para Quem Doar



Fonte: Globo, 2022

4.3 Comparativo com o Doe+

O Doe+ foi desenvolvido com o objetivo de dar maior visibilidade para instituições de caridade, de acordo com a geolocalização do usuário, podendo ser usado em qualquer lugar do mundo.

Todos os aplicativos encontrados no mercado possuem algum tipo de desvantagem ou diferença de funcionamento em relação à solução desenvolvida neste trabalho. O Charity Miles realiza doações para instituições cadastradas de acordo com a quantidade de milhas percorridas por cada usuário, e só faz doações para instituições de outros países, a grande maioria sendo desconhecida no Brasil. Além de o usuário não ter nenhuma relação direta com as instituições, ele só está convertendo suas milhas em doações, e não realizando o ato diretamente.

Existe um aplicativo brasileiro semelhante ao Charity Miles, chamado Kilômetro Solidário (KMS), que tem as mesmas funções, mas as doações são feitas para instituições brasileiras, mais conhecidas pelos usuários. Uma das instituições beneficiadas é o Instituto da ativista da causa animal Luísa Mell (KMS, 2022).

O Dalai Lama Gebetsmühle tem o problema da localização, já que encontra-se disponível apenas na Alemanha, e realiza doações para instituições específicas dos países de Uganda e Tibete, por meio do valor pago para download do aplicativo.

No Brasil, o aplicativo que mais se destaca é o Solidarius, que não tem o problema da localização, entretanto, a solução está disponível apenas para iOS, e suas doações são destinadas apenas ao grupo de assistência social Sociedade São Vicente de Paulo, uma instituição vinculada à Igreja Católica composta por 33 conselhos metropolitanos em todo o país. A associação ajuda famílias carentes em todo o território nacional, mas não realiza ações em auxílio a outros grupos em vulnerabilidade socioeconômica.

O programa da Rede Globo, Para Quem Doar, não está disponível para dispositivos móveis. Há uma quantidade pequena de instituições cadastradas, já que a adição de novas instituições vem da indicação feita pelos próprios usuários, e o programa não tem a divulgação devida. O processo de doação também poderia ser simplificado. As doações são feitas por diferentes sistemas, como o Apoia.se, mudando de acordo com a instituição selecionada. Apenas algumas organizações possibilitam doação direta via transferência bancária ou carteiras digitais.

O Doe+ foi desenvolvido utilizando o *framework* Flutter, que possibilita a criação de aplicações nativas para iOS e Android com o mesmo código-fonte. Além da maior abrangência, o aplicativo utiliza geolocalização para mostrar as instituições cadastradas mais próximas do usuário.

As instituições são cadastradas com um tipo, que serve para filtrá-las no mapa, permitindo que o usuário encontre com mais facilidade a instituição que ele procura.

A geolocalização permite que o usuário possa apoiar instituições que atuem na cidade em que ele reside, dando mais visibilidade a pequenas ONGs que fazem trabalhos tão importantes quanto as mais conhecidas.

Por questão de segurança, as doações não são feitas diretamente pelo aplicativo. Para garantir que a doação chegue ao destinatário correto, o aplicativo mostra os dados bancários, incluindo chave Pix e usuário do PicPay, se houver. Além dessas informações, há também dados para contato, como telefone e redes sociais, caso o usuário queira confirmar que as informações mostradas são verídicas.

O Doe+ também permite que o usuário salve uma ou mais instituições entre as suas favoritas, e faça uma avaliação de acordo com critérios objetivos. A média aritmética das avaliações de uma instituição é mostrada na página dos dados desta.

5

Desenvolvimento

Esta Seção apresenta os conceitos de Engenharia de Software, seguidos pelos detalhes do desenvolvimento do Doe+, incluindo a lista de Requisitos Funcionais e Não Funcionais, o Diagrama de Caso de Uso, o Modelo de Processo de Negócio, a Hierarquia de Telas, as especificações técnicas e o protótipo de telas.

5.1 Engenharia de Software

Segundo Sommerville (2011), a Engenharia de Software é uma disciplina de engenharia que se preocupa com todos os aspectos da produção de um software. Seu objetivo é apoiar o desenvolvimento profissional de software, mais do que a programação individual. Esse conceito inclui técnicas que apoiam especificação, projeto e evolução de programas, que normalmente não são relevantes para o desenvolvimento de software pessoal.

O processo de software é um conjunto de atividades, ações e tarefas realizadas na criação de algum artefato. Uma metodologia de processo para Engenharia de Software compreende cinco atividades: comunicação, planejamento, modelagem, construção e entrega. Essas atividades podem ser utilizadas para o desenvolvimento de programas pequenos e simples e para a engenharia de grandes e complexos sistemas baseados em computador (PRESSMAN; MAXIM, 2016).

Existem modelos de desenvolvimento que podem ser aplicados num processo de Engenharia de Software. O modelo mais comum adotado nesse tipo de projeto é o modelo em cascata, cujo nome foi dado com base no encadeamento de uma fase do processo em outra. Esse modelo é composto por três principais estágios de um projeto de software: análise e definição de requisitos; projeto de sistema e software; e implementação e teste unitário

(SOMMERVILLE, 2011).

5.2 Requisitos do projeto

Pressman & Maxim (2016) definem os requisitos de um projeto de software como um conjunto de tarefas a serem executadas pela solução, que levam a um maior entendimento sobre qual será o impacto do produto, o que o cliente deseja e como os usuários finais vão interagir. Os requisitos podem ser funcionais, que lidam com interfaces de usuário, hardware, software e comunicação; não-funcionais, que abordam necessidades de desempenho, proteção, segurança e qualidade de software; e inversos, que descrevem as ações que o produto não pode realizar.

5.2.1 Requisitos Funcionais (RF)

Os requisitos funcionais (RF) de um sistema descrevem o que ele deve fazer. Para defini-los, é preciso ter algumas informações, como o tipo de software a ser desenvolvido, quem são seus possíveis usuários e a abordagem adotada. Os RF podem descrever especificamente ou não, as funções do sistema, suas entradas e saídas e suas exceções (SOMMERVILLE, 2011).

O **Quadro 1** apresenta o Requisito Funcional número 1 (RF001), que tem como objetivo permitir que o usuário se cadastre no banco de dados do sistema.

Quadro 1 – [RF001] permitir cadastro de usuário

Prioridade:	Importante
Ator(es):	Usuário.
Requisitos associados:	Não há.
Objetivo:	O Sistema deve permitir que o usuário se cadastre no banco de dados.

Fonte: Elaborado pelos autores, 2022

O **Quadro 2** mostra o Requisito Funcional número 2 (RF002), cujo objetivo é

permitir que o usuário busque por instituições de caridade próximas para fazer doações.

Quadro 2 - [RF002] permitir busca por instituições de caridade próximas

Prioridade:	Essencial
Ator(es):	Usuário.
Requisitos associados:	Não há.
Objetivo:	O Sistema deve permitir que o usuário busque por instituições de caridade próximas para fazer doações.

Fonte: Elaborado pelos autores, 2022

O **Quadro 3** descreve o Requisito Funcional número 3 (RF003), que tem como objetivo dispor de um filtro para que o usuário possa decidir para qual tipo de instituição deseja doar.

