

UNIVERSIDADE EVANGÉLICA DE GOIÁS - UNIEVANGÉLICA
ENGENHARIA DE COMPUTAÇÃO/ENGENHARIA DE SOFTWARE

**GABRIEL HENRIQUE SILVA
HENRIQUE VALEIRO DE SOUZA
JÚLIO CÉSAR CAIXETA RODRIGUES DA CUNHA
VINICIUS REIS CAMPOS**

**Mínimo Produto Viável - MVP para Arrecadação e Distribuição de
Vestimentas para Pessoas Necessitadas**

Anápolis

2021

UNIVERSIDADE EVANGÉLICA DE GOIÁS - UNIEVANGÉLICA
ENGENHARIA DE COMPUTAÇÃO/ENGENHARIA DE SOFTWARE

**GABRIEL HENRIQUE SILVA
HENRIQUE VALEIRO DE SOUZA
JÚLIO CÉSAR CAIXETA RODRIGUES DA CUNHA
VINICIUS REIS CAMPOS**

**Mínimo Produto Viável - MVP para Arrecadação e Distribuição de
Vestimentas para Pessoas Necessitadas**

Trabalho apresentado ao Curso de Engenharia de Software da Universidade Evangélica de Goiás – UniEVANGÉLICA, da cidade de Anápolis-GO como requisito parcial para obtenção do Grau de Bacharel em Engenharia de Software.

Orientador (a): Prof. Alexandre Moraes Tannus

Anápolis

2021

UNIVERSIDADE EVANGÉLICA DE GOIÁS - UNIEVANGÉLICA
ENGENHARIA DE COMPUTAÇÃO/ENGENHARIA DE SOFTWARE

**GABRIEL HENRIQUE SILVA
HENRIQUE VALEIRO DE SOUZA
JÚLIO CÉSAR CAIXETA RODRIGUES DA CUNHA
VINICIUS REIS CAMPOS**

**Mínimo Produto Viável - MVP para Arrecadação e Distribuição de
Vestimentas para Pessoas Necessitadas**

Monografia apresentada para Trabalho de Conclusão de Curso de Engenharia de Software da Universidade Evangélica de Goiás - UniEVANGÉLICA, da cidade de Anápolis-GO como requisito parcial para obtenção do grau de Engenheiro(a) de Software.

Aprovado por:

(ORIENTADOR)

(AVALIADOR)

Anápolis

2021

FICHA CATALOGRÁFICA

SILVA, Gabriel Henrique; SOUZA, Henrique Valeiro; CUNHA, Júlio César Caixeta Rodrigues; CAMPOS, Vinicius Reis; **Mínimo Produto Viável - MVP para Arrecadação e Distribuição de Vestimentas para Pessoas Necessitadas**. Anápolis, 2021. (Universidade Evangélica de Goiás – UniEVANGÉLICA, Engenheiro(a) de Software, 2021).

Monografia. Universidade Evangélica de Goiás, Curso de Engenharia de Software, da cidade de Anápolis-GO.

1. MVP. Vestimentas. Doação.

REFERÊNCIA BIBLIOGRÁFICA

SILVA, Gabriel Henrique; SOUZA, Henrique Valeiro; CUNHA, Júlio César Caixeta Rodrigues; CAMPOS, Vinicius Reis; **Mínimo Produto Viável - MVP para Arrecadação e Distribuição de Vestimentas para Pessoas Necessitadas**. Anápolis, 2021. 31 páginas. Monografia - Curso de Engenharia de Software Universidade Evangélica de Goiás - UniEVANGÉLICA.

CESSÃO DE DIREITOS

NOMES DOS AUTORES: GABRIEL HENRIQUE SILVA, HENRIQUE VALÉRIO DE SOUZA, JÚLIO CÉSAR CAIXETA RODRIGUES DA CUNHA, VINICIUS REIS CAMPOS

TÍTULO DO TRABALHO: MÍNIMO PRODUTO VIÁVEL - MVP PARA ARRECADAÇÃO E DISTRIBUIÇÃO DE VESTIMENTAS PARA PESSOAS NECESSITADAS

GRAU/ANO: Graduação / 2021

É concedida à Universidade Evangélica de Goiás - UniEVANGÉLICA, permissão para reproduzir cópias deste trabalho, emprestar ou vender tais cópias para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste trabalho pode ser reproduzida sem a autorização por escrito do autor.

