

CENTRO UNIVERSITÁRIO SENAC
SANTO AMARO

Eduardo Henrique Batista Gomes

Eduardo Kobilinski de Araujo

Elias Gonçalves de Albuquerque Júnior

Ezequias Manoel Monteiro Dos Santos

Felipe de Paula Silva

Fernanda Fonseca Scherer

**DESENVOLVIMENTO DE SISTEMAS ORIENTADO A DISPOSITIVOS
MÓVEIS BASEADOS NA WEB**

São Paulo
2022

Eduardo Henrique Batista Gomes
Eduardo Kobilinski de Araujo
Elias Gonçalves de Albuquerque Júnior
Ezequias Manoel Monteiro Dos Santos
Felipe de Paula Silva
Fernanda Fonseca Scherer

**DESENVOLVIMENTO DE SISTEMAS ORIENTADO A DISPOSITIVOS
MÓVEIS BASEADOS NA WEB**

Trabalho de Projeto Integrador: Parte II, desenvolvido como exigência para a obtenção de nota do 4º semestre do curso de Análise e Desenvolvimento de Sistemas, sob orientação do Professor Thomás da Costa.

São Paulo
2022

RESUMO

O projeto DoeAção é uma aplicação desenvolvida para dispositivos móveis baseada em web que visa trabalhar os conceitos de desenvolvimento de projetos, bem como sua implementação, sendo que, o foco é ter uma visão geral da solução e o planejamento para o desenvolvimento, com prototipação e uso de ferramentas de trabalho Ágil, versionamento de projetos, editor gráfico para protótipos e editor de diagramas. O DoeAção consiste em uma aplicação de doações coletivas, um mercado que cresceu nos últimos anos e que vem se difundindo pela facilidade de alcance de campanhas de ajuda com utilização das redes sociais como divulgação. Nesse sentido, desenvolvemos o DoeAção que oferece interface simples e intuitiva para o usuário, com plataforma adaptativa, arquitetada em *microservices*, de fácil manutenção e com possibilidade de melhorias, sem afetar os usuários.

Palavras-chave: Mobile, doação, *microservices*

LISTA DE FIGURAS

FIGURA 1 – Desenvolvimento do Diagrama do Caso de Uso da aplicação no Drawio.	15
FIGURA 2 – Login, cadastro e página inicial.	16
FIGURA 3 – Acessar campanhas e fazer contribuição.	17
FIGURA 4 – Criando campanhas na aplicação.	18
FIGURA 5 – Referências buscadas no Google.	19
FIGURA 6 – Interface da aplicação com os seus fluxos de protótipo entre as telas.	20
FIGURA 7 – Grupo do WhatsApp e definições pendentes de reunião passada.	21
FIGURA 8 – Utilização do Trello para segmentar o desenvolvimento do projeto.	22
FIGURA 9 – Instalação das dependências do NPM na pasta do projeto.	29
FIGURA 10 – Rodando o <code>http-server</code> na porta 8080.	30
FIGURA 11 – Abrindo aplicação no <i>browser</i>	30
FIGURA 12 – Acesso ao banco de dados do projeto.	31
FIGURA 13 – Tela de cadastro.	32
FIGURA 14 – Menu da aplicação.	33
FIGURA 15 – Preenchendo dados da campanha.	34
FIGURA 16 – Revisando dados inseridos.	35
FIGURA 17 – Campanha criada com sucesso.	36
FIGURA 18 – Campanha criada no banco de dados do projeto.	37
FIGURA 19 – Usuário criado no banco de dados do projeto.	37

LISTA DE SIGLAS

CAF – *Charities Aid Foundation*
SUS – Sistema Único de Saúde
UX – *User Experience*
UI – *User Interface*
SaaS – *Software as a Service*
MVC – *Model View Controller*
REST – *Representational State Transfer*
API – *Application Programming Interface*
HTTP – *Hypertext Transfer Protocol*
SOAP – *Simple Object Access Protocol*
POM – *Project Object Model*
HTML – *HyperText Markup Language*
XML – *Extensible Markup Language*
CSS – *Cascading Style Sheets*

SUMÁRIO

1	INTRODUÇÃO	7
2	OBJETIVO	8
3	VISÃO DO PRODUTO	9
3.1	Público Alvo	9
3.1.1	Responsáveis pela Doação	9
3.1.2	Responsáveis pela Criação da Campanha	9
3.1.3	Personas	9
3.1.3.1	Doador	9
3.1.3.2	Campanha	10
3.1.4	Jornada dos Usuários	10
3.2	Dificuldade do Projeto	11
3.3	Missão do Projeto	12
3.4	Benefícios do Projeto	12
3.5	Concorrências	13
3.6	Síntese da Visão do Produto	14
4	PROTÓTIPO	15
4.1	Diagrama de Caso de Uso	15
4.2	Construção da Aplicação	19
5	PLANEJAMENTO	21
6	PADRÃO DE ARQUITETURA MVC	23
7	PADRÃO DE DESENVOLVIMENTO	24
8	TECNOLOGIAS UTILIZADAS	25
8.1	Backend	25
8.1.1	Java	25
8.1.2	Apache Maven	25
8.1.3	Dependências	26
8.1.3.1	Spring	26
8.1.3.2	Banco de Dados	26
8.1.3.3	Outras Ferramentas Utilizadas	27
8.1.4	Cloud Pública Heroku	27
8.2	Frontend	28
8.2.1	HTML, CSS e JavaScript	28
9	DEMONSTRAÇÃO DA SOLUÇÃO PROPOSTA	29
10	CONSIDERAÇÕES FINAIS	38
	REFERÊNCIAS	39

1 INTRODUÇÃO

Doar algo a quem precisa é um gesto de carinho, de respeito e acima de tudo, de empatia.

Com a pandemia em que vivemos recentemente, esse sentimento de querer ajudar e as necessidades de receber ajuda aumentaram muito, segundo (CASTELANI, 2021), em matéria para o site Folha de S. Paulo, um estudo do *Word Giving Index 2021*, índice da organização britânica *Charities Aid Foundation* (CAF) que mede o engajamento em filantropia em 114 países, revela que em 2021 o Brasil avançou 20 posições da sua colocação média dos últimos 10 anos no ranking mundial.

Esse é o reflexo do aumento não só da caridade, mas também, da inflação, das necessidades, das disparidades sociais, da urgência social que vem crescendo cada dia, do desemprego, da falta de acesso à saúde, educação e por aí vai.

Com a finalidade de promover e facilitar essa crescente social, desenvolvemos o DoeAção com o intuito de fazer parte de uma causa social que só retorna o bem, tanto a quem ajuda quanto a quem recebe e tornar cada vez mais comum o gesto de caridade.

O DoeAção tem a necessidade de atender esse generoso crescimento e facilitar as interações entre os usuários com algumas funcionalidades como:

- Poder criar campanhas de necessidades pessoais ou destinada a outra pessoa que esteja precisando de algum tipo de ajuda financeira;
- Poder escolher uma campanha para fazer parte em forma de ajuda financeira a qual mais se identifique;
- Para ambos os casos acima, ter facilidade de propagar as campanhas em redes sociais, onde hoje em dia mais se atinge um número grande de pessoas.

Para os usuários da aplicação, a experiência deve ser fluida e totalmente transparente, confiável e que o único sentimento referente ao uso da aplicação, seja o da satisfação de ajudar e ser ajudado.

2 OBJETIVO

O objetivo do projeto é criar uma ferramenta *web* que possa ser utilizada para promover campanhas de doação e arrecadação coletiva para causas sociais, atreladas a divulgação em redes sociais dos usuários, para promover a comunicação e atender às necessidades de pessoas que precisam de ajuda e assim, usando o DoeAção como veículo de ajuda e comunicação.

3 VISÃO DO PRODUTO

A Visão do Produto é a razão de existência do produto e é o que direciona as decisões de como serão as características desse produto. Será utilizado para tomada de decisões pelo time de desenvolvimento e entender as necessidades e prioridades para o desenvolvimento do produto (TORRES, 2022).

Para compor Visão do Produto, foram definidos: o público alvo, as dificuldades do projeto, a missão, benefícios do produto e concorrências.

3.1 Público Alvo

Para o público alvo do DoeAção, nós teremos dois tipos de públicos, e dois tipos de persona também. Um será o doador para as campanhas e o outro é quem solicita a ajuda, que irá criar as campanhas.

3.1.1 Responsáveis pela Doação

Pessoas de Classe Média (B, C1, C2), (LOCOMOTIVA, 2021), entre 25 e 40 anos, com renda entre três a oito mil reais, residentes em capitais e metrópoles regionais (FREITAS, 2019), com formação superior, profissionais com salário fixo ou que tenham seu próprio empreendimento, com planejamento financeiro e preocupadas com causas sociais.

3.1.2 Responsáveis pela Criação da Campanha

Pessoas de Classe Média e Baixa (C2, C3, D1, D2, E), com renda até três mil reais, residentes em capitais e metrópoles regionais, sem formação ou formação incompleta, que trabalham em empregos formais ou informais, que possuem acesso à internet e não tem educação financeira.

3.1.3 Personas

Persona é um personagem fictício com as características reais do usuário da aplicação. São baseados conforme a definição do Público Alvo e serve para definir ações de marketing, usabilidade da aplicação, otimizar seu produto, etc.

3.1.3.1 Doador

Mariana, 30 anos, mãe de uma menina de dois anos e adotou recentemente a Dora, uma cachorrinha abandonada. Mora na zona sul de São Paulo. Formada em Design

de Interiores, é empreendedora e presta serviço especializado como *Personal Organizer* e usa redes sociais como *Digital Influencer* para postar dicas de organização e cuidados da casa. Sempre acompanha eventos voltados a arquitetura e design, valoriza as amizades que faz no seguimento e busca sempre manter os contatos próximos. Frequenta academia, se mantém em forma e nas horas vagas passa o tempo com sua filha.

3.1.3.2 Campanha

Roberto, 41 anos, casado e pai de duas meninas, uma com 2 anos e a outra de 7 anos, mora em Osasco, São Paulo, trabalha com serviços gerais, fez curso técnico quando garoto e nas horas de folga trabalha com entregas do Mercado Livre para complementar a renda de casa. Aos finais de semana, joga bola de manhã com os amigos, em casa passa o tempo com as filhas e quando pode, gosta de fazer um churrasco para a família almoçar.

3.1.4 Jornada dos Usuários

Roberto é um homem trabalhador e que adora suas filhas. Sua esposa trabalha, mas ganha pouco também, por isso, trabalha dobrado para não deixar faltar nada em casa.

Recentemente sua filha mais nova passou mal e teve que ir ao médico. Ela foi diagnosticada com uma doença de difícil tratamento, algo que não é coberto pelo SUS (Sistema Único de Saúde).

Sem plano de saúde, sem reserva econômica, ganhando pouco e tendo que lidar com processo jurídico para que o Estado cubra os custos do tratamento no exterior, se vê em um momento em que toda ajuda do mundo será bem-vinda nesse difícil momento.

Como trabalha com serviços de entrega por aplicativo, não foi difícil descobrir um aplicativo de ajuda coletiva e doações para casos em que se precisa de muito dinheiro para um problema específico.

Ele faz o cadastro no site, e já cria uma campanha, coloca um título impactante, o valor para o tratamento, coloca uma imagem da família reunida no hospital, faz um vídeo explicando a situação, coloca toda informação médica passada na descrição e como será o tratamento, finaliza e logo tem os links da campanha e do pix para transferência.

Como é bastante conhecido no bairro, por ter muitos amigos e ser um homem honesto e trabalhador, logo o *link* da campanha se espalha pelos grupos de WhatsApp, Instagram, Facebook e até o vídeo que fez para a campanha foi parar no TikTok.

A esperança do Roberto é arrecadar fundos para o tratamento, pois sabe que tanto o SUS (Sistema Único de Saúde), quanto os processos Jurídicos são lentos e isso pode custar a vida de sua filha.

Mariana é uma mulher muito para cima, decidida, empreendedora, fala bem e lida

com pessoas o dia todo, seja pessoalmente no trabalho ou nas redes sociais, as quais são de onde tira boa parte da renda, com publicações patrocinadas e com a divulgação do seu portfólio de trabalho.