Quadro 3 - [RF003] filtrar tipos de instituições

Prioridade:	Importante
Ator(es):	Usuário.
Requisitos associados:	RF002.
Objetivo:	O Sistema deve dispor de um filtro para que o usuário possa decidir para qual tipo de instituição deseja doar.

Fonte: Elaborado pelos autores, 2022

O **Quadro 4** mostra o Requisito Funcional número 4 (RF004), cujo objetivo é mostrar os dados da instituição selecionada pelo usuário, contendo endereço, telefone, redes sociais e dados para doação.

Quadro 4 - [RF004] exibir dados das instituições

Prioridade:	Importante
Ator(es):	Sistema e usuário.

Requisitos associados:	RF002.
Objetivo:	O Sistema deve mostrar os dados da instituição selecionada pelo usuário, contendo endereço, telefone, redes sociais e meios de doação.

Fonte: Elaborado pelos autores, 2022

O **Quadro 5** traz o Requisito Funcional número 5 (RF005), em que o objetivo é permitir que o representante de uma instituição cadastre essa instituição para disponibilizar dados de contato e para doação.

Quadro 5 - [RF005] permitir cadastro de instituição

Prioridade:	Importante
Ator(es):	Usuário.
Requisitos associados:	Não há.
Objetivo:	O Sistema deve permitir que o representante de uma instituição cadastre uma instituição para disponibilizar dados como conta bancária e/ou chave pix.

Fonte: Elaborado pelos autores, 2022

O **Quadro 6** mostra o Requisito Funcional número 6 (RF006), do qual o objetivo é mostrar um botão para copiar dados da instituição, como conta bancária e chave pix, para que o usuário possa fazer doações.

Quadro 6 - [RF006] permitir cópia de dados

Prioridade:	Desejável
Ator(es):	Sistema.
Requisitos associados:	Não há.

Objetivo:	O Sistema deve mostrar um botão para copiar dados da instituição, como conta bancária e chave pix, para que o usuário possa fazer doações.
------------------	--

Fonte: Elaborado pelos autores, 2022

O **Quadro 7** mostra o Requisito Funcional número 7 (RF007), que tem como objetivo permitir que o usuário busque por instituições e acesse os dados para fazer doações sem estar cadastrado.

Quadro 7 - [RF007] permitir busca e doação sem cadastro

Prioridade:	Essencial
Ator(es):	Sistema.
Requisitos associados:	Não há.
Objetivo:	O Sistema deve permitir que o usuário busque por instituições e acesse os dados para fazer doações sem estar cadastrado.

Fonte: Elaborado pelos autores, 2022

O **Quadro 8** detalha o Requisito Funcional número 8 (RF008), em que o objetivo é mostrar nome e uma breve descrição da instituição selecionada, antes de levar o usuário à tela de doação.

Quadro 8 - [RF008] exibir informações básicas ao selecionar uma instituição

Prioridade:	Desejável
Ator(es):	Sistema.
Requisitos associados:	RF002
Objetivo:	O Sistema deve mostrar nome e uma breve descrição da instituição selecionada, antes de levar o usuário à tela de doação.

5.2.2 Requisitos Não Funcionais (RNF)

De acordo com Sommerville (2011), os requisitos não funcionais (RNF) são requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários. Eles podem se referir às propriedades do sistema, confiabilidade, tempo de resposta e ocupação de área.

As tabelas abaixo listam os requisitos não funcionais do aplicativo Doe+, classificados por Usabilidade (NFUS), Eficiência (NFEF), Portabilidade (NFPO), Implementação (NFIM) e Interoperabilidade (NFIN).

O **Quadro 9** mostra o Requisito Não Funcional de Usabilidade número 1 (NFUS001), cujo objetivo é ordenar a listagem das instituições de acordo com a proximidade com a localização do usuário.

Quadro 9 - [NFUS001] listar por localização

Prioridade:	Importante
RF associados:	RF002 e RF003
Descrição:	O sistema deve ordenar a listagem das instituições de acordo com a proximidade com o endereço do doador.

Fonte: Elaborado pelos autores, 2022

O **Quadro 10** traz o Requisito Não Funcional de Usabilidade número 2 (NFUS002), em que o objetivo é ter consistência nas cores e termos presentes nos formulários.

Quadro 10 - [NFUS002] consistência na interface do usuário

Prioridade:	Importante
RF associados:	Não há

Descrição:	O sistema deve ter consistência nas cores e termos presentes nos formulários.
-------------------	---

Fonte: Elaborado pelos autores, 2022

O **Quadro 11** detalha o Requisito Não Funcional de Portabilidade número 1 (NFPO001), cujo objetivo é que o sistema seja disponibilizado em versões compatíveis com Android e iOS.

Quadro 11 - [NFPO001] versões multiplataforma

Prioridade:	Essencial
RF associados:	Não há
Descrição:	O sistema deve ser disponibilizado em versões compatíveis para iOS e Android.

Fonte: Elaborado pelos autores, 2022

O **Quadro 12** apresenta o Requisito Não Funcional de Implementação número 1 (NFIM001), que indica que a aplicação deverá ser desenvolvida na linguagem Dart, utilizando o *framework* Flutter.

Quadro 12 - [NFIM001] linguagem de programação

Prioridade:	Importante
RF associados:	Não há
Descrição:	A aplicação deverá ser desenvolvida na linguagem Dart, utilizando o <i>framework</i> Flutter.

Fonte: Elaborado pelos autores, 2022

O **Quadro 13** exhibe o Requisito Não Funcional de Interoperabilidade número 1 (NFIN001), onde consta que a aplicação deverá se comunicar com o banco de dados PostgreSQL, hospedado na plataforma Heroku.

Quadro 13 - [NFIN001] banco de dados

Prioridade:	Importante
RF associados:	Não há
Descrição:	A aplicação deverá se comunicar com o banco de dados PostgreSQL por meio da plataforma Heroku.

Fonte: Elaborado pelos autores, 2022

5.3 Modelagem do Doe+

Na engenharia de software, a modelagem é um “esboço” do projeto, para que se possa ter uma ideia do todo, como qual será o aspecto da solução em termos de arquitetura, como as partes constituintes se encaixarão e outras características. A modelagem é essencial para entender melhor as necessidades do software e o projeto que vai atender a essas necessidades (PRESSMAN; MAXIM, 2016).