Gabriel Henrique Silva

Henrique Valério de Souza

Júlio César Caixeta Rodrigues da Cunha

Vinicius Reis Campos

RESUMO

Ações sociais como doações e serviços voluntários são temas sempre em alta para se obter contribuições para diversas causas, sejam elas naturais ou não. A evolução das tecnologias digitais e o acesso a uma gama cada vez maior da população à internet, possibilita o surgimento de plataformas para contribuir com causas sociais e a disseminação de informações pelas redes sociais. Vendo que diversas pessoas têm dificuldade de encontrar lugares para doações de vestimentas devido a não conhecer pessoas que realizam esse tipo de ação. Este trabalho tem como objetivo desenvolver um software para a arrecadação e distribuição de vestimentas para pessoas necessitadas. O software será desenvolvido usando o formato Mínimo Produto Viável - MVP.

Palavras-chave: Ações Sociais; Doações; Campanhas; Vestimentas; MVP.

ABSTRACT

Social actions such as donations and volunteer services are always on the rise to obtain contributions to various causes, whether natural or not. The evolution of digital technologies and the access to an increasing range of the population to the internet, allows the emergence of platforms to contribute to social causes and the dissemination of information through social networks. Seeing that many people have difficulty finding places for clothing donations due to not knowing the places or people who carry out this type of action. This work aims to develop a software for collecting and distributing clothing to needy people. The software will be developed using the Minimum Viable Product - MVP format.

Keywords: Social Actions; Donations; Campaigns; Clothing; MVP.

LISTA DE ILUSTRAÇÕES

Figura 1 - MVP – Modo correto e modo incorreto.....	13
Figura 2 - Quadro do Kanban	14
Figura 3 - Ciclo da Sprint	15
Figura 4 - Início e fim da TAG.....	19
Figura 5 - O corpo humano como uma página web	20
Figura 6 - Comunicação direta do cliente para o serviço	21
Figura 7 - Divisão de um aplicativo monolítico em microsserviços	22
Figura 8 - Processo Elixir	23
Figura 9 - Quadro Trello.....	25
Figura 10 - Formulário para levantamento de requisitos.....	26
Figura 11 - Continuação do formulário para levantamento de requisitos	27

LISTA DE ABREVIATURAS E SIGLAS

Siglas	Descrição
ONG	Organização Não Governamental
MVP	Minimum Viable Product
TDD	Test Driven Development
HTML	HyperText Markup Language
WEB	World Wide Web
URLs	Uniform Resource Locators
HTTP	HyperText Transfer Protocol
CSS	Cascading Style Sheets

SUMÁRIO

1 INTRODUÇÃO	10
1.1 Problema da pesquisa.....	10
2 OBJETIVOS	10
2.1 Objetivo Geral	10
2.2 Objetivos Específicos	10
3 JUSTIFICATIVA	11
4 CRONOGRAMA	12
5. FUNDAMENTAÇÃO TEÓRICA.....	13
5.1 Mínimo produto viável (MVP)	13
5.2. Processos de desenvolvimento	14
5.3 Desenvolvimento Web	17
5.3.1 Front-end e suas tecnologias	18
5.3.2 Back end e suas tecnologias	20
6. METODOLOGIA.....	24
7. RESULTADOS	28
REFERÊNCIAS.....	29

1 INTRODUÇÃO

1.1 Problema da pesquisa

Seja por questões de administração econômica, acontecimentos climáticos ou desastres causados pela humanidade, sempre há um número alto de pessoas que necessitam de doações. Alimentos, moradia, cobertores e vestimentas, são exemplos de recursos que são arrecadados para ajudar pessoas que passam por dificuldades como as acima citadas. Órgãos governamentais, ONGs, pessoas jurídicas e físicas criam ações para arrecadamento de suprimentos, buscando ajudar ao próximo.