Como é muito ligada em redes sociais, acaba que sempre vê oportunidades de ajudar outras pessoas ou animais abandonados. Enquanto olhava o Reels do Instagram, se deparou com um homem pedindo ajuda para um tratamento de sua filha de dois anos. Ela se impactou imediatamente, ela se imaginou na mesma situação, hora, a filha do homem tem a mesma idade que sua própria filha.

Não foi difícil encontrar um link que a direcionasse para o site, é uma campanha de doações coletiva em prol de uma causa de extrema necessidade. O *link* a levou diretamente para a campanha, onde tinha o vídeo e a descrição do tratamento.

Ela decidiu então colaborar com o Roberto, clicou em Doar, foi redirecionada a tela de login, como não tinha, fez sua inscrição e prosseguiu com a doação, colocou os dados bancários, o valor de contribuição e pronto, em poucos minutos ela fez parte da vida do Roberto e sua filha.

Em poucos minutos depois, ela recebe uma notificação de mensagem do DoeAção, era do Roberto agradecendo pela contribuição. Ela ficou surpresa e desejou melhoras pela menina e se comprometeu a divulgar em suas redes sociais o caso todo e usar sua influência para atingir o maior número de pessoas possíveis.

3.2 Dificuldade do Projeto

Uma das maiores dificuldades a serem enfrentadas pelo projeto será o Marketing. Com as ferramentas atuais de bloqueios de anúncios tanto para o *mobile* como para *desktop*, ficará cada vez mais difícil alcançar o público alvo do nicho desejado.

Também será de extrema importância analisar com a equipe financeira os métodos de arrecadação da aplicação. Esse retorno poderá ser gerado através de um percentual da transação realizada pelos usuários e/ou por interações com anúncios na plataforma. Porém, de qualquer maneira, o projeto precisará chegar ao conhecimento do público alvo, e para isso deverá ser feito por publicações e anúncios em diversas plataformas, o que nos leva a primeira dificuldade apresentada.

O suporte para os usuários da plataforma também é essencial para uma aplicação fluída, pois todo aplicativo, além de ter uma página dedicada para dúvidas e respostas, também possui um atendimento automatizado que consegue ajudar os seus usuários de forma ágil. Não somente isso, mas também ter pessoas que integram o time de suporte para auxiliar o usuário em problemas técnicos, ainda mais se referindo a uma aplicação de movimentações financeiras, é de extrema importância possuir pessoas aptas para o atendimento ao cliente.

Em suma, a equipe deverá alinhar com o time financeiro e de marketing as melhores estratégias para driblar a dificuldade apresentada, pois isso poderá gerar um atraso ou até mesmo o não atingimento do público desejado, como também a não arrecadação de verbas para a continuação do projeto e suas melhorias.

3.3 Missão do Projeto

A missão do DoeAção é promover e melhorar o relacionamento entre doadores e seus donatários e tornar o processo mais eficiente em prol de suas causas, simplificando a captação de doações online e fazendo com que o sistema de doação seja seguro e passe confiança aos usuários.

A DoeAção existe para servir como uma ferramenta facilitadora para realização de doações, ligando quem necessita a quem pode oferecer ajuda, fornecendo mecanismos que tornam o processo de doações para pessoas ou instituições mais fáceis, simples e confiável.

3.4 Benefícios do Projeto

Ajudar o próximo é algo que reflete a empatia que temos pela necessidade do outro, e a ação de ajudar alguém geram diversos benefícios para nossa sociedade, para o meio em que vivemos e principalmente para quem depende dessa ajuda para sanar um problema ou atravessar um momento difícil na vida.

Como benefício, o DoeAção visa disponibilizar um meio simples de arrecadação de fundos para causas sociais e facilitar a propagação delas em redes sociais, contribuindo para difundir a ajuda coletiva em nossa sociedade.

Outro benefício é a normalização das doações, devido à facilidade de contribuição e a possibilidade de assinar contribuições recorrentes e automáticas, minimizando o trabalho de transferências e pagamentos, mas sem perder a interação entre os usuários e o sentimento de ajuda ao próximo.

O DoeAção visa sempre atingir o maior público possível, desse modo, a divulgação em redes sociais é fundamental para o projeto, sendo assim, o usuário responsável pela causa e o usuário que contribui, podem usar links para publicar em suas redes sociais e replicar a causa para o maior número de pessoas possíveis.

3.5 Concorrências

Neste tópico apresentaremos nosso diferencial em relação aos nossos concorrentes, assim para ganharmos destaque estamos focados em sermos únicos, com uma plataforma sustentável, adaptável a múltiplas situações e nos tornar referência em nosso segmento.

Nosso grande destaque é uma plataforma robusta, ágil, adaptativa, pensada para o usuário, cliente criador da campanha e para seu doador. Aplicamos estudos de UX (*User Experience*) e UI (*User Interface*) a fim de otimizar todo esse processo. Em relação a tecnologias, arquitetura e engenharia da aplicação vale ressaltar que já nascemos SaaS (*Software as a Service*), com isso nossa aplicação estará hospedada em nuvem pública, arquitetada para *microservices* e *micro frontends*, assim todo o processo de engenharia e desenvolvimento do software distribuído passa por processos de constantes melhorias, novas implementações e possíveis ajustes sem prejudicar nosso cliente/usuário.

Outros diferenciais serão as taxas cobradas, tempo de repasse dos valores obtidos em campanhas e formas de recebimentos.

Em relação às taxas cobradas aplicamos inicialmente $X\%$, durante os X primeiros anos, a fim de ganhar adesão ao mercado, porém mesmo assim não serão prejudicados nossos retornos financeiros, pois nosso modelo de negócio é otimizado em relação ao desenvolvimento e implementação, que passa por processos de testes e homologações assim dificilmente gerará algum retrabalho o que prejudicaria nossos custos com contratações adicionais e possíveis perdas de prazos.

O tempo de repasse dos valores obtidos será de $D+1$ após o recebimento da doação para o criador da campanha, assim o criador da campanha poderá receber com maior agilidade e aplicar a sua necessidade, podendo ser sacado via PIX ou transferência na rede bancária.

Como formas de recebimentos das doações utilizaremos *gateways* de pagamentos de cartões de crédito e débito de todas as bandeiras aplicadas em nosso país como Amex, Visa, Mastercard, Elo, etc, as quais serão homologadas junto a nossa aplicação. Serão aplicados recebimentos por transferências bancárias (DOC/TED/TEF) e a transferência instantânea PIX.

Em nosso segmento esses serão nossos principais diferenciais até o momento, pois estamos passando por constantes evoluções e melhorias, o que será nosso grande destaque no mercado, pois somos uma empresa VIVA, focada em melhorar a vida de nossos clientes e seus usuários.

3.6 Síntese da Visão do Produto

O Grupo 31 decidiu como projeto, desenvolver aplicação web funcional de doações coletivas e arrecadação de causas sociais. Essa aplicação será usada por Mariana e Roberto, pessoas comuns de classes sociais diferentes e que utilizam o DoeAção com intuítos e motivações diferentes e que resultam em um bem maior, seja coletivo ou pessoal, a junção desses dois personas é uma referência direta ao que o produto se propõe, de ajudar e ser ajudado, e que a aplicação espera oferecer ao público alvo, opções entre os concorrentes que seja mais vantajosa, do ponto de vista financeiro, oferecendo menos taxas de uso da plataforma e melhores meios de propagação das campanhas criadas para concorrer diretamente com Vakinha.com.br e Donativa.org.br.

Com o aumento do número de pessoas que hoje contribuem pela internet e a facilidade que temos hoje de movimentações financeiras como PIX, um mercado se abre para termos um produto que possa se destacar entre os concorrentes de mesmo seguimento e assim, oferecer mais opções para as pessoas que querem ajudar e que precisam de ajuda.

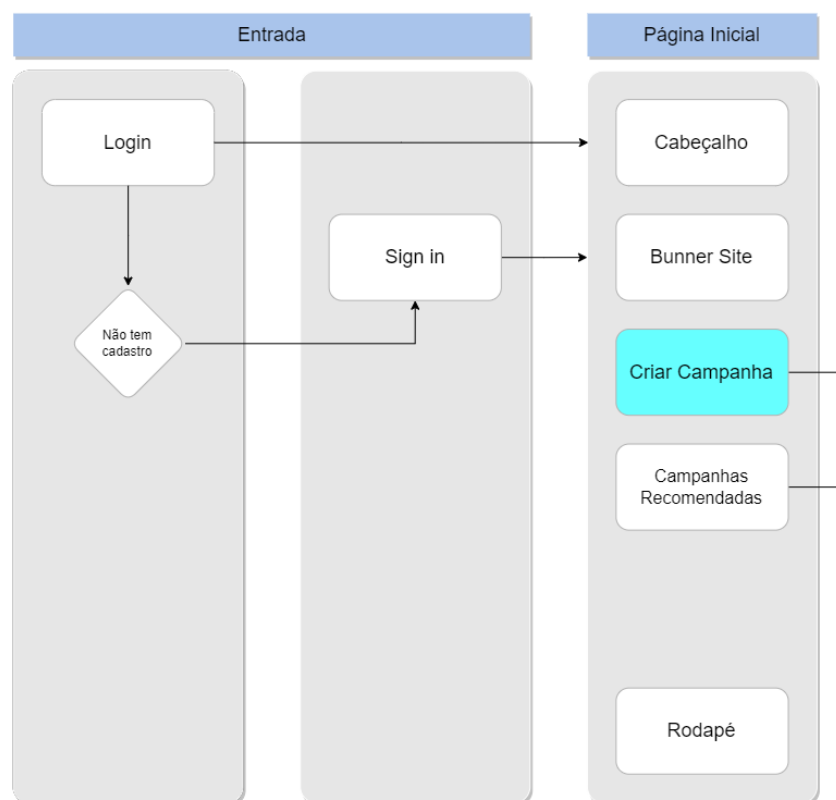
Inicialmente o usuário deve realizar o login, sendo então direcionado para a página inicial. No caso de ainda não ser cadastrado, será direcionado para a página de cadastro. Figura 2.

Na página inicial, visualizamos a logomarca do aplicativo no canto superior esquerdo da área do cabeçalho e ao lado o menu, contendo os componentes de navegação para o usuário acessar as diversas funcionalidades do aplicativo, com as opções Criar Campanha, Minhas Campanhas, Configurações, Doe, Sobre Nós, Políticas e Taxas, FAQ, Trabalhe Conosco e Sair.

Também na página inicial encontramos as Campanhas Recomendadas, onde serão disponibilizadas de forma aleatória, conforme engajamento ou campanhas que precisam de pouco valor para serem finalizadas para que o usuário possa ter um primeiro contato.

Ao final da página inicial está o rodapé, com informações de contato dos desenvolvedores, dúvidas frequentes, link para redes sociais e política de privacidade.