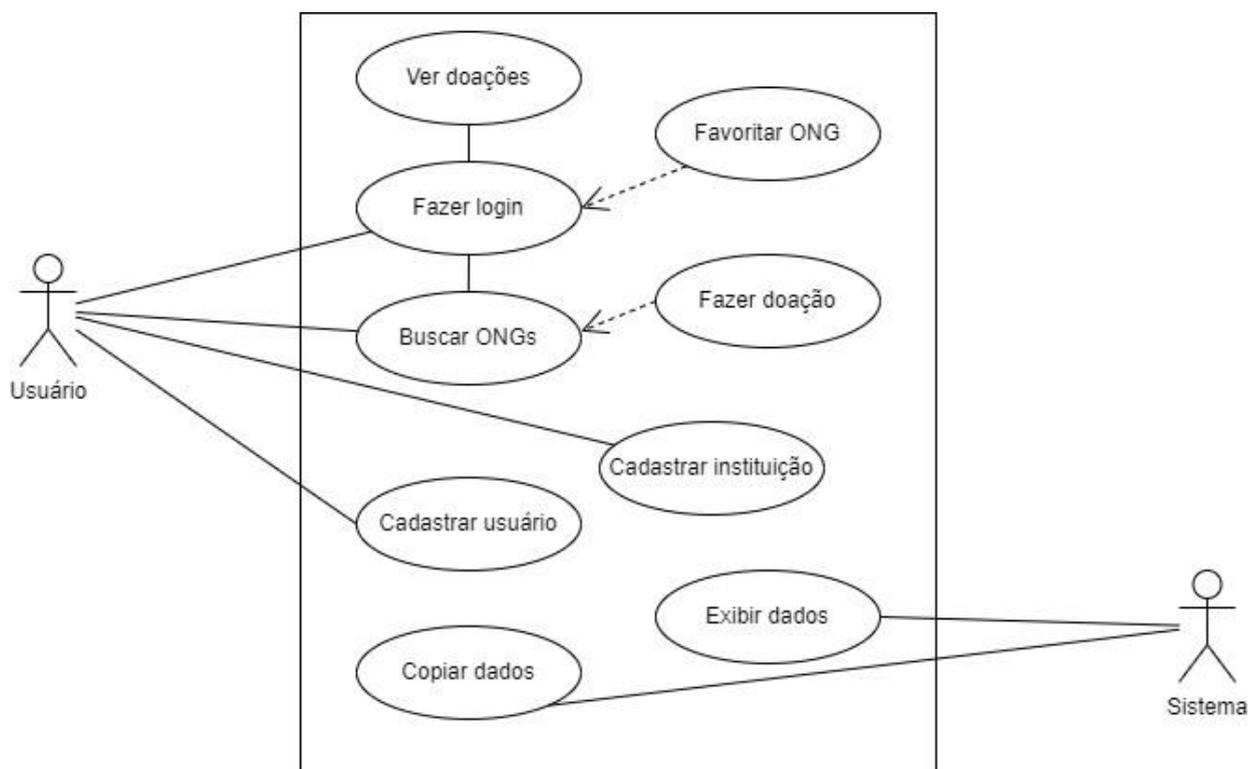
5.3.1 Diagrama de Caso de Uso

De acordo com Pressman & Maxim (2016), a Linguagem de Modelagem Unificada (UML) é uma linguagem-padrão para descrever ou documentar projetos de software. A UML pode ser usada para visualizar, especificar, construir e documentar os artefatos de um sistema de software intensivo. Dessa forma, com o padrão, é possível criar diagramas que auxiliem no desenvolvimento do software.

O diagrama de casos de uso é um dos principais diagramas do padrão UML, a sua função é ajudar a determinar a funcionalidade e as características do software sob o ponto de vista do usuário (PRESSMAN; MAXIM, 2016).

A **Figura 5** ilustra o diagrama de casos de uso do aplicativo Doe+. O relacionamento entre o caso de uso Fazer Login e o caso de uso Favoritar ONG tem tipo *extend*, ou seja, quando o caso de uso Fazer Login for executado, o caso de uso Favoritar ONG poderá ou não ser executado também. O mesmo se aplica para o relacionamento entre o caso de uso Buscar ONGs e o caso de uso Fazer Doação.

Figura 5 - Diagrama de Casos de Uso



Fonte: Elaborado pelos autores, 2022

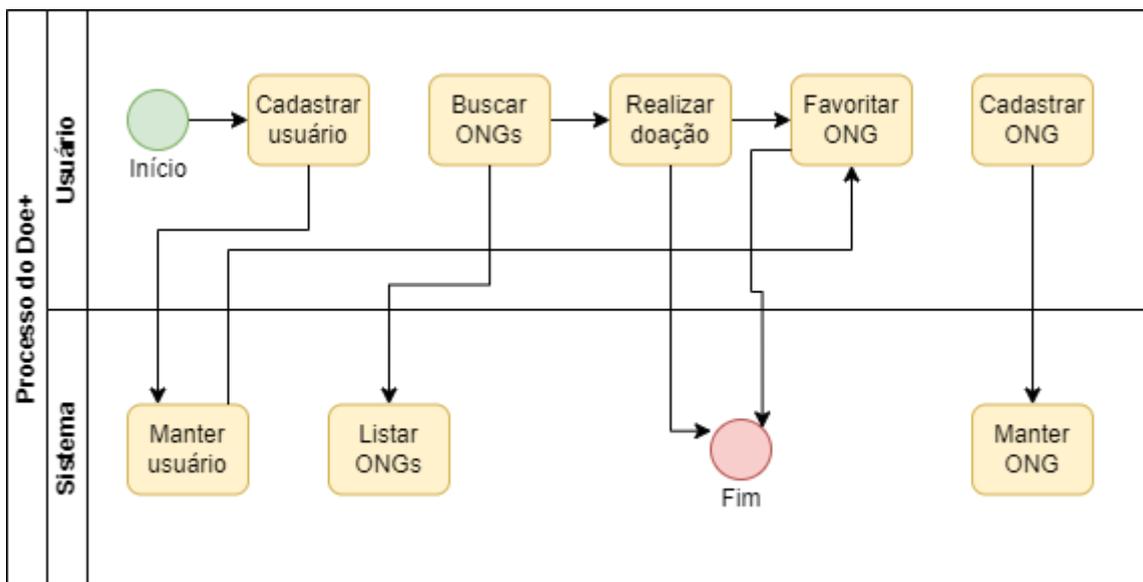
O diagrama representa todas as funções que o usuário, enquanto doador ou enquanto instituição, e o sistema poderão desempenhar no aplicativo. No contexto do Doe+, o usuário não é obrigado a criar uma conta e realizar login para buscar instituições de caridade e fazer doações, mas pode realizar essas ações para salvar uma instituição nos seus favoritos e visualizar as doações feitas.

5.3.2 Modelo de processo de negócio

A Notação e Modelagem de Processos de Negócio (BPMN) é um padrão de modelagem criado para representar os processos graficamente por meio de diagramas. Essa notação possui um conjunto de símbolos e regras que permite modelar diferentes fluxos de processos, com vários níveis de detalhamento (ALMEIDA, 2017).

A **Figura 6** mostra o BPMN do aplicativo Doe+.