As campanhas para arrecadação de doações ocorrem normalmente através de veículos de comunicação como panfletos, noticiários, redes sociais e comunicação verbal. No entanto, com o avanço das tecnologias digitais, novas formas de divulgação podem surgir e serem mais eficazes, otimizando o alcance de divulgação e arrecadação de produtos. Existe uma forma para doar ou receber de forma simples e rápida? Esses recursos são confiáveis? Diante desses questionamentos, o problema desta pesquisa é: Considerando o impacto da internet nos tempos atuais, o desenvolvimento de um software para a arrecadação e distribuição de vestimentas para pessoas necessitadas se faz relevante?

2 OBJETIVOS

2.1 Objetivo Geral

Desenvolver um software no formato MVP para o arrecadamento e distribuição de vestimenta para pessoas necessitadas.

2.2 Objetivos Específicos

- Realizar um formulário para verificar a viabilidade
- Realizar o levantamento de requisitos necessários para o desenvolvimento do software;
- Mapear de forma lógica e conceitual a base de dados para a armazenagem correta dos dados;

- Prototipar as telas para ter uma visualização prévia de como será a interface;
- Desenvolver o backend e frontend da aplicação;
- Realizar testes para verificar o nível de qualidade do sistema;

3 JUSTIFICATIVA

A doação de vestimentas é algo que ocorre no mundo todo. Como por exemplo Anápolis, teve no último dia 10 de junho uma arrecadação de 500 itens por uma iniciativa da Ala 2 da Base Aérea de Anápolis utilizando como canal principal de veiculação o site da prefeitura (ANÁPOLIS, 2021). Partindo para a capital de Goiás, Goiânia utilizou seu canal de comunicação oficial para divulgar a campanha de doação de agosto de 2021 e fez uso de número fixo como canal principal para doações (GOIÂNIA, 2021).

Tendo em vista o esforço manual para a divulgação e realização de doações, foi realizada uma pesquisa em diversas plataformas como a Google, Play Store, Apple Store, Microsoft Store, Samsung Store, na intenção de encontrar uma aplicação que divulgasse locais para doar e receber vestimentas. Após a realização de uma busca foram encontrados alguns aplicativos que cumprem parcialmente a ideia proposta neste estudo, um exemplo é o aplicativo Paraná Solidário.

O aplicativo Paraná Solidário é um sistema com o intuito de promover a doação direta entre os cidadãos e as entidades beneficentes, fazendo uma ponte com empresas e pessoas físicas que tenham interesse em fazer doações (PARANÁ, 2019). Entretanto, o aplicativo possui alcance apenas para a localização do Paraná, enquanto o sistema que será desenvolvido, procura abranger uma área maior de alcance, podendo futuramente ser utilizado em qualquer localidade do país.

Diante desse cenário, esta pesquisa busca ajudar pessoas necessitadas por meio do desenvolvimento de um software para arrecadação, distribuição e divulgação de locais para doar e receber vestimentas. Por meio desse software, surge a oportunidade de ampliar o canal de comunicação entre o doador e as partes interessadas em realizar a coleta e a distribuição.

Com o acesso ao sistema, a pessoa poderá buscar entidades que estarão arrecadando vestimentas nas proximidades da região. Pessoas, empresas e órgãos locais serão avaliados e cadastrados no sistema por um administrador em primeiro momento, trazendo ao sistema e ao local de doação mais segurança e confiabilidade.