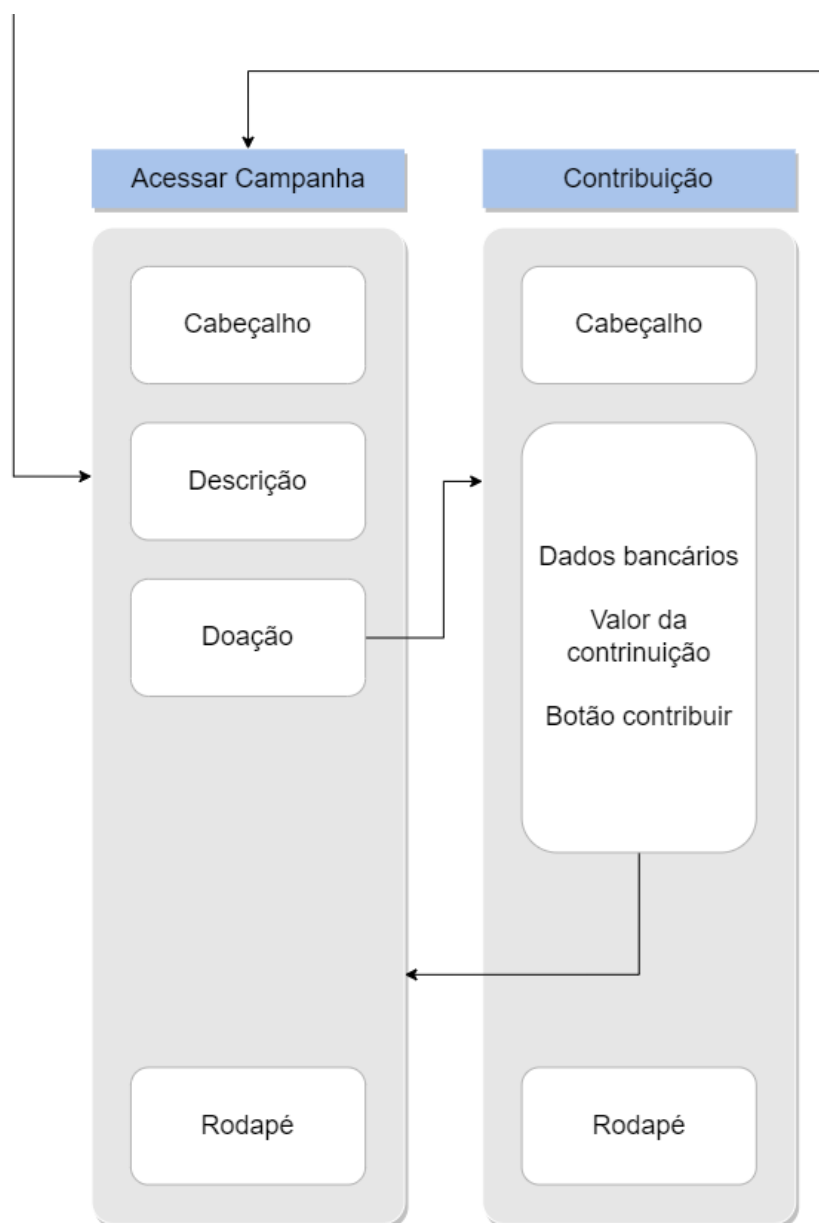
Figura 2 – Login, cadastro e página inicial.



Fonte: Desenvolvimento próprio.

Ao acessar Campanhas Recomendadas, o usuário terá a descrição da campanha selecionada e a opção de realizar a doação. Se optar por doar, deverá preencher seus dados bancários e o valor da doação. Figura 3.

Figura 3 – Acessar campanhas e fazer contribuição.

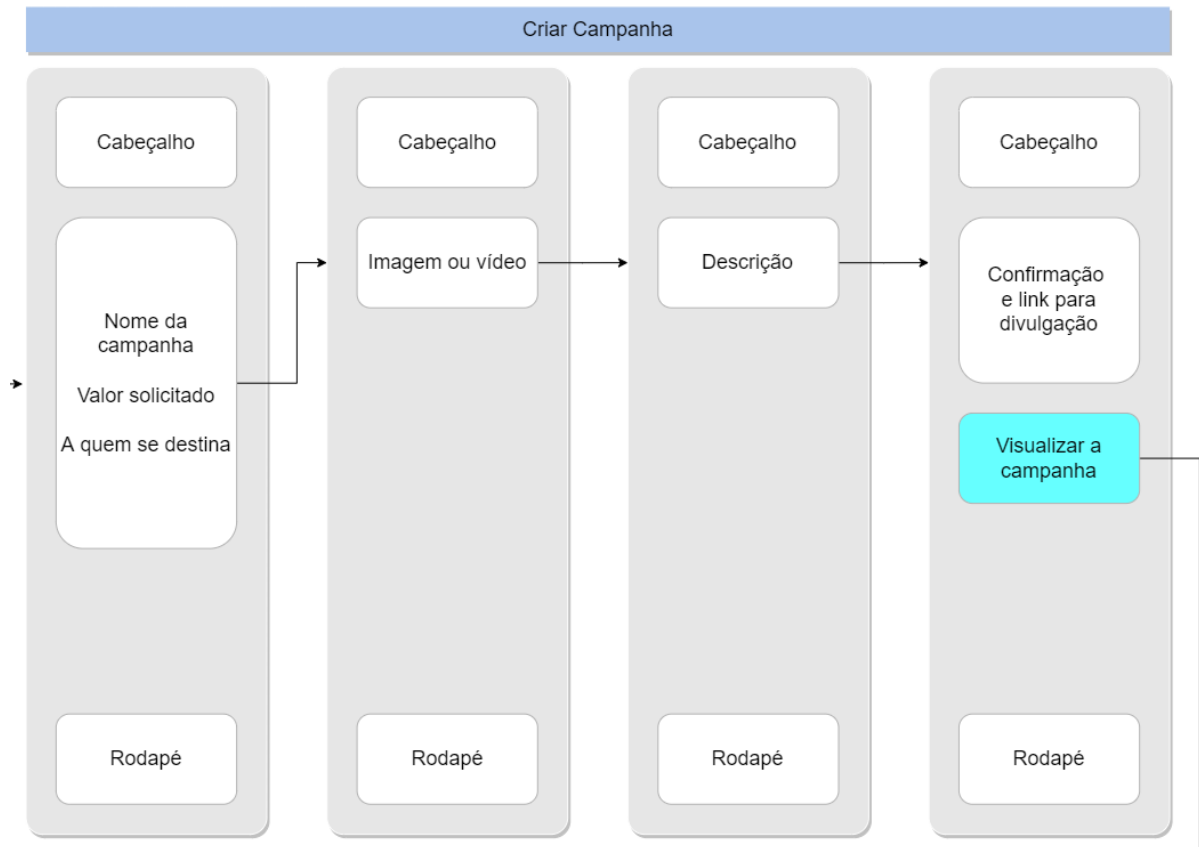


Fonte: Desenvolvimento próprio.

Selecionando a opção Criar Campanha, Figura 4, haverá um passo a passo para a criação da campanha, com inclusão do nome da campanha, valor almejado, a quem se destina, imagem e descrição a serem preenchidas e após a confirmação, será gerado um link para divulgação.

Ao clicar no logo do aplicativo, o usuário será direcionado para a página inicial.

Figura 4 – Criando campanhas na aplicação.



Fonte: Desenvolvimento próprio.

Buscamos, desta forma, criar um ambiente intuitivo e funcional, com identidade visual para que o usuário se sinta familiarizado com o produto, gerando conforto e credibilidade ao aplicativo e estimulando sua utilização e a realização de doações.

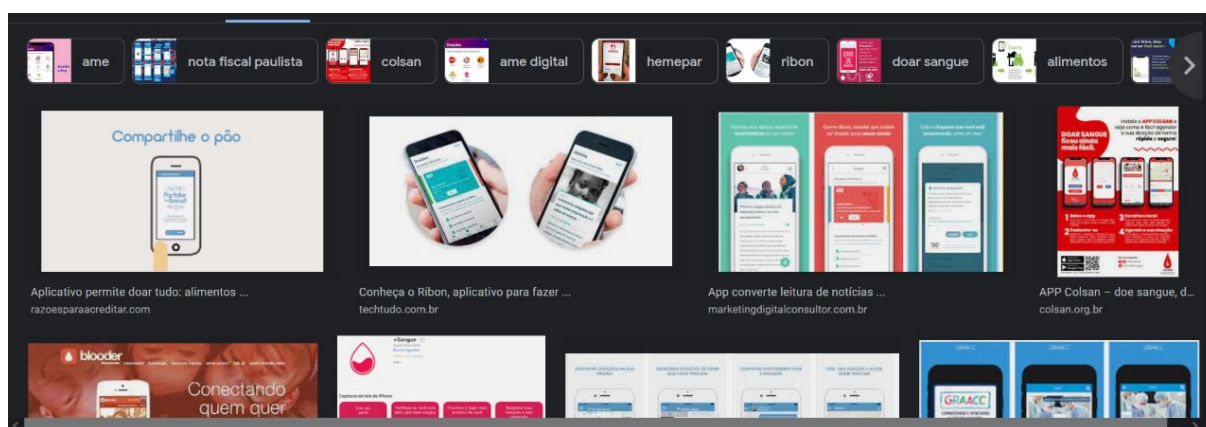
4.2 Construção da Aplicação

Com um layout minimalista e voltado para uma interface mais amigável, o protótipo do DoeAção visa mostrar as funcionalidades básicas do aplicativo, como também deixa a curiosidade do usuário para o que ainda pode ser elaborado no produto.

A construção do protótipo se deu através do aplicativo Figma (2022), sendo um aplicativo para design de interfaces, tanto mobile como desktop.

Antes de se iniciar a construção de um protótipo, é necessário levantar alguns outros aplicativos que já atuam no mercado, Figura 5, e possuem uma finalidade parecida com o DoeAção. Logo, com esse processo feito, é importante começar a analisar o visual de cada um deles, pois dessa forma se tem uma base para começar a construção.

Figura 5 – Referências buscadas no Google.



Fonte: Desenvolvimento próprio.

Não somente utilizando outros aplicativos do mesmo nicho de guia, um entendimento sobre como montar um design fluido e minimalista é importante para a construção da aplicação, pois dessa forma é possível alcançar uma consistência visual para que os futuros usuários consigam vislumbrar o potencial do aplicativo com uma navegação fluida.

Com esses fundamentos em mente e com boas referências, se inicia o processo de criação do protótipo.

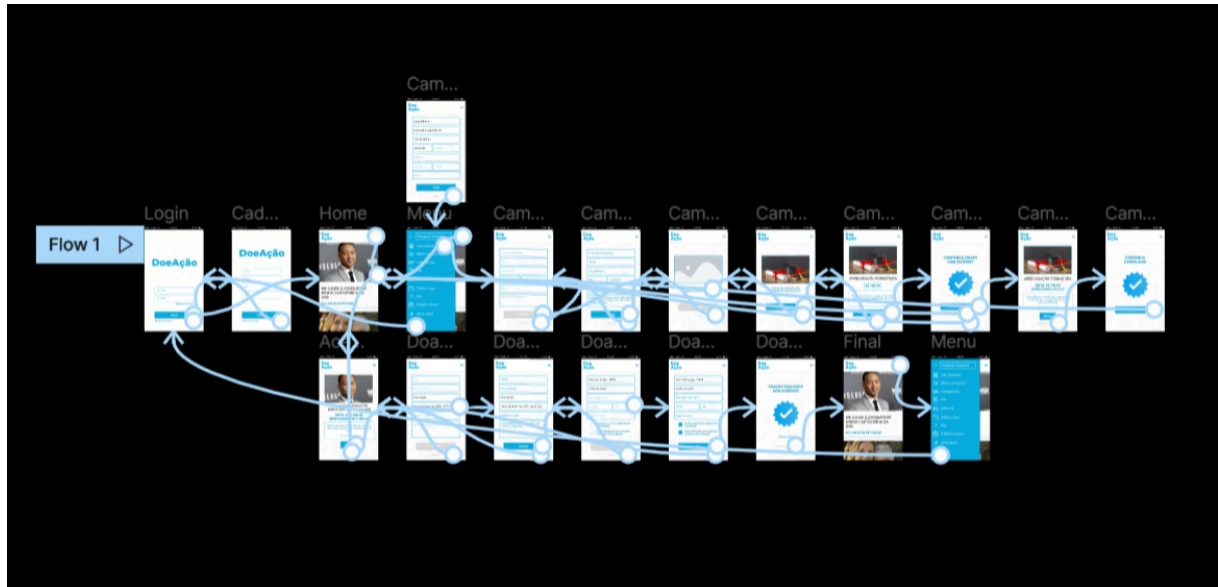
Com o processo de desenvolvimento iniciado dentro do Figma, é importante manter uma comunicação assertiva com o time e uma coerência com o diagrama desenvolvido, pois dessa forma será possível chegar a um resultado mais acurado do que foi planejado.

É importante ressaltar que a ferramenta utilizada não somente mostra a interface visual criada, como também faz uma versão “viva” do protótipo, que seria uma demonstração de como o protótipo funcionaria.

Através de links entre as imagens, o usuário ao clicar em alguma parte da tela, como, por exemplo, um botão de login, ele será levado para uma próxima tela em que poderá fazer alguma outra ação pré-programada pelo time de design responsável.

Dessa forma é possível criar testes fluídos apenas com imagens sem mesmo ter uma aplicação desenvolvida. Figura 6.