Figura 6 - Modelo de Processos de Negócio

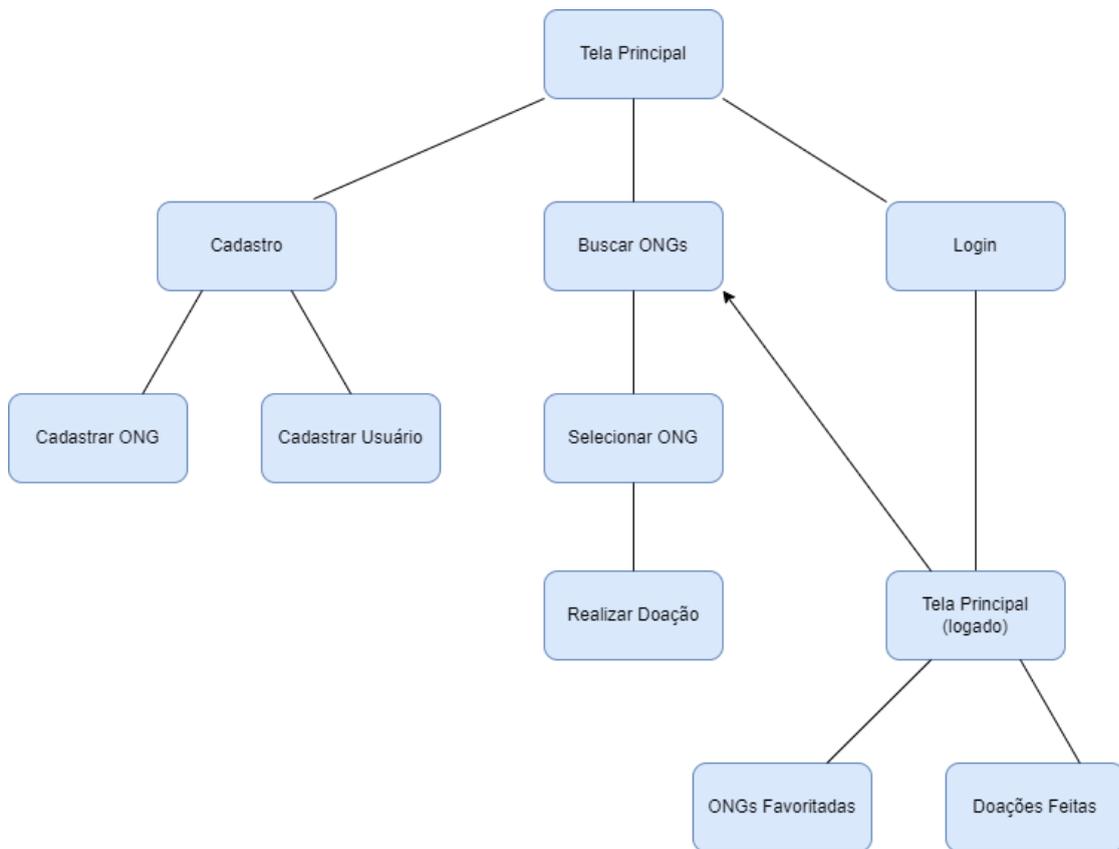


Fonte: Elaborado pelos autores, 2022

5.3.3 Hierarquia de telas da aplicação

A **Figura 7** ilustra a hierarquia de telas do Doe+, para facilitar o entendimento da navegação entre as telas do aplicativo. No Apêndice I, é mostrado um manual de utilização do aplicativo com todas as telas do mesmo.

Figura 7 - Hierarquia de Telas



Fonte: Elaborado pelos autores, 2022

5.4 Codificação

Abaixo é descrito o processo utilizado na codificação da API e do aplicativo, assim como os *frameworks*, classes e padrões de projeto utilizados.

5.4.1 API

O *Back-end* do projeto foi desenvolvido na linguagem C#, através do *framework* de desenvolvimento web Asp.NET Core 6.0. Para a autenticação, foi utilizado *framework Identity* da Microsoft, que cria uma classe única para os usuários, chamada de *ApplicationUser* (**Código 1**), com atributos padrão, mas que pode ser incrementada com outros atributos que o desenvolvedor desejar. Dessa forma, tanto os Usuários quanto as Instituições são armazenados na mesma classe a nível de código e de banco de dados.

Código 1 - Classe ApplicationUser

```
public class ApplicationUser : IdentityUser
{
    public string? Tipo { get; set; }
    [StringLength(500)]
    public string? Descricao { get; set; }
    [StringLength(500)]
    public string? Endereco { get; set; }
    public double Latitude { get; set; }
    public double Longitude { get; set; }
    [StringLength(100)]
    public string? ChavePix { get; set; }
    [StringLength(50)]
    public string? Banco { get; set; }
    [StringLength(5)]
    public string? Agencia { get; set; }
    [StringLength(10)]
    public string? Conta { get; set; }
    [StringLength(100)]
    public string? PicPay { get; set; }

    public string? Site { get; set; }

    public double Avaliacao { get; set; }

    public double AvaliacaoTotal { get; set; }

    public int QtdAvaliacoes { get; set; }

    public virtual ICollection<ApplicationUser>? InstituicoesFavoritas {
get; set; }
}
```

Fonte: Elaborado pelos autores, 2022

Também foi criada uma classe Fotos, que armazena as informações dos arquivos enviados no cadastro de uma Instituição. A classe contém um atributo *InstituicaoId*, para indicar a qual instituição ela pertence.

Para comunicação com o banco de dados, foi utilizado o *Entity Framework*, a principal ferramenta de conexão com banco de dados do Asp.NET. O *Migration* é uma ferramenta do *Entity* que permite sincronizar os modelos do código com as tabelas do banco de dados. Foram criadas duas migrações, uma para as classes do *Identity*, e uma para a classe Fotos, que usam arquivos de contexto diferentes.

A utilização de dois arquivos de contexto se deu porque o *Identity* gera um tipo

específico de contexto, que estende a classe abstrata *IdentityDbContext*, e é útil para as classes do *Identity*, como mostrado no **Código 2**. Para a classe Fotos foi necessário estender a classe abstrata *DbContext*, onde estão os métodos que permitem registrar entidades no banco de dados. O **Código 3** mostra o arquivo de contexto da classe Fotos.

Código 2 - Arquivo de Contexto da classe ApplicationUser

```
public class ApiDoePlusDbContext : IdentityDbContext<ApplicationUser>
{
    public ApiDoePlusDbContext(DbContextOptions<ApiDoePlusDbContext>
options) : base(options)
    {}
}
```

Fonte: Elaborado pelos autores, 2022

Código 3 - Arquivo de Contexto da classe Fotos

```
public class FotosContext : DbContext
{
    public FotosContext(DbContextOptions<FotosContext> options) :
base(options)
    { }

    public DbSet<Fotos> fotos { get; set; }

    public void RegistrarAlterado(object entidade)
    {
        Entry(entidade).State = EntityState.Modified;
    }
}
```

Fonte: Elaborado pelos autores, 2022

Foram criados quatro *Controllers*. A classe *AutenticacaoController* gerencia o cadastro e login de Usuários e Instituições. As classes *UsuariosController* e *InstituicoesController* contêm o restante dos métodos CRUD, responsáveis por consultar, atualizar ou deletar os respectivos registros. Por fim, a classe *FotosController* contém dois métodos de consulta. Um retorna a primeira foto cadastrada para a instituição, que é mostrada quando o usuário escolher uma instituição no mapa. O outro retorna todas as fotos cadastradas para esta instituição, que são exibidas em uma lista quando o usuário ir para a página da instituição.

5.4.2 Mobile

O aplicativo para dispositivos móveis foi desenvolvido utilizando a linguagem Dart e o framework Flutter. A criação do mapa e a geolocalização foram obtidos por meio da API

do Google Maps, que permite a inserção de marcadores para indicar localizações.