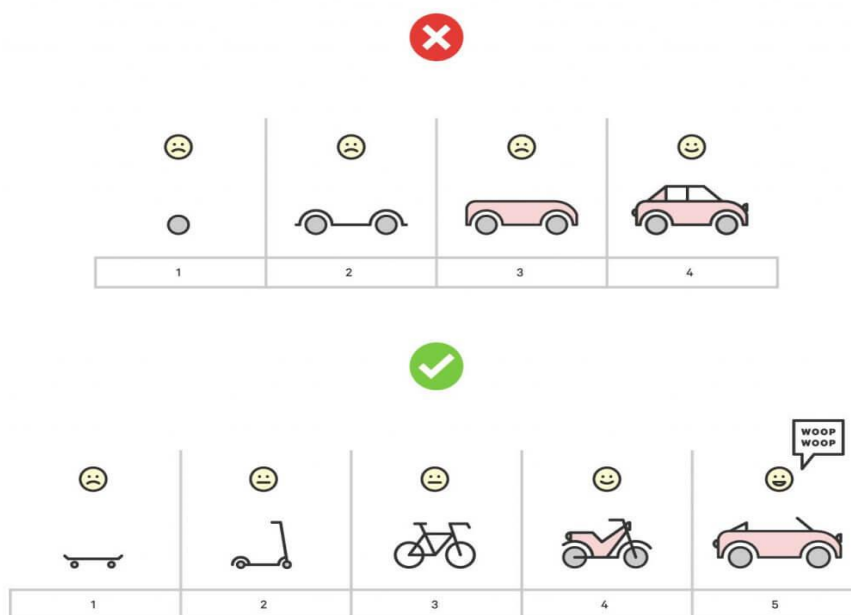
5. FUNDAMENTAÇÃO TEÓRICA

5.1 Mínimo produto viável (MVP)

O surgimento do termo mínimo produto viável ou do inglês *minimum viable product* - MVP vem de encontro com o conceito de Startup Enxuta, ou *Lean Startup*. De acordo com Ries (2011), o “MVP é aquela versão do produto que permite uma volta completa do ciclo construir-medir-aprender, com o mínimo de esforço e o menor tempo de desenvolvimento”.

O conceito mais básico e propagado de um MVP é a construção de um carro, ao qual inicia o desenvolvimento do produto realizando da forma mais simples possível, incluindo apenas rodas e um suporte para que possa ficar em cima. Após lançar essa versão, são colhidos os feedbacks dos usuários e então um novo ciclo será realizado para aprimorar o produto, como vemos a seguir na figura 1.

Figura 1 - MVP – Modo correto e modo incorreto



Fonte: MVP - Serasa Empreendedor (2018)

De acordo com Ries (2009), "O mínimo produto viável é a versão de um novo produto que permite que uma equipe colete o máximo de aprendizado validado sobre os clientes com o mínimo esforço". O MVP não se trata de construir um produto final rápido, mas sim despende

tempo para aprender algo com a primeira iteração do produto e para que isso ocorra, existe uma quantidade alta de energia que deve ser investida para metrificar, colher feedback de clientes e realizar análises cruciais.

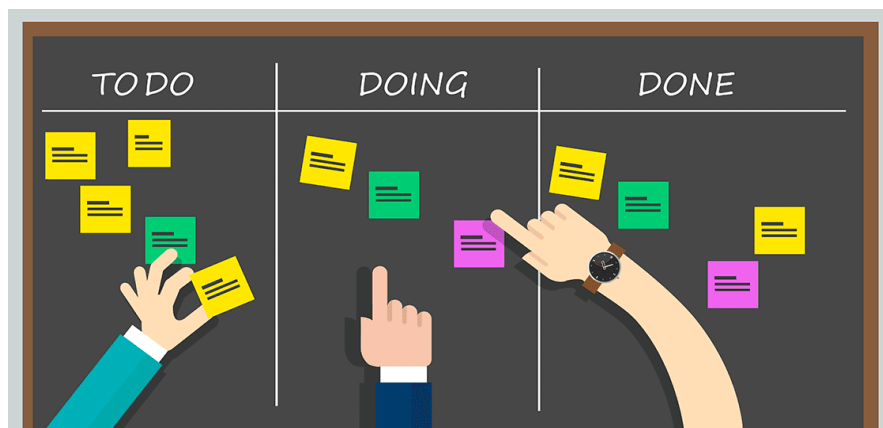
5.2. Processos de desenvolvimento

O Kanban foi desenvolvido no Japão pela empresa Toyota, é um sistema ágil e visual de cartões de cores para designar e administrar tarefas para melhor controle de fluxos. O foco do Kanban é priorizar a produtividade e a organização das entregas. O objetivo é proporcionar um trabalho mais transparente e direcionado.