Figura 6 – Interface da aplicação com os seus fluxos de protótipo entre as telas.



Fonte: Desenvolvimento próprio.

Com o protótipo criado, validado pelo time e também por alguns possíveis usuários, será possível passar para a etapa de desenvolvimento e de fato iniciar a produção do projeto.

Em suma, é possível analisar que a construção de um protótipo é fundamental para a posterior construção da aplicação, sendo uma parte totalmente dependente da outra.

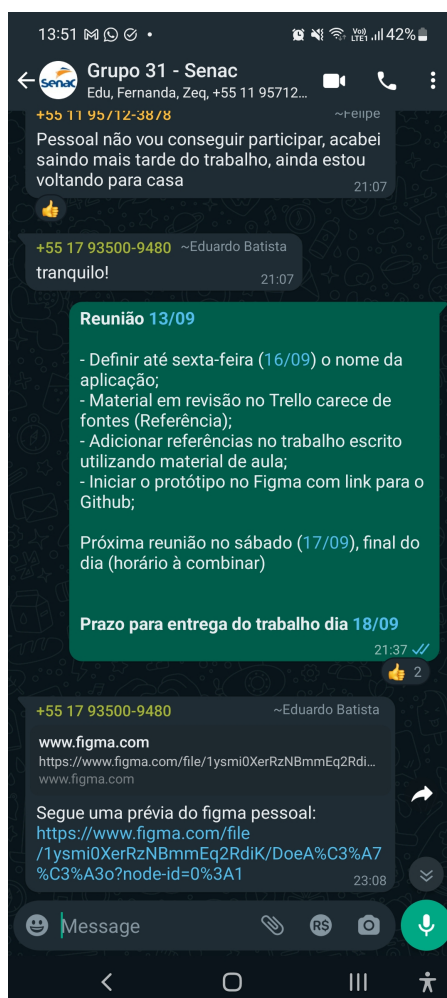
5 PLANEJAMENTO

A organização do projeto foi dividida em Comunicação e Planejamento, Organização e Desenvolvimento, e por fim, Entrega e Finalização.

A comunicação foi feita ao longo do projeto entre os participantes para as tomadas de decisões e o planejamento do trabalho escrito e o desenvolvimento da aplicação. Para isso foram utilizados os seguintes meios:

- Grupo no WhatsApp: Para reunir e atualizar o grupo sobre às atualizações do projeto, enviar lembretes referente ao cronograma, avisos sobre reuniões e atas de reuniões realizadas. Figura 7.
- Google Meet: Utilizado para realizar as reuniões do grupo, onde eram definidas as tomadas de decisões do projeto e planejamento de escopo e desenvolvimento da aplicação.

Figura 7 – Grupo do WhatsApp e definições pendentes de reunião passada.

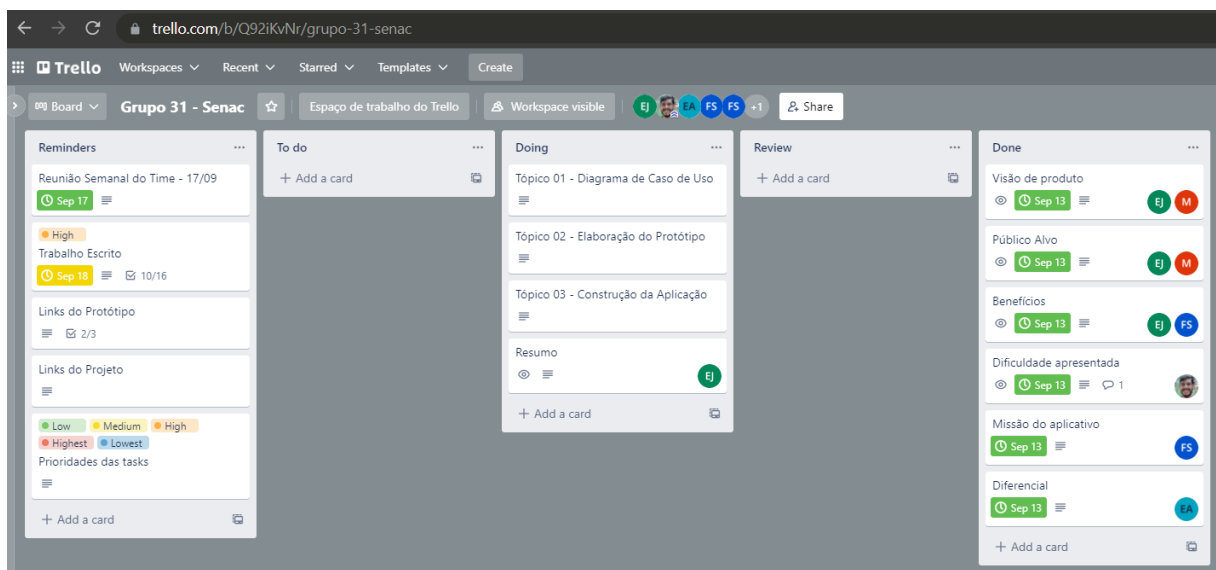


Fonte: Desenvolvimento próprio.

A organização do projeto foi toda baseada em *Kanban* (RADIGAN, 2022), uma metodologia não estrutural de projeto, utilizada para implementar *agile* (ATLASSIAN, 2022) e desenvolvimento de *software*, onde cada etapa do projeto ficou disponível para o desenvolvimento, revisão e conclusão. Para isso foram utilizados as seguintes ferramentas:

- Trello: Todas as tarefas foram segmentadas em *cards*, onde cada integrante assumia a responsabilidade de desenvolver o conteúdo e passando por etapas de desenvolvimento, revisão e conclusão. Ainda eram utilizados *cards* para lembretes de reuniões e *links* do projeto (Google Drive, GitHub, Overleaf, Drawio e Figma). Figura 8.
- Google Drive: Cada tópico foi desenvolvido utilizando o Google Docs e linkado em seu respectivo *card* no Trello.
- GitHub: Repositório criado para o desenvolvimento do *frontend* e do *backend* da aplicação (GRUPO31, 2022).

Figura 8 – Utilização do Trello para segmentar o desenvolvimento do projeto.



Fonte: Desenvolvimento próprio.

Os entregáveis eram todo o conteúdo produzido e finalizado para compor o projeto, sendo a documentação, diagramas, imagens ou a prototipação do projeto.

6 PADRÃO DE ARQUITETURA MVC

O MVC funciona como um padrão de arquitetura de software que proporciona conexão entre as camadas de dados, lógica de negócio e interação com usuário. Através da sua divisão em três componentes, o processo de programação se torna algo mais simples e dinâmico (ZUCHER, 2020).

Alguns desenvolvedores consideram este como um padrão de *design* de interface, porém ele é um padrão de arquitetura de software, que é responsável por contribuir na otimização da velocidade entre as requisições feitas pelo comando dos usuários. Resta uma recorrente dúvida se no processo de desenvolvimento podemos ter apenas essas 3 camadas ou se é possível incluir algum caso haja necessidade.

Concluimos que sim para a possibilidade de inserir uma camada extra. Essa sequência de códigos pode ser alterada conforme a necessidade do projeto. Por padrão existem a camada *Model*, *Controller* e *View* que deram origem a sigla dessa arquitetura de software mais utilizado entre os desenvolvedores.

Nossa motivação para utilizar esse padrão de arquitetura são:

- **Segurança:** O *controller* se estabelece como um filtro, cuja suas funções são de impossibilitar que algum dado incorreto chegue até a camada modelo.
- **Organização:** Pois temos a facilidade no entendimento do que foi construído, assim como os possíveis erros se tornam mais fáceis de serem encontrados e corrigidos.
- **Eficiência:** A implementação é mais leve, permitindo a possibilidade de mais desenvolvedores trabalharem no projeto de forma independente.
- **Tempo:** A colaboração entre os profissionais de desenvolvimento se torna mais clara, assim o projeto pode ser concluído em um menor tempo e o projeto se torna escalável.
- **Transformação:** Mudanças necessárias podem ser mais fluidas, como o foco está em trabalhar nas regras de negócio.

7 PADRÃO DE DESENVOLVIMENTO

O padrão REST (*Representational State Transfer*) é um estilo de arquitetura para o desenvolvimento de *Web Services* e APIs (*Application Programming Interface*). Seu criador foi Roy Fielding, um dos principais autores da especificação HTTP (*Hypertext Transfer Protocol*) e cofundador do projeto *Apache HTTP Server*.

É utilizado como um padrão para fazer a troca das informações entre as aplicações. É uma alternativa ao modelo tradicional SOAP (*Simple Object Access Protocol*), pela leveza dos pacotes de dados transmitidos na rede, pela simplicidade, sendo desnecessária a criação de camadas intermediárias para encapsular dados e pela facilidade de desenvolver. Utiliza a sintaxe universal para identificar os recursos (URIs) (LIMA, 2020).

Tem como objetivo utilizar de forma mais eficiente o protocolo HTTP, principalmente em relação a sua semântica na utilização dos métodos HTTP (*GET*, *POST*, *PUT* e *DELETE*). A ideia principal é poder fazer a manipulação básica de dados de uma aplicação através de simples chamadas HTTP. Quando uma solicitação REST é feita, o servidor envia ao cliente uma representação dos estados que foram requeridos, ou seja, dá um retorno das informações no momento em que foram solicitadas. Também permite que se realize ações, como inserir, alterar e excluir registros.

A implementação do *Web Service* visa permitir a interoperabilidade entre aplicações, podendo ser utilizado em diferentes plataformas, sistemas operacionais e hardwares.

É uma tecnologia cliente-servidor, que serve para enviar e receber dados através do uso da internet pelo protocolo HTTP, provendo serviços entre softwares executados em diferentes plataformas. Utilizam padrões de comunicação para solicitar e retornar os tipos de conteúdo utilizados na requisição, onde irá especificar o tipo de entrada e retorno da aplicação, como xml, html, json, entre outros (DEVMEDIA, 2016).

A grande vantagem da utilização do *Web Service* é o fato de não exigir uma determinada tecnologia para os envolvidos no processo e a utilização do padrão REST para o desenvolvimento do DoeAção foi a mais adequada por suas características de integração de sistemas pela internet, propiciando soluções de qualidade.

8 TECNOLOGIAS UTILIZADAS

Em relação às tecnologias utilizadas podemos citar as principais como as de base, que são as linguagens de programação Java no *backend* e JavaScript, HTML e CSS no *frontend*.

Optamos no *backend* por utilizar o *framework* Spring que utiliza neste cenário para o gerenciamento de bibliotecas e dependências o Apache Maven, implementamos algumas dependências para otimizar o processo de desenvolvimento e diminuir o uso de códigos *boilerplate* como o Lombok, para acesso ao banco de dados de forma simplificada utilizamos o Spring Data JPA já que estamos utilizando uma base de dados relacional, e como esta é uma aplicação em fase de demonstração e testes utilizamos um banco de dados em memória H2 a fim de minimizar custos já que implantamos a aplicação em *cloud* pública Heroku.

No frontend optamos pela não utilização de *frameworks* como Angular, Vue, React, etc, pois é uma aplicação bastante simples e com um único fluxo implementado, porém utilizamos para melhorar o estilo e se adequar ao protótipo desenvolvido no Figma o Bootstrap e foi utilizado a biblioteca jQuery para otimizar o uso do JavaScript assim diminuindo a verbosidade do código escrito.

8.1 Backend

8.1.1 Java

Java é uma linguagem de programação compilada e orientada a objetos, com aplicações portáteis entre multiplataformas, que oferece um ambiente de desenvolvimento robusto e confiável (GOSLING E MCGILTON, 1995), por se tratar de uma das linguagens com bastante tempo de mercado e produção, tendo seu início em 1995 (BINSTOCK, 2015).

Para o desenvolvimento do backend foi utilizado a linguagem Java na versão 11, que oferece diversos modelos de projetos e ambientes de desenvolvimento, inclusive, uma gama de dependências acrescentadas ao projeto que facilitam a produção e a integração dos serviços.

8.1.2 Apache Maven

A linguagem Java permite a utilização de bibliotecas externas através da importação de pacotes, que devem ser implementadas por um gerenciador de build e dependências, chamado Apache Maven. Ele é baseado no conceito de *project object model* (POM) e ainda pode gerenciar, além da construção e utilização de dependências, relatórios e documentação de um projeto (FONDATION, 2022).