A classe *OngsController* é responsável por gerenciar a busca e seleção de instituições. Ela faz requisições para o *Back-end* e gera uma lista de Instituições. Para criar os marcadores, é necessária a latitude e a longitude das instituições. Esses dados são obtidos por meio do *Geocode*, uma ferramenta da *Google Maps API* que retorna um *JSON* contendo todas as informações de um determinado endereço, como mostrado no **Código 4**. Esse *JSON* é convertido em código pela classe *EnderecoJson*, e a partir dela é possível obter os dados do endereço. Dessa forma, o usuário não precisa saber a latitude e a longitude do endereço que está cadastrando. A codificação para obtenção dos marcadores é mostrada no **Código 5**.

Código 4 - Obtenção da latitude e longitude pela Google Maps API

```
salvarDados() async {
  String retorno = "";
  try {
    String url =
      "https://maps.googleapis.com/maps/api/geocode/json?address=${controller.endereco.text}&key=chaveApi";
    final response = await http.get(Uri.parse(url));
    if (response.statusCode == 200) {
      var json = EnderecoJson.fromJson(jsonDecode(response.body));
      var lat = json.results?[0].geometry?.location?.lat;
      var lng = json.results?[0].geometry?.location?.lng;
      controller.registrar(lat, lng, fotos);
      retorno = "Instituição cadastrada com sucesso!";
      ToastGenerico.mostrarMensagemSucesso(retorno);
    } else {
      retorno = "Endereço inválido.";
      ToastGenerico.mostrarMensagemErro(retorno);
    }
  } catch (e) {
    retorno = "Erro! Instituição não cadastrada.";
    ToastGenerico.mostrarMensagemErro(retorno);
  }
  controller.limparDados();
}
```

Fonte: Elaborado pelos autores, 2022

Código 5 - Método para obter posição do marcador

```
getPosicao() async {
  try {
    Position posicao = await _posicaoAtual();
    latitude = posicao.latitude;
    longitude = posicao.longitude;
    _mapsController
      .animateCamera(CameraUpdate.newLatLng(LatLng(latitude,
longitude)));
  } catch (e) {
    erro = e.toString();
  }

  update();
}
```

Fonte: Elaborado pelos autores, 2022

Quando o usuário toca em um marcador no mapa, é mostrado um resumo da instituição selecionada, com uma foto, nome, e endereço, além de um botão para navegar até a página com todos os dados.

Na página da instituição, são mostradas todas as fotos, uma descrição, a avaliação, além de informações de contato e dados bancários. Os dados bancários são copiáveis, e caso a instituição tenha informado uma conta no *PicPay*, há um link que redireciona para o site ou aplicativo.

Quando o usuário realiza o login, ele pode avaliar uma instituição dando de uma a cinco estrelas, além de poder marcar uma instituição como favorita. Após realizar esta última ação, é gerada uma lista, que pode ser acessada a partir da tela inicial, onde estarão todas as instituições favoritas do usuário.

5.4.3 Repositórios

Os códigos da API e do Mobile estão disponíveis nos repositórios do GitHub abaixo.

API: [ApiDoePlus](#)

Mobile: [DoePlusMobile](#)

6

Conclusão

O Doe+ foi desenvolvido com o intuito de facilitar e difundir o ato de doar para instituições de caridade utilizando geolocalização, visando incentivar o apoio a pessoas em situação de vulnerabilidade socioeconômica, e assim entregar um serviço de real importância para a sociedade. A ideia pretende resolver uma problemática comum enfrentada por pessoas que pretendem fazer doações: a falta de conhecimento acerca dessas organizações de caridade e a dificuldade encontrada nas soluções existentes. Há uma falta de divulgação de endereços, ações realizadas, pessoas que são ajudadas por essas instituições, e em alguns casos, o próprio projeto que deveria servir para divulgar as instituições, não tem a divulgação que deveria.

Foi realizada uma pesquisa nas bases de dados acadêmicas para definir termos como computação social e aplicativos sociais, as linhas de estudo da Tecnologia da Informação (TI) nas quais este aplicativo se encaixa, além de encontrar outras soluções presentes no mercado para realizar um comparativo entre a solução desenvolvida e as existentes.

O trabalho também descreveu os detalhes do desenvolvimento do aplicativo, seguindo os conceitos da Engenharia de Software e mostrando as principais classes e métodos criados na codificação.

A partir das pesquisas realizadas, foi possível concluir que o Doe+ apresenta o diferencial de funcionar por geolocalização, permitindo que o aplicativo possa ser utilizado em qualquer parte do mundo, além de permitir que o usuário encontre o tipo exato de ONG que procura, por meio dos filtros, para casos em que o doador deseje contribuir com bens materiais ao invés de quantias em dinheiro.

A solução também permite o estabelecimento de uma relação de confiança entre o usuário e a instituição por meio das avaliações. Com esse método, o usuário terá uma garantia

de que está contribuindo com uma organização confiável que recebeu elogios de outros doadores.

Como o cadastro de ONGs pode ser feito pelo próprio usuário, o aplicativo facilita a adição de novas instituições, permitindo que até o próprio doador possa cadastrar uma instituição que ele conheça.

Para trabalhos futuros, sugere-se a realização de testes e validação do aplicativo. Como o *deploy* da API será desfeito devido ao fim do suporte a *deploy* gratuito pelo *Heroku*, sugere-se também a busca por novos meios de hospedar a API para que o aplicativo consiga se comunicar com ela.

Referências

- ANDROID STUDIO. Android Developers, 2022. Android Studio and SDK Tools. Disponível em <<https://developers.android.com/studio>>. Acesso em: 06/05/2022.
- BARRA, D. C. C., PAIM, S. M. S., DAL SASSO, G. T. M., COLLA, G. W. Métodos para Desenvolvimento de Aplicativos Móveis em Saúde: Revisão Integrativa da Literatura. *Texto & Contexto - Enfermagem*. 2017, v. 26, n. 4.
- BRITO, N., & GOIA, M. Uma rede sociotécnica à luz do paradigma da dádiva: análise do aplicativo de empréstimo de objetos Tem Açúcar? no Rio de Janeiro. *PAAKAT: Revista de Tecnología y Sociedad*. 10. 1-22. 2020.
- BRUNET, K. S., FREIRE, J. *Cultura Digital e Geolocalização: A Arte Ante o Contexto Técnico-Político*, 2010.
- CARROLL, J. Community computing as human - Computer interaction. *Behaviour & IT*. 20. 307-314, 2001.
- CHAMOSO, P., GONZÁLEZ-BRIONES, A., RIVAS, A. et al. Social computing in currency exchange. *Knowl Inf Syst* 61, 733–753, 2019.
- CHARITY MILES. Charity Miles App: Walk, Run, Bike for a Causes, 2022. Disponível em <www.charitymiles.org>
- CHOI, B., & Kim, M. Donation via Mobile Applications: A Study of the Factors Affecting Mobile Donation Application Use, *International Journal of Human– Computer Interaction*, 32:12, 967-974, 2016.
- COSTA, A. M. A. M., SILVA, K. S., BONAN, C. *Organizações Não Governamentais na Área da Saúde da Criança: Revisão da Literatura*, 2008.
- CUNHA, A. M. Prospecção Tecnológica sobre Sistemas de Geolocalização Voltados para Passageiros de Trem. *Cadernos de Prospecção*, [S. l.], v. 12, n. 5, p. 1447, 2019. DOI: 10.9771/cp.v12i5.32929.
- ERCEG, N., BURGHART, M., COTTONE, A., LORIMER, J., MANKU, K., PÜTZ, H., VLASICEK, D., & WILLEMS, M. The Effect of Moral Congruence of Calls to Action and Salient Social Norms on Online Charitable Donations: A Protocol Study, 2018.
- FLUTTER. Flutter: App Development Toolkit, 2022. Página Inicial. Disponível em <<https://www.flutter.dev>>. Acesso em: 06/05/2022.
- FRANZONI, G. B., & SILVA, T. N. *Inovação Social e Tecnologia Social: o caso da Cadeia Curta de Agricultores Familiares e a Alimentação Escolar em Porto Alegre/RS*.