Os cartões são utilizados para definir prioridades, pessoas ou tipo de tarefa. O que se trata de uma ação ou tarefa que precisa ser realizada para que o resultado final seja entregue. O Kanban geralmente possui três colunas, a fazer, em execução e feito. Os cartões devem ser movidos entre elas de acordo com os status que forem mudando, tendo uma visão do que precisa ser feito até o que já foi concluído. O quadro é a junção dos cartões com as colunas, geralmente ficando em um ponto de fácil acesso e visão para todos, uma equipe pode trabalhar com mais de um quadro, conforme mostrado na Figura 2.

Segundo Prikladnicki (2014), o termo Kanban tornou-se mais diretamente relacionado a este modelo de visualização do status dos itens em desenvolvimento. Essa correlação gira em torno da ideia de que cada ficha do quadro representa um sinal que "permite" a equipe de trabalho realizar o que está indicado no cartão.

Figura 2 - Quadro do Kanban



Fonte: Agile, Scrum e Kanban - Heflo (2017)

Já o Scrum é um framework ágil, um conjunto de boas práticas usadas no gerenciamento de projetos complexos. Tem como base as equipes pequenas, as reuniões constantes e a colaboração dos envolvidos.

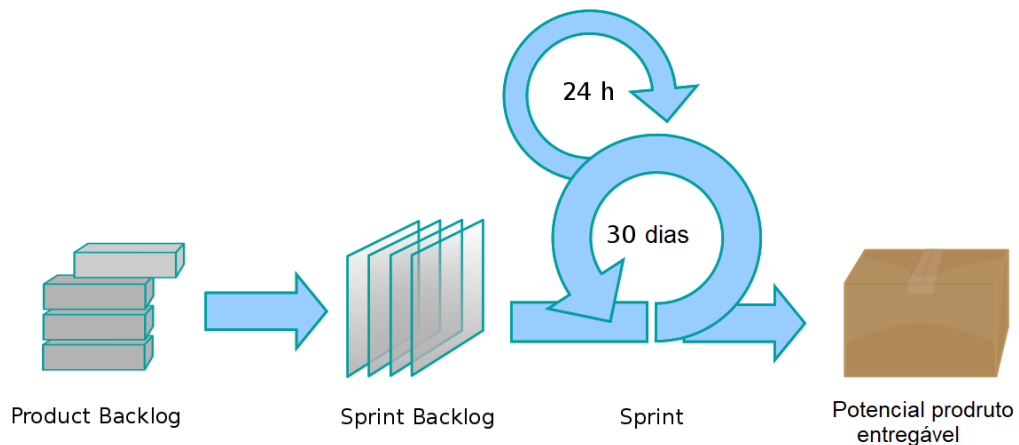
Segundo Prikladnicki (2014) o Scrum maximiza efetivamente a entrega de software, adaptando-se às novas realidades. Desenvolve os recursos mais valiosos com antecedência, enquanto considera se os recursos de prioridade mais baixa são necessários. Se mudanças forem necessárias, a equipe ágil pode facilmente alterar a prioridade. Sua principal motivação é que o desenvolvimento de software envolve muitas variáveis técnicas, como requisitos, recursos e tecnologia, que podem mudar no decorrer do processo, assim tornando-o imprevisível e precisando acompanhar as mudanças com flexibilidade.

A essência do Scrum é a sprint. Sprints tem durações de acordo com o esforço de desenvolvimento da equipe de desenvolvimento. Sprint são os ciclos do projeto, ou seja, cada ciclo é um sprint. Uma sprint se inicia assim que uma anterior é encerrada. É um evento time-boxed de até um mês, durante o qual um incremento potencialmente