No projeto DoeAção, o Maven gerencia as dependências referente ao Spring, banco de dados e outras ferramentas utilizadas no projeto, como o Lombok, OpenAPI e o Thymeleaf, que serão abordados a seguir.

8.1.3 Dependências

Dependências são bibliotecas externas que são adicionadas ao projeto em que cada uma tem uma determinada função específica no projeto. São utilizadas pelo Maven, através do arquivo pom.xml, para definir quais bibliotecas e as características da aplicação (BEZERRA, 2018).

8.1.3.1 Spring

- **Spring Boot Starter Web:** Sua função principal é auxiliar na construção de aplicações web combinando as várias dependências provenientes de um projeto Spring Boot em uma única dependência, retirando a necessidade de configuração de múltiplas dependências no Maven, tornando o processo de definição de componentes da aplicação muito mais simples (GEEKHUNTER; ZUP, 2022, 2021).
- **Spring Boot Starter Test:** Spring Boot fornece vários utilitários e anotações para ajudar nos testes da aplicação e a dependência Starter Test traz uma série de bibliotecas usadas nesses testes e no desenvolvimento no projeto, dentre elas a JUnit, Spring Test, Mockito, JsonPath, etc (WEBB ET AL., 2017).
- **Spring Boot Devtools:** DevTools é um módulo que adiciona algumas ferramentas para ajudar no desenvolvimento. Um benefício é que DevTools configura algumas propriedades com valores que são convenientes na questão de tempo de desenvolvimento, por exemplo, algumas bibliotecas suportadas pelo Spring Boot usam cache para melhorar sua performance, mas tem que recarregá-las sempre que forem requisitadas, o pode atrapalhar o tempo do desenvolvimento, então o DevTools configura algumas propriedades com valores que são convenientes para otimizar esse processo. Ele também realiza a reinicialização automática, cada arquivo criado nos diretórios que fazem parte do seu classpath serão monitorados e qualquer alteração neles acarretará em uma reinicialização do Spring Boot (AFONSO, 2017).

8.1.3.2 Banco de Dados

- **Spring Boot Data JPA:** Spring Data JPA é um módulo que implementa camadas de acesso a dados e facilita a criação de aplicativos baseados em Spring.

Ele reduz o esforço do desenvolvimento bastando o desenvolvedor escrever suas interfaces de repositório e os métodos de localização, e o Spring implementa tudo isso de maneira automática (TANZU, 2022).

- **H2 Data Base:** É um banco de dados relacional escrito em Java, de código aberto e que funciona em memória, ou seja, seu armazenamento é volátil. Tem o intuito de ser um banco de dados de configuração rápida e fácil, favorecendo a produtividade. Pode ser integrado em aplicativos Java ou executado no modo cliente servidor. Não há limite para o número de bancos de dados abertos simultaneamente ou para o número de conexões abertas (CORPORATION, 2022).

8.1.3.3 Outras Ferramentas Utilizadas

- **Lombok:** A utilização do Lombok nos permite escrever um código com menos verbosidades. É um *framework* para Java focado em produtividade e redução de código *boilerplate*, através de anotações adicionadas ao código. Seu uso permite gerar em tempo de compilação os métodos *getters* e *setters*, métodos construtores, padrão *builder*, entre outros (DEVMEDIA, 2013).
- **OpenAPI:** São interfaces de código aberto que ficam disponíveis para desenvolvedores externos, podendo ser integradas com outros sistemas ou terem seus códigos utilizados. Possibilita a troca de informações em tempo real, pois quando uma API aberta se conecta a uma outra aplicação, essa aplicação retorna dados para ela. Entre os benefícios estão a fácil implementação, são baseadas em nuvem, podendo ser utilizadas onde um usuário tiver acesso à internet e possibilitam um fluxo de trabalho integrado e ajustável para melhorar a experiência do cliente (NEGHERBON, 2022).
- **Spring Boot Starter Thymeleaf:** Thymeleaf é uma biblioteca baseada em Java usada para a criação de *templates* para aplicações web com Spring Boot, capaz de processar HTML (*HyperText Markup Language*), XML (*Extensible Markup Language*), JavaScript, CSS (*Cascading Style Sheets*), texto e RAW (TEAM, 2018).

8.1.4 Cloud Pública Heroku

Heroku é uma plataforma de nuvem que fornece infraestrutura de hardware e servidores base para a implantação de aplicativos na web e seu foco é permitir que desenvolvedores foquem no desenvolvimento da aplicação e implantá-la imediatamente para produzir valores desde o início (HEROKU, 2022).

8.2 Frontend

8.2.1 HTML, CSS e JavaScript

Para estruturar o site foi utilizado HTML5 uma linguagem de marcação de hipertexto sendo o fundamentos de diversas aplicações na *web*, utiliza *tags* de marcação para distinguir os conteúdos visuais e não visuais que seriam os metadados das páginas (DOCS, 2022c).

Na apresentação da pagina foi utilizado CSS junto ao *framework* Bootstrap, CSS é uma linguagem que define os estilos da aplicação escrita em HTML descrevendo como eles devem ser renderizados, já o Bootstrap condensa diversas classes com estilos já pré-definidos em CSS o que otimiza a criação do estilo da página e deixa padronizado (DOCS; TEAM, 2022b, 2022).

A comunicação com o *backend* foi feita com a linguagem Javascript e sua biblioteca JQuery, javascript é uma linguagem interpretada e se baseia em protótipos, múltiplos paradigmas e dinâmica muito utilizado na web para efeitos visuais ou comunicação com o *backend*, atualmente ela não se limita só ao *frontend* com o nodeJS ela também pode ser utilizada para o *backend*, já o JQuery tem como objeto facilitar a elaboração do javascript no *front* facilitando a interação com o DOM da página e também na elaboração de *requests* como *POST* e *GET* (DOCS; FOUNDATION, 2022a, 2022).

As ferramentas estão conectadas através de *links* nas páginas HTML que referenciam os arquivos CSS e JS, para que os estilos do bootstrap sejam visíveis é necessário utilizar as classes prontas que ele oferece enquanto o JQuery é necessário utilizar o caractere \$ em um arquivo JS ou dentro da *tag* “script” para utilizar as funções que ele oferece.

9 DEMONSTRAÇÃO DA SOLUÇÃO PROPOSTA

Para rodar a aplicação, será necessário atender alguns pré-requisitos de instalações de ferramentas que irão trabalhar em conjunto para a utilização do aplicativo.

Deverá ter instalado o Git, para clonar o repositório do projeto, e o Node.js e NPM, para subir a aplicação.

- Node v14.17.0
- Npm v6.14.13

Após isso, clone o repositório do projeto:

```
$ git clone <https://github.com/edukobilinski/pi-senac>
```

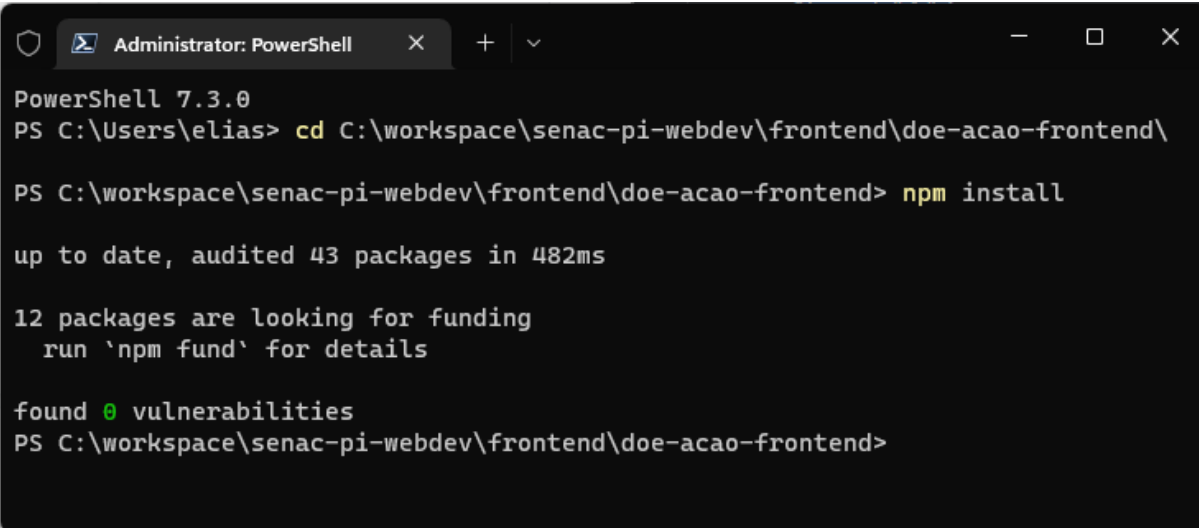
Abra o terminal CMD e acesse o diretório onde o projeto clone foi salvo. Siga até o diretório do projeto indicado:

```
$ cd frontend/does-acao-frontend
```

Instale as dependências NPM com o comando abaixo, Figura 9:

```
$ npm install
```

Figura 9 – Instalação das dependências do NPM na pasta do projeto.



```
Administrator: PowerShell
PowerShell 7.3.0
PS C:\Users\elias> cd C:\workspace\senac-pi-webdev\frontend\does-acao-frontend\

PS C:\workspace\senac-pi-webdev\frontend\does-acao-frontend> npm install

up to date, audited 43 packages in 482ms

12 packages are looking for funding
  run `npm fund` for details

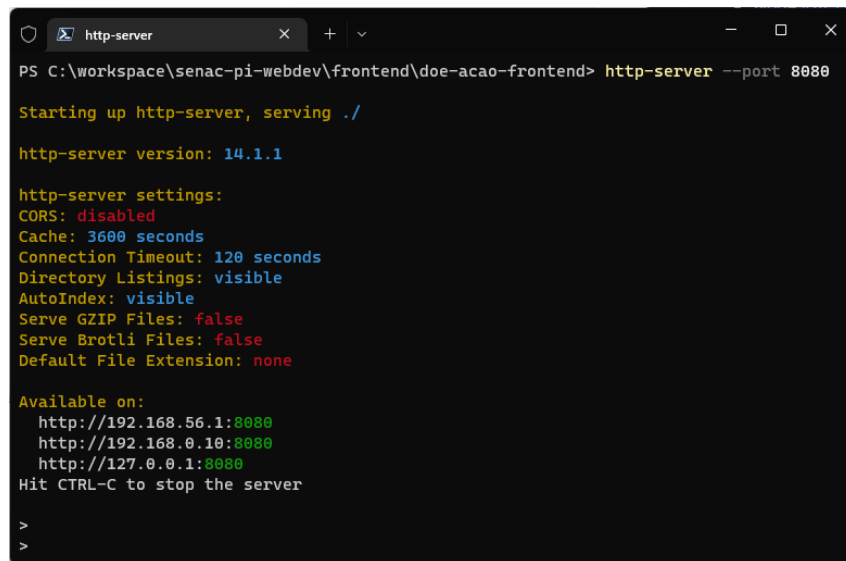
found 0 vulnerabilities
PS C:\workspace\senac-pi-webdev\frontend\does-acao-frontend>
```

Fonte: Desenvolvimento próprio.

Execute a aplicação em modo de desenvolvimento, Figura 10:

```
$ http-server --port 8080
```

Figura 10 – Rodando o http-server na porta 8080.