Desenvolvimento Em Questão, 14(37), 353–386, 2016.

GLOBO. Para Quem Doar, 2022. Página Inicial. Disponível em <www.paraquemdoar.com.br>

HEROKU. Heroku – Cloud Application Platform, 2022. Página inicial. Disponível em <<https://www.heroku.com>>. Acesso em: 06/05/2022.

HOU, T., HOU, K., WANG, X., & LUO, X. Why I give money to unknown people? An investigation of online donation and forwarding intention. *Electronic Commerce Research and Applications*, 47. 2021.

JENSEN, O. Use of new mobile media by nonprofits and development of societal mobile apps, 2011.

KIM, D. H., KIM, B. Y. Online Donation Attitude and Satisfaction with Simple Mobile Payments: A Case of the Korean Red Cross. *Societies*, 2022.

LANDIM, L. Experiência Militante: História das Assim Chamadas ONGs, 2002.

LESZCZYNSKI, A. Platform Affects of Geolocation, 2019.

MAIA, J. S., MARIN, H. F. Aplicativos móveis para as sociedades menos favorecidas. *Acta Paul Enferm.* 2021;34.

MORESI, E.A., GODINHO, S.G., MARIZ, R.S., FILHO, M.D., BARBOSA, J.A., LOPES, M.C., JÚNIOR, W.A., & MORAIS, M.A. Tecnologia Social: a doação na perspectiva do aplicativo Solidarius. *RISTI: Revista Ibérica de Sistemas e Tecnologias de Informação*, 1-16, 2017.

PRESSMAN, R. MAXIM, B. Engenharia de Software - Uma Abordagem Profissional. 8 ed. Porto Alegre – AMGH, 2016.

SCHULER, D. Social Computing. *Communications of the ACM*, pp. 28-29. 1994.

SOMMERVILLE, I. Engenharia de Software, 9. ed. — São Paulo: Pearson Prentice Hall, 2011.

VAN DER LINDEN, S. Charitable Intent: A Moral or Social Construct? A Revised Theory of Planned Behavior Model. *Current psychology*, 30(4), 355-374, 2011.

VISUAL STUDIO. Visual Studio Code, 2022. Code Editing – Refined. Página Inicial. Disponível em <<https://code.visualstudio.com>>. Acesso em: 06/05/2022.

WANG F., CARLEY, K. M., ZENG, D. & MAO, W. Social Computing: From Social Informatics to Social Intelligence, *IEEE Intelligent Systems*, vol. 22, no. 2, pp. 79-83, 2007.

WIKIPÉDIA. Wikipédia, a enciclopédia livre. PostgreSQL, 2021. Disponível em <<https://pt.wikipedia.org/wiki/PostgreSQL>>. Acesso em: 06/05/2022.

WIKIPÉDIA. Wikipédia, a enciclopédia livre. SQL, 2022. Disponível em <<https://pt.wikipedia.org/wiki/SQL>>. Acesso em: 06/05/2022.

WU, H., ZHU X., Developing a Reliable Service System of Charity Donation During the

Covid-19 Outbreak. IEEE Access, vol. 8, pp. 154848-154860, 2020.

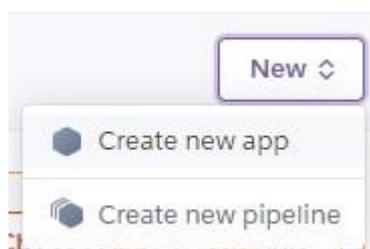
Apêndices

Apêndice I – Manual de Deploy da API

A API do Doe+ foi hospedada no *Heroku* através de um *Contêiner Docker*. A seguiré mostrado um tutorial de como fazer o *deploy*.

Para fazer o *deploy* pelo *Heroku*, é necessário criar uma conta em www.heroku.com, criar um App na plataforma (**Figura 8**), e através da guia *Resources*, adicionar a extensão do *PostgreSQL* ao App.

Figura 8 - Criar novo app no Heroku



Fonte: Heroku, 2022

Ao adicionar a extensão, o *Heroku* criará um banco de dados e disponibilizará as credenciais de acesso a ele por meio da guia *Settings*. Quando a codificação estiver completa e conectada com o banco de dados, é necessário “empacotar” a aplicação utilizando um *Contêiner Docker*.

Para realizar o empacotamento, é necessário baixar o *Docker* em www.docker.com e após instalar, configurar para utilizar uma máquina virtual *Linux* (clicando no ícone do *Docker* na barra de tarefas e depois na opção de “*Switch to Linux Containers*”), que é a única suportada pelo *Heroku*.