A terminal window titled 'http-server' showing the command 'http-server --port 8080' being executed. The output indicates the server is starting up, serving the current directory, and lists the version (14.1.1) and various settings like CORS (disabled), Cache (3600 seconds), and connection timeout (120 seconds). It also provides available URLs for access: http://192.168.56.1:8080, http://192.168.0.10:8080, and http://127.0.0.1:8080.

```
PS C:\workspace\senac-pi-webdev\frontend\doe-acao-frontend> http-server --port 8080

Starting up http-server, serving ./

http-server version: 14.1.1

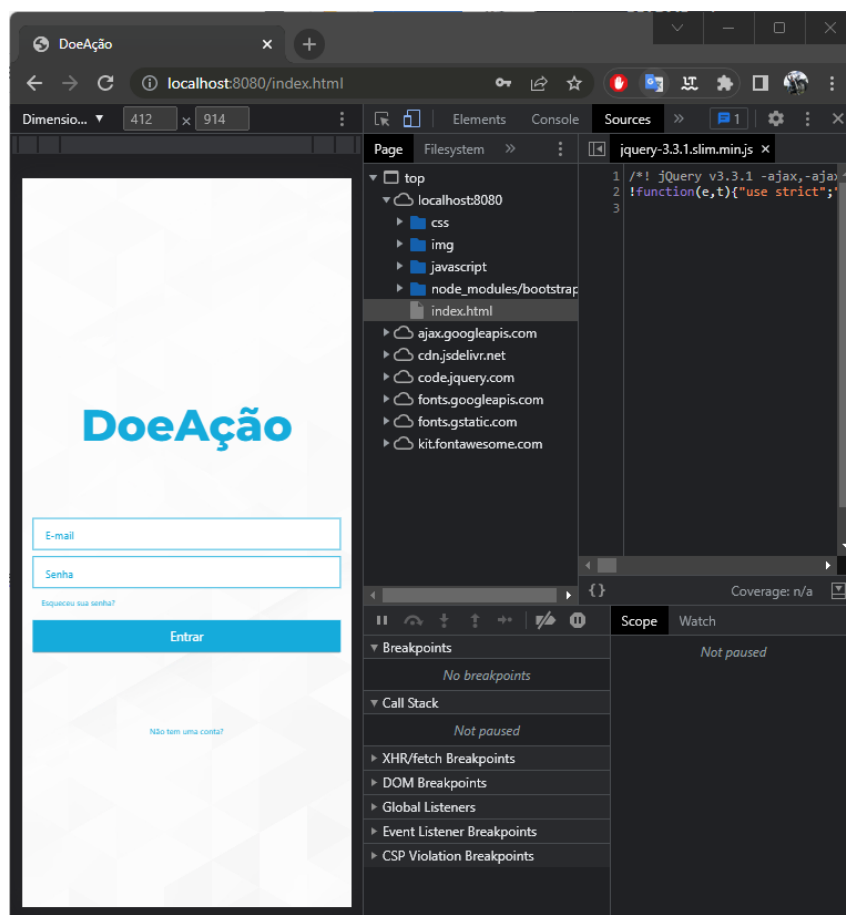
http-server settings:
CORS: disabled
Cache: 3600 seconds
Connection Timeout: 120 seconds
Directory Listings: visible
AutoIndex: visible
Serve GZIP Files: false
Serve Brotli Files: false
Default File Extension: none

Available on:
  http://192.168.56.1:8080
  http://192.168.0.10:8080
  http://127.0.0.1:8080
Hit CTRL-C to stop the server

>
>
```

Fonte: Desenvolvimento próprio.

O servidor iniciará na porta: 8080, e para acessar a aplicação, abra o *browser* e acesse `http://localhost:8080/`. Figura 11:

Figura 11 – Abrindo aplicação no *browser*.

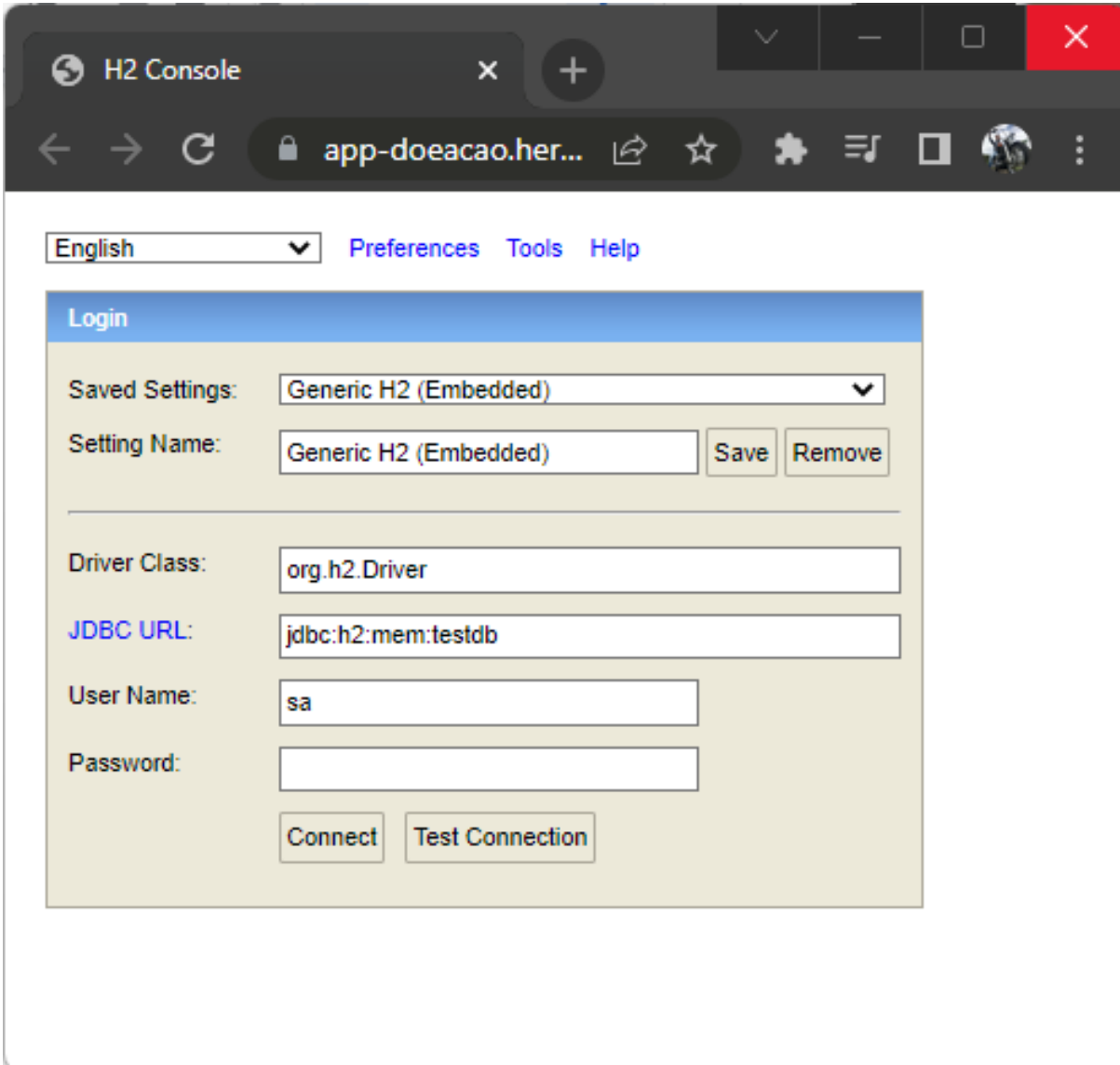
Fonte: Desenvolvimento próprio.

Nesse ponto, a aplicação se conectou ao *backend* no Heroku, onde fica disponível para acesso, e onde está o bando de dados H2 relacional do projeto.

Nesse [link](#) , é possível acessar o banco de dados.

Após abrir o link, no campo JDBC URL, deve ser preenchido com o caminho: `jdbc:h2:mem:testdb`, Figura 12:

Figura 12 – Acesso ao banco de dados do projeto.



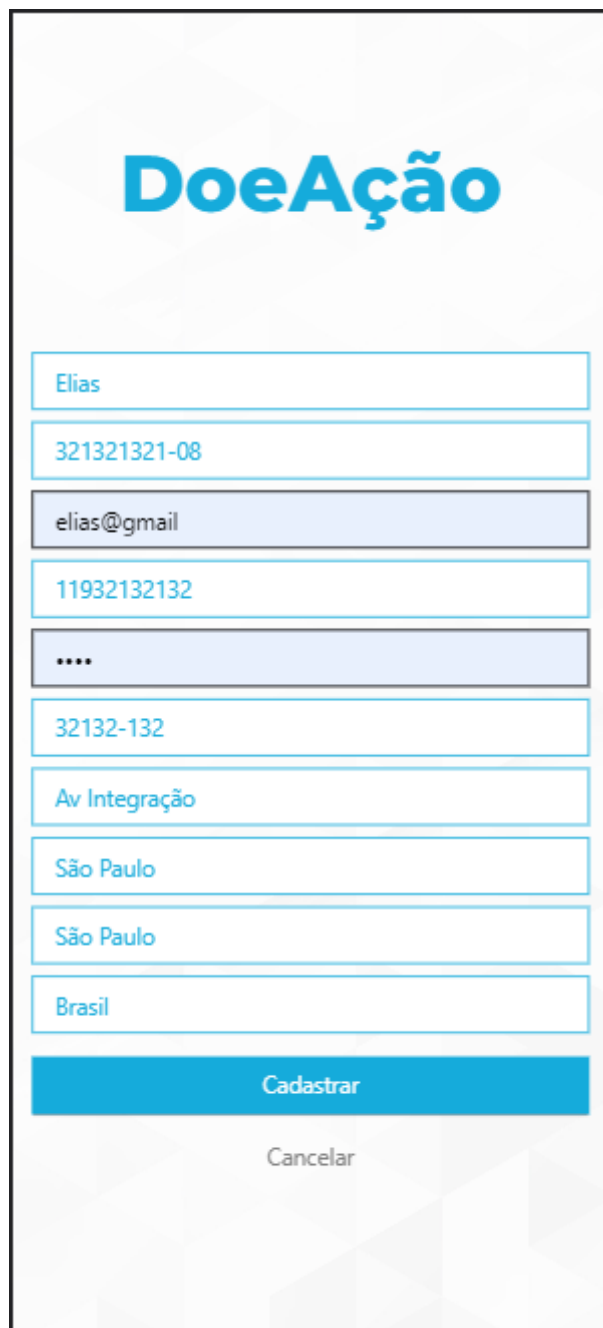
The screenshot shows a web browser window with the title "H2 Console". The address bar displays "app-doeacao.her...". The page has a navigation bar with "English" (a dropdown menu), "Preferences", "Tools", and "Help". The main content area is titled "Login" and contains a form with the following fields and controls:

- Saved Settings:** A dropdown menu showing "Generic H2 (Embedded)".
- Setting Name:** A text input field containing "Generic H2 (Embedded)", followed by "Save" and "Remove" buttons.
- Driver Class:** A text input field containing "org.h2.Driver".
- JDBC URL:** A text input field containing "jdbc:h2:mem:testdb".
- User Name:** A text input field containing "sa".
- Password:** An empty text input field.
- At the bottom of the form are "Connect" and "Test Connection" buttons.

Fonte: Desenvolvimento próprio.

Voltando ao *frontend*, criaremos um usuário caso ainda não tenha um em “Não tem uma conta?”, Figura 13:

Figura 13 – Tela de cadastro.



A tela de cadastro do DoeAção apresenta o logo da organização no topo. Abaixo, há uma série de campos de entrada para coletar dados pessoais e de contato. Os campos são: nome (Elias), CPF (321321321-08), e-mail (elias@gmail.com), número de telefone (11932132132), senha (representada por pontos), confirmação de senha (32132-132), endereço (Av Integração), cidade (São Paulo), estado (São Paulo) e país (Brasil). No final da tela, há dois botões: 'Cadastrar' em azul e 'Cancelar' em cinza.

Elias
321321321-08
elias@gmail
11932132132
....
32132-132
Av Integração
São Paulo
São Paulo
Brasil
Cadastrar
Cancelar

Fonte: Desenvolvimento próprio.

No menu, selecione “Criar Campanha”, Figura 14:

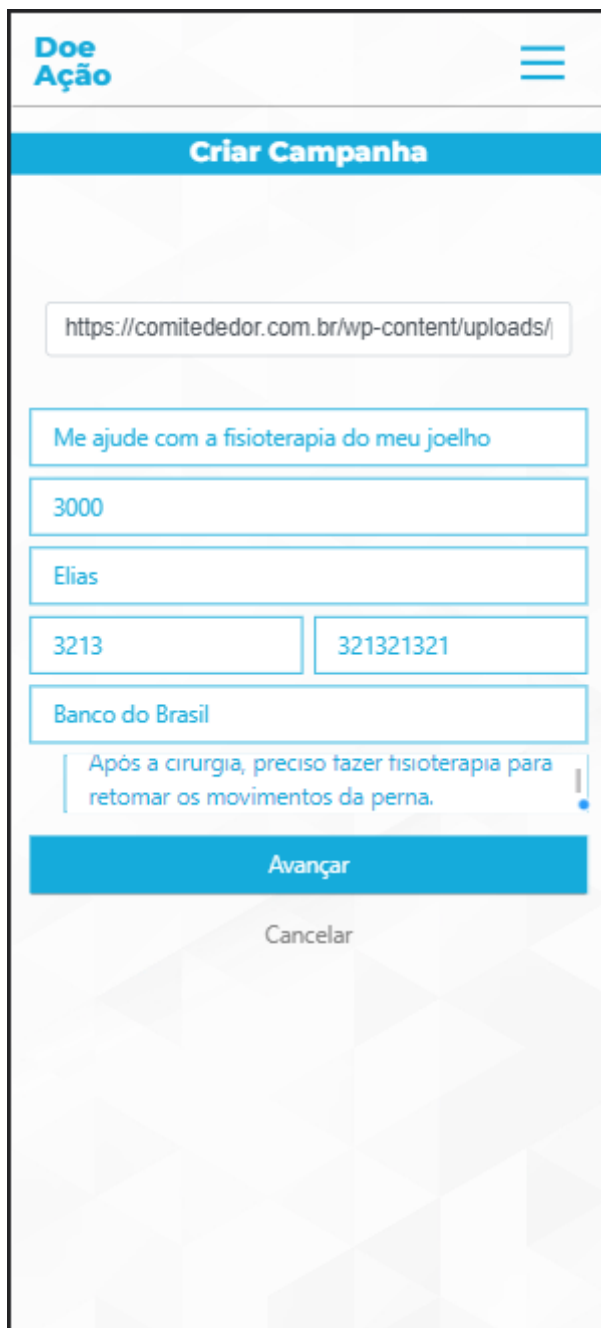
Figura 14 – Menu da aplicação.



Fonte: Desenvolvimento próprio.

Insira uma url da imagem da campanha e preencha os dados necessários para a divulgação e arrecadação da campanha, Figura 15:

Figura 15 – Preenchendo dados da campanha.



A captura de tela mostra a interface de um aplicativo móvel com o cabeçalho "Doe Ação" e um ícone de menu hambúrguer. O título da tela é "Criar Campanha". O formulário contém os seguintes campos:

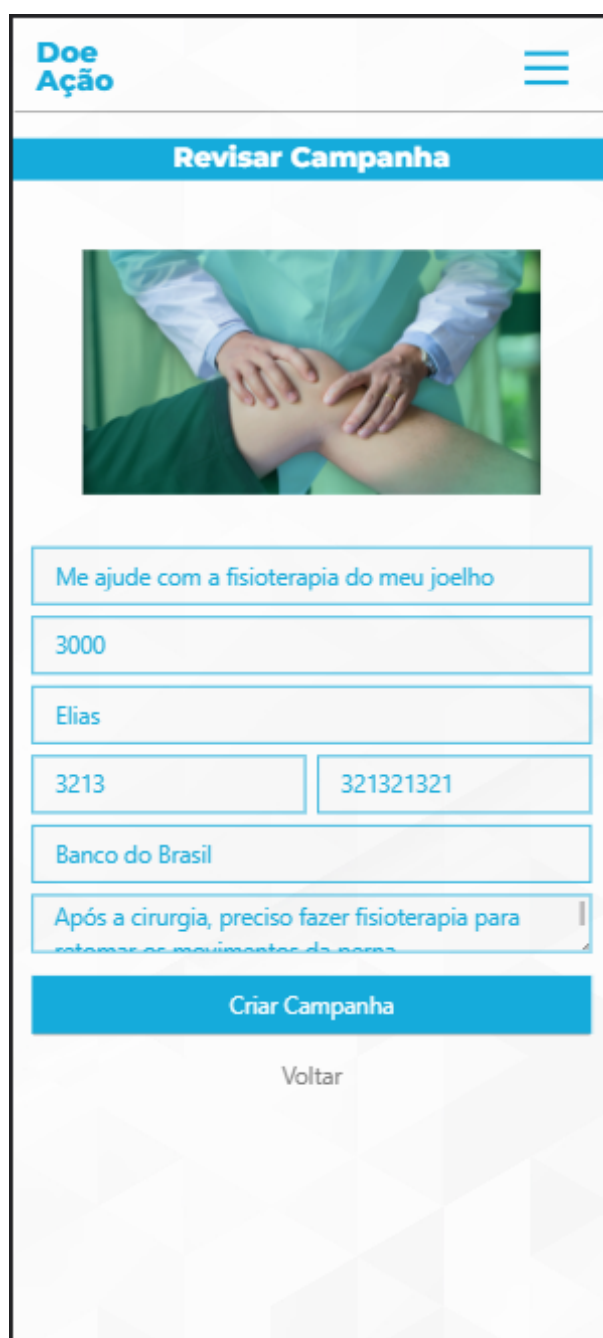
- URL da imagem da campanha: `https://comitededor.com.br/wp-content/uploads/`
- Título da campanha: "Me ajude com a fisioterapia do meu joelho"
- Valor da doação: "3000"
- Nome: "Elias"
- CPF: "3213" e CNPJ: "321321321"
- Banco: "Banco do Brasil"
- Descrição da campanha: "Após a cirurgia, preciso fazer fisioterapia para retomar os movimentos da perna."

Na base do formulário, há dois botões: "Avançar" (em azul) e "Cancelar" (em cinza).

Fonte: Desenvolvimento próprio.

Confira os dados e crie a campanha. Figura 16


Figura 16 – Revisando dados inseridos.



The image shows a mobile application interface for reviewing campaign data. At the top, there is a header with the logo "Doe Ação" and a hamburger menu icon. Below the header is a blue bar with the text "Revisar Campanha". The main content area features a large image of a person's knee being massaged by a healthcare professional. Below the image, there are several input fields for campaign details: a text field with the placeholder "Me ajude com a fisioterapia do meu joelho", a numeric field with the value "3000", a text field with the name "Elias", two numeric fields with the values "3213" and "321321321", a text field with the bank name "Banco do Brasil", and a text field with the description "Após a cirurgia, preciso fazer fisioterapia para retomar os movimentos da perna". At the bottom of the form, there are two buttons: "Criar Campanha" (Create Campaign) and "Voltar" (Go Back).

Doe
Ação

Revisar Campanha



Me ajude com a fisioterapia do meu joelho

3000

Elias

3213 321321321

Banco do Brasil

Após a cirurgia, preciso fazer fisioterapia para retomar os movimentos da perna

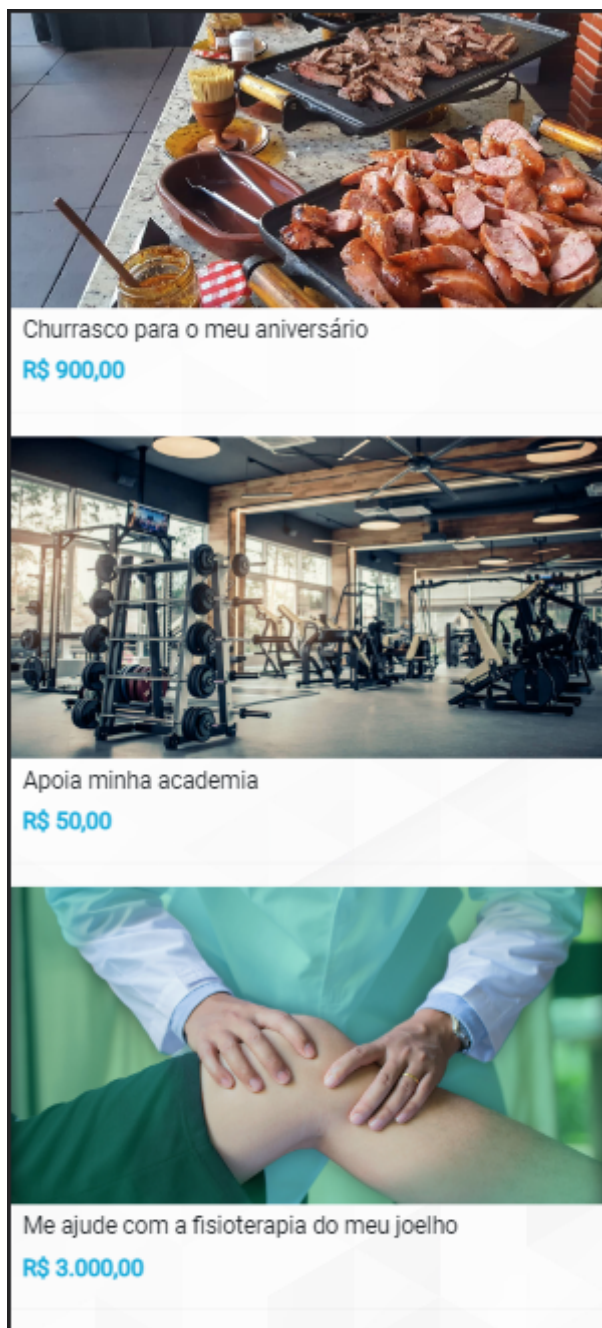
Criar Campanha

Voltar

Fonte: Desenvolvimento próprio.

Para visualizar a campanha recém criada clique em “Minhas Campanhas” ou volte para o início e sua campanha estará em meio a outras campanhas do app, Figura 17:

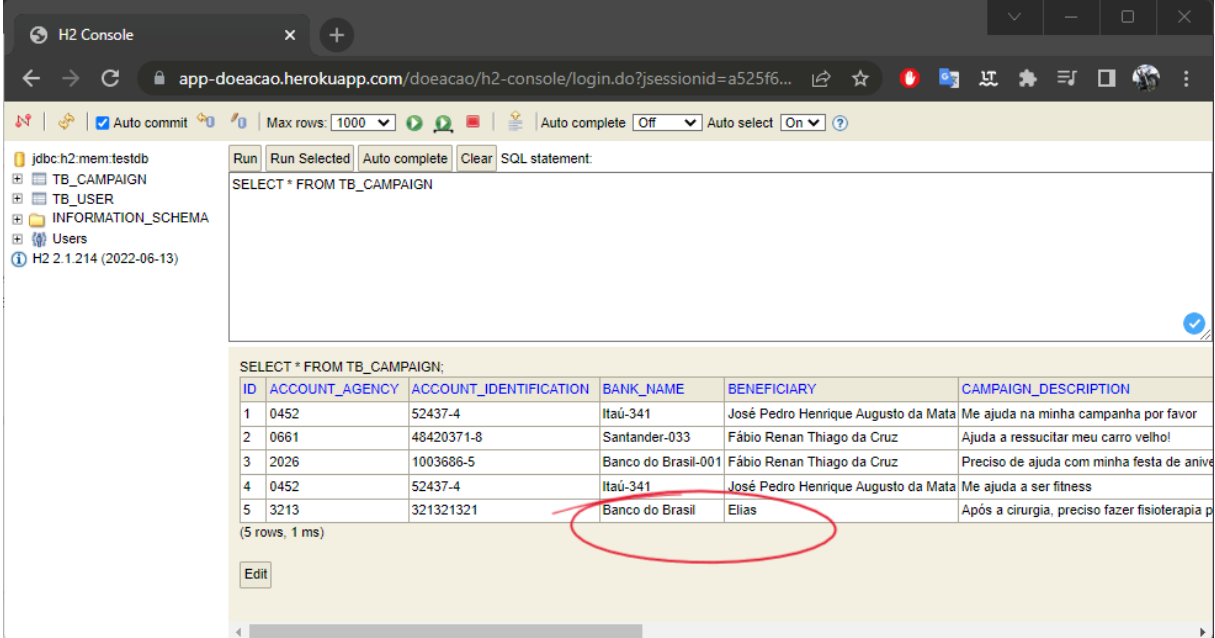
Figura 17 – Campanha criada com sucesso.



Fonte: Desenvolvimento próprio.

Lá no H2, é possível visualizar a campanha criada e o usuário, Figura 18 e Figura 19:

Figura 18 – Campanha criada no banco de dados do projeto.



Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM TB_CAMPAIGN

SELECT * FROM TB_CAMPAIGN;

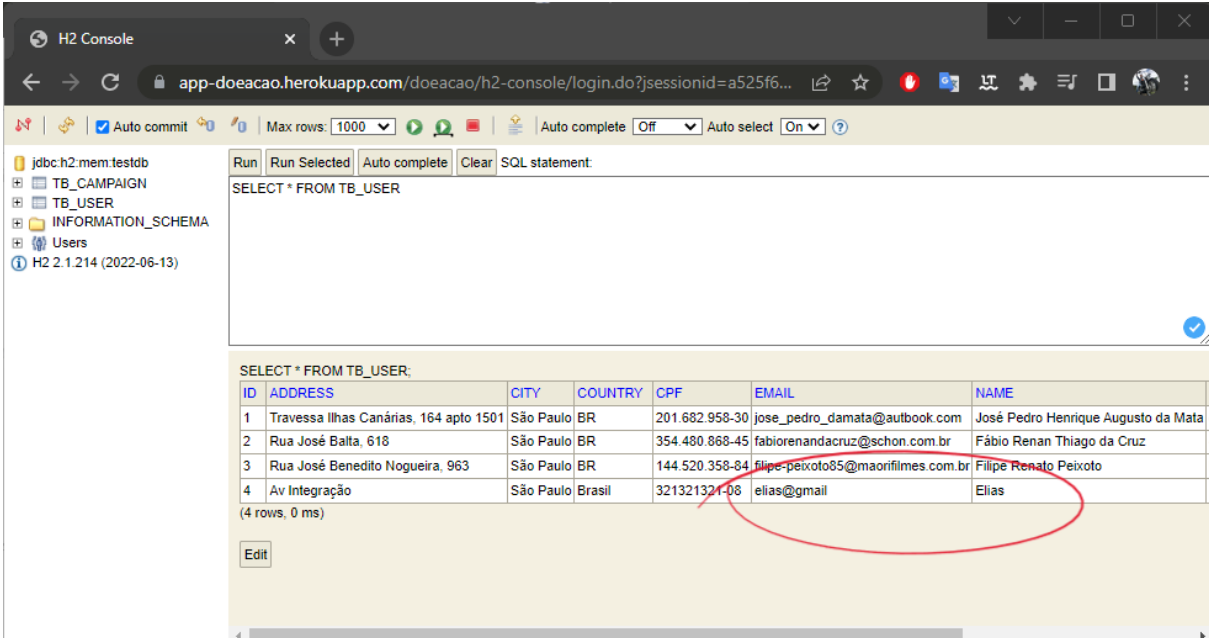
ID	ACCOUNT_AGENCY	ACCOUNT_IDENTIFICATION	BANK_NAME	BENEFICIARY	CAMPAIGN_DESCRIPTION
1	0452	52437-4	Itaú-341	José Pedro Henrique Augusto da Mata	Me ajuda na minha campanha por favor
2	0661	48420371-8	Santander-033	Fábio Renan Thiago da Cruz	Ajuda a ressuscitar meu carro velho!
3	2026	1003686-5	Banco do Brasil-001	Fábio Renan Thiago da Cruz	Preciso de ajuda com minha festa de aniversário
4	0452	52437-4	Itaú-341	José Pedro Henrique Augusto da Mata	Me ajuda a ser fitness
5	3213	321321321	Banco do Brasil	Elias	Após a cirurgia, preciso fazer fisioterapia por favor

(5 rows, 1 ms)

Edit

Fonte: Desenvolvimento próprio.

Figura 19 – Usuário criado no banco de dados do projeto.



Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM TB_USER

SELECT * FROM TB_USER;

ID	ADDRESS	CITY	COUNTRY	CPF	EMAIL	NAME
1	Travessa Ilhas Canárias, 164 apto 1501	São Paulo	BR	201.682.958-30	jose_pedro_damata@autobook.com	José Pedro Henrique Augusto da Mata
2	Rua José Balta, 618	São Paulo	BR	354.480.868-45	fabiorenandacruz@schon.com.br	Fábio Renan Thiago da Cruz
3	Rua José Benedito Nogueira, 963	São Paulo	BR	144.520.358-84	filipe-peixoto85@maorifilmes.com.br	Filipe Renato Peixoto
4	Av Integração	São Paulo	Brasil	321321321-08	elias@gmail	Elias

(4 rows, 0 ms)

Edit

Fonte: Desenvolvimento próprio.

Ao final desse capítulo, percorremos a jornada do usuário que tem a necessidade de criar uma campanha, desde o seu primeiro acesso ao aplicativo, até a conclusão da campanha, conhecendo também o backend e como ele funciona.

10 CONSIDERAÇÕES FINAIS

O projeto DoeAção foi elaborado visando atender às competências de desenvolvimento de *software*, com a definição de uma proposta de projeto que pudesse atender a um determinado público alvo e a jornada de um usuário na aplicação, foram identificadas as dificuldades do projeto, análise do mercado concorrente e definido a partir de então, a visão do produto.

Em seguida foram elaborados os diagramas do projeto, o seu caso de uso, para darmos início ao protótipo inicial do DoeAção. Nessa fase iniciamos o desenvolvimento com ferramentas comuns ao mercado de programação, como, por exemplo, o Figma, aplicamos conceitos de design e trabalhamos para atender a melhor forma possível de uso da aplicação por parte do usuário.

O desenvolvimento da aplicação ocorreu de forma simultânea entre *frontend* e *backend*, utilizando o padrão Ágil de trabalho e desenvolvimento, segmentando as *features* a serem implementadas no projeto e a comunicação entre as ferramentas utilizadas, Spring, HTML, banco de dados H2 e a *cloud* pública Heroku.

O objetivo desse projeto era conseguir integrar diversas ferramentas de desenvolvimento do *Web Service* de maneira funcional e que possibilitasse a continuidade do projeto.

O DoeAção, ao final desse projeto está funcional e escalável, possibilitando a implementação de diversas melhorias que podem ser feitas e atender outras funcionalidades propostas nesse trabalho, como a jornada do usuário que irá realizar a contribuição, opções não implementadas do menu, integrar a aplicação com redes sociais, etc.

Por fim, disponibilizamos todo o nosso desenvolvimento em repositório no GitHub e a documentação da aplicação no link <<https://app-doeacao.herokuapp.com/doeacao/>>.

Sendo assim, visamos atender a melhor forma de apresentar o projeto e atender às exigências da disciplina de Projeto Integrador, com o trabalho em equipe, organização, pesquisa e mão na massa, desenvolver algo o mais próximo de um projeto profissional possível.

REFERÊNCIAS

- AFONSO, Alexandre. **O que é Spring Boot?** 2017. Acesso em: 17.11.2022. Disponível em: <<http://bit.ly/3XtBfyn>>.
- ATLASSIAN, Campany. **What is Agile?** 2022. Acesso em: 18.09.2022. Disponível em: <<https://www.atlassian.com/agile>>.
- BEZERRA, Giuliana. **Gerenciando dependências com o Maven em projetos Java.** 2018. Acesso em: 16.11.2022. Disponível em: <<http://bit.ly/3OlZ3jx>>.
- BINSTOCK, Andrew. **Java's 20 Years Of Innovation.** 2015. Acesso em: 16.11.2022. Disponível em: <<https://www.forbes.com/sites/oracle/2015/05/20/javas-20-years-of-innovation/?sh=7e2886eb11d7>>.
- CASTELANI, Clayton. **Pandemia gera R\$ 7 bilhões em doações e muda investimento social privado.** 2021. Acesso em: 12.09.2022. Disponível em: <<https://bit.ly/3L8nQG0>>.
- CORPORATION, IBM. **H2 database.** 2022. Acesso em: 17.11.2022. Disponível em: <<https://www.ibm.com/docs/en/spm/8.0.2?topic=database-h2>>.
- DEVMEDIA. **Uma visão sobre o Projeto Lombok.** 2013. Acesso em: 17.11.2022. Disponível em: <<https://www.devmedia.com.br/uma-visao-sobre-o-projeto-lombok/28321>>.
- DEVMEDIA. **Introdução a web services RESTful.** 2016. Acesso em: 18.11.2022. Disponível em: <<http://bit.ly/3EPvx29>>.
- DOCS, MDN Web. **JavaScript.** 2022. Acesso em: 15.11.2022. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>.
- DOCS, MDN Web. **CSS: Cascading Style Sheets.** 2022. Acesso em: 15.11.2022. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/CSS>>.
- DOCS, MDN Web. **HTML: HyperText Markup Language.** 2022. Acesso em: 15.11.2022. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/HTML>>.
- FIGMA, Campany. **Creative tools meet the internet.** 2022. Acesso em: 18.09.2022. Disponível em: <<https://www.figma.com/about/>>.
- FONDATION, Apache Software. **Welcome to Apache Maven.** 2022. Acesso em: 16.11.2022. Disponível em: <<https://maven.apache.org/>>.
- FOUNDATION, OpenJS. **What is jQuery?** 2022. Acesso em: 15.11.2022. Disponível em: <<https://jquery.com/>>.
- FREITAS, Eduardo de. **Hierarquia das Cidades no Brasil.** 2019. Acesso em: 15.09.2022. Disponível em: <<https://brasileSCO.la/b9062>>.
- GEEKHUNTER. **O que é Spring Boot?** 2022. Acesso em: 16.11.2022. Disponível em: <<https://blog.geekhunter.com.br/tudo-o-que-voce-precisa-saber-sobre-o-spring-boot/>>.

GOSLING, James; MCGILTON, Henry. *The Java Language Environment*. 1995. Acesso em: 16.11.2022. Disponível em: <<https://www.oracle.com/java/technologies/introduction-to-java.html#318>>.

GRUPO31, Senac. **Repositório GitHub - pi-senac**. 2022. Acesso em: 25.08.2022. Disponível em: <<https://github.com/educobilinski/pi-senac>>.

HEROKU, Salesforce Developers. *How Heroku Works*. 2022. Acesso em: 17.11.2022. Disponível em: <<https://devcenter.heroku.com/articles/how-heroku-works>>.

LIMA, Guilherme. **REST: Conceito e fundamentos**. 2020. Acesso em: 18.11.2022. Disponível em: <<https://www.alura.com.br/artigos/rest-conceito-e-fundamentos>>.

LOCOMOTIVA, Instituição. **Classe Média - Impacto Econômico da Pandemia**. 2021. Acesso em: 15.09.2022. Disponível em: <<https://bit.ly/3qOiTbT>>.

NEGHERBON, Rafael. **Open API: o que é e quais os benefícios**. 2022. Acesso em: 17.11.2022. Disponível em: <<https://transfeera.com/blog/open-api/>>.

RADIGAN, Dan. *What is kanban?* 2022. Acesso em: 18.09.2022. Disponível em: <<https://bit.ly/3RTaQXp>>.

TANZU, Spring By VMware. *Spring Data JPA*. 2022. Acesso em: 18.11.2022. Disponível em: <<https://spring.io/projects/spring-data-jpa#overview>>.

TEAM, Bootstrap. *Build fast, responsive sites with Bootstrap*. 2022. Acesso em: 15.11.2022. Disponível em: <<https://getbootstrap.com/>>.

TEAM, Thymeleaf. *Tutorial: Using Thymeleaf*. 2018. Acesso em: 16.11.2022. Disponível em: <<https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html>>.

TORRES, Joaquim. **O que é e como criar a visão e a estratégia do produto?** 2022. Acesso em: 16.09.2022. Disponível em: <<https://bit.ly/3LkseBL>>.

WEBB, Phillip et al. *Part IV. Spring Boot features*. 2017. Acesso em: 16.11.2022. Disponível em: <<https://docs.spring.io/spring-boot/docs/1.5.2.RELEASE/reference/html/boot-features-testing.html>>.

ZUCHER, Vitor. *Le Wagon*. 2020. Acesso em: 21.11.2022. Disponível em: <<https://www.lewagon.com/pt-BR/blog/o-que-e-padrao-mvc>>.

ZUP. **Spring Boot: como começar**. 2021. Acesso em: 16.11.2022. Disponível em: <<http://bit.ly/3EpDVUK>>.