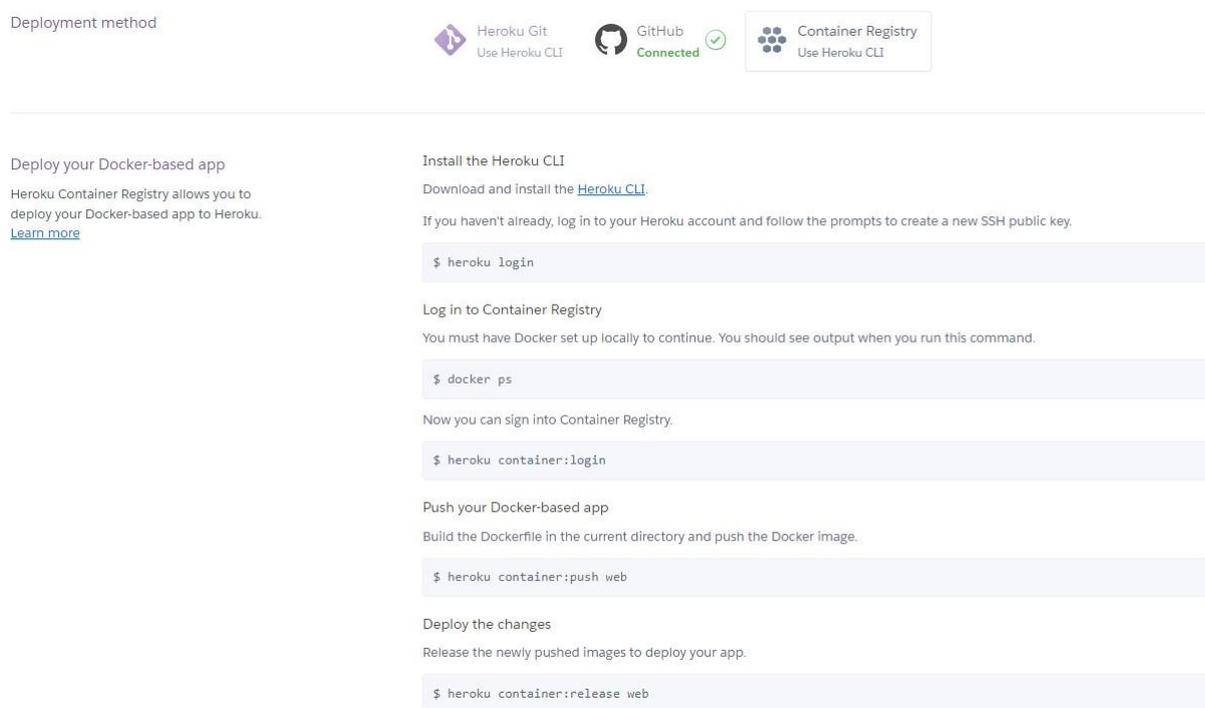
A API do Doe+ já possui o *Dockerfile*, um arquivo de configuração necessário para empacotar a aplicação, portanto, após as configurações iniciais serem feitas, só é necessário executar a aplicação, e o arquivo será lido.

Na aba de *Deploy* do *Heroku* encontra-se um passo a passo para hospedar o Contêiner gerado pela execução da API. Para isso, é necessária a instalação da Interface de Linha de Comando do *Heroku* (*Heroku CLI*), disponível em www.devcenter.heroku.com/articles/heroku-cli.

Após feita a instalação da interface, segue-se o passo a passo descrito na opção

Container Registry na guia *Deploy* da plataforma, mostrado na **Figura 9**.

Figura 9 - Passo a passo para deploy via Contâiner



The screenshot shows the Heroku deployment interface. At the top, under 'Deployment method', three options are visible: 'Heroku Git Use Heroku CLI', 'GitHub Connected', and 'Container Registry Use Heroku CLI', which is highlighted with a blue border. Below this, the 'Deploy your Docker-based app' section provides instructions and terminal commands for installing the Heroku CLI, logging into Container Registry, and pushing the Docker image to Heroku.

Deployment method

Heroku Git Use Heroku CLI GitHub Connected **Container Registry Use Heroku CLI**

Deploy your Docker-based app

Heroku Container Registry allows you to deploy your Docker-based app to Heroku. [Learn more](#)

Install the Heroku CLI

Download and install the [Heroku CLI](#).

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

Log in to Container Registry

You must have Docker set up locally to continue. You should see output when you run this command.

```
$ docker ps
```

Now you can sign into Container Registry.

```
$ heroku container:login
```

Push your Docker-based app

Build the Dockerfile in the current directory and push the Docker image.

```
$ heroku container:push web
```

Deploy the changes

Release the newly pushed images to deploy your app.

```
$ heroku container:release web
```

Fonte: Heroku, 2022

Caso o passo a passo seja executado corretamente, a aplicação será hospedada com sucesso. Após fazer qualquer alteração no código, basta repetir o passo a passo da **Figura 9** para atualizar a aplicação no *Heroku*.

A utilização do Contâiner só é necessária para deploy de aplicações em Asp.NET ou qualquer outra tecnologia que não seja suportada pelo *Heroku*. Em casos de aplicações desenvolvidas em Java, por exemplo, basta conectar o repositório do *GitHub* com o *Heroku* na opção ao lado de *Container Registry*.

É necessário lembrar que o suporte gratuito a hospedagem de aplicações pelo *Heroku* será descontinuado em 28 de novembro de 2022. A API do Doe+ será retirada do ar até essa data. Caso seja preciso hospedar no *Heroku* novamente, deve-se escolher um dos planos pagos disponibilizados na plataforma.

Apêndice II – Manual de Utilização do Doe+

Ao abrir o aplicativo do Doe+, o usuário terá acesso à Tela Inicial (**Figura 10**), onde ele terá a opção de Buscar ONGs, Cadastrar ou Fazer Login. Além disso, o usuário pode abrir um widget tocando no botão no canto superior esquerdo, ou arrastando para a direita, onde terá acesso a informações de contato do desenvolvedor.

Figura 10 - Tela Inicial



Fonte: Elaborado pelos autores, 2022

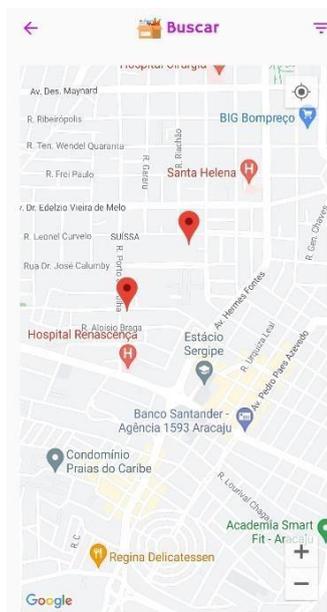
Se o usuário tocar no botão Buscar ONGs, será levado à Tela de Busca de ONGs. Enquanto a requisição à API está sendo executada, é mostrado uma tela de carregamento (**Figura 11**), e ao finalizar a consulta, o mapa é mostrado, onde o usuário pode arrastar para procurar as instituições nas proximidades (**Figura 12**).

Figura 11 - Tela de Carregamento



Fonte: Elaborado pelos autores, 2022

Figura 12 - Mapa



Fonte: Elaborado pelos autores, 2022

No canto superior direito da tela do mapa, o usuário tem acesso ao filtro, que contém alguns tipos comuns de instituições, para que o mapa mostre apenas as instituições do tipo escolhido (**Figura 13**).

Figura 13 - Filtro



Fonte: Elaborado pelos autores, 2022

Ao tocar em um marcador que indica uma ONG cadastrada, o usuário tem acesso a uma tela com algumas informações da instituição (**Figura 14**), para que ele possa confirmar antes de seguir para a tela com todas as informações, onde estão os dados para doação (**Figura 15**).

Figura 14 - Informações da ONG



Fonte: Elaborado pelos autores, 2022

Figura 15 - Informações da ONG



Fonte: Elaborado pelos autores, 2022

Ao tocar no ícone de Favoritar (coração no canto superior direito), ou na nota para dar uma Avaliação, ambos na **Figura 15**, o usuário verá uma mensagem avisando que para realizar essas ações sem fazer Login.

Voltando para a Tela Inicial (**Figura 10**), o usuário pode tocar no botão Cadastrar, onde ele será levado à tela de Escolha de Cadastro (**Figura 16**).

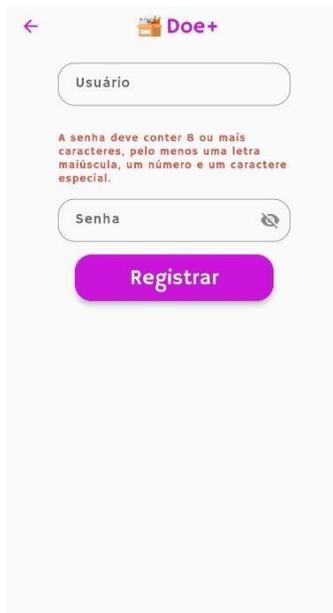
Figura 16 - Escolha de Cadastro



Fonte: Elaborado pelos autores, 2022

Tocando no botão Usuário, o aplicativo redireciona para a tela de Cadastro do Usuário, mostrada na **Figura 17**.

Figura 17 - Cadastro de Usuário



A screenshot of the user registration screen. At the top left is a back arrow, and at the top right is a profile icon labeled 'Doe+'. Below this is a text input field for 'Usuário'. A red warning message states: 'A senha deve conter 8 ou mais caracteres, pelo menos uma letra maiúscula, um número e um caractere especial.' Below the warning is a text input field for 'Senha' with a toggle icon on the right. At the bottom is a purple button labeled 'Registrar'.

Fonte: Elaborado pelos autores, 2022

Tocando no botão Instituição, o aplicativo redireciona para a tela de Cadastro de Instituição, cuja primeira parte é mostrada na **Figura 18**, a continuação é mostrada na **Figura 19**, e a tela com imagens selecionadas é mostrada na **Figura 20**.

Figura 18 - Cadastro de Instituição



A screenshot of the institution registration screen. At the top left is a back arrow, and at the top right is a profile icon labeled 'Doe+'. The title 'Cadastro de instituição' is centered. Below are text input fields for 'Nome da instituição', 'Telefone', 'E-mail', 'Senha', 'Endereço completo', and 'Descrição'. A red warning message is present: 'A senha deve conter 8 ou mais caracteres, pelo menos uma letra maiúscula, um número e um caractere especial.' Below the warning is a purple button labeled 'Escolha o tipo'.

Fonte: Elaborado pelos autores, 2022

Figura 19 - Cadastro de Instituição (continuação)

← Instituição

Preencha pelo menos um dos campos abaixo.

Chave Pix

Banco

Agência

Conta bancária

Usuário do Picpay

Adicionar imagem

Registrar

Fonte: Elaborado pelos autores, 2022

Figura 20 - Cadastro de Instituição (imagens selecionadas)

← Instituição

Banco

Agência

Conta bancária

Usuário do Picpay

image_picker1219611909510262820.jpg
image_picker1425752244804228975.jpg
image_picker673761362951798003.jpg
image_picker7553647739413557191.jpg

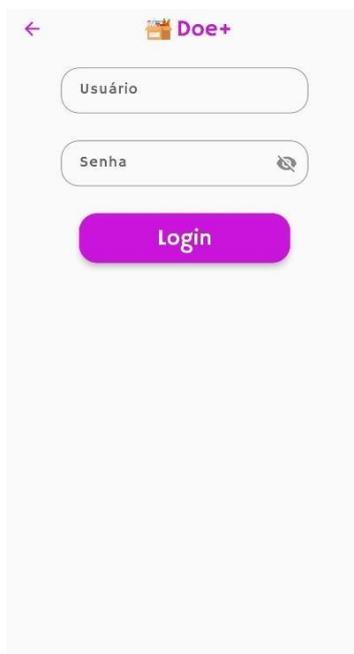
Adicionar imagem

Registrar

Fonte: Elaborado pelos autores, 2022

Tendo feito o cadastro de usuário ou instituição, o aplicativo redireciona de volta para a Tela Inicial (**Figura 10**), e o usuário pode tocar no botão Entrar, para ser levado à Tela de Login (**Figura 21**).

Figura 21 - Tela de Login



Fonte: Elaborado pelos autores, 2022

Ao fazer o login, o usuário é redirecionado para a Tela Inicial, mas com botões diferentes, como o botão de listar as ONGs favoritas e o botão de sair (**Figura 22**).

Figura 22 - Tela Inicial (logado)

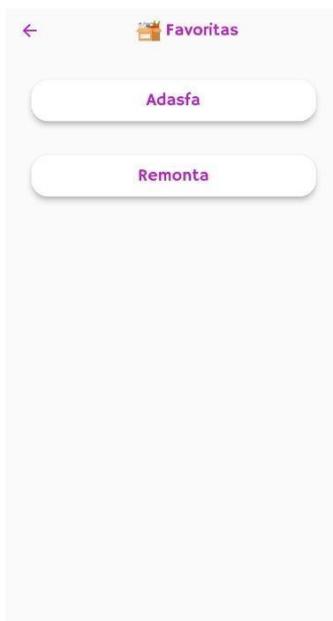


Fonte: Elaborado pelos autores, 2022

Tocando no botão ONGs Favoritas, o usuário é redirecionado para a tela com a listagem das ONGs que ele marcou como favoritas (**Figura 23**). Ao clicar no nome de alguma

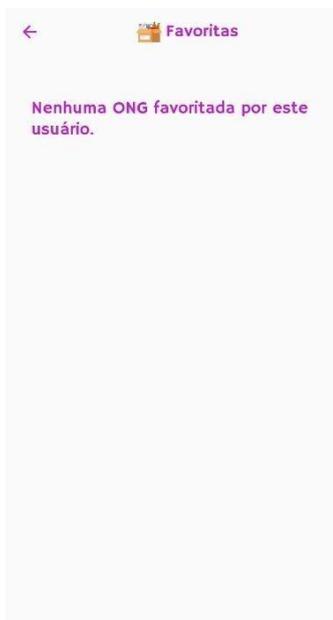
instituição, ele é redirecionado para a tela de informações (**Figura 15**), onde pode fazer doações. Caso o usuário não tenha favoritado nenhuma ONG, a tela mostra uma mensagem de que ele ainda não realizou a ação (**Figura 24**).

Figura 23 - Tela de Favoritas



Fonte: Elaborado pelos autores, 2022

Figura 24 - Tela de Favoritas (sem ONGs favoritadas)

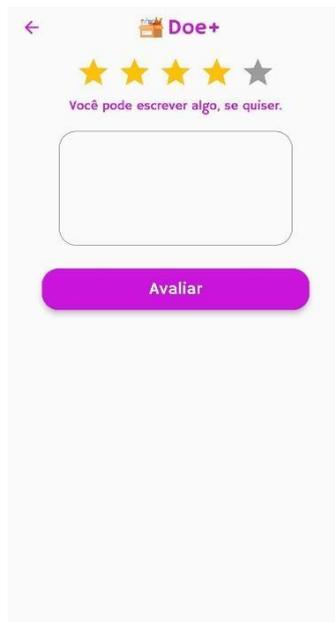


Fonte: Elaborado pelos autores, 2022

Por fim, quando o usuário está logado, na tela de informações da instituição (**Figura 15**), ele pode tocar no ícone do coração para favoritizar a instituição, ou pode tocar na nota da

instituição para enviar uma avaliação, e então será redirecionado para a Tela de Avaliação (Figura 25).

Figura 25 - Tela de avaliação



Fonte: Elaborado pelos autores, 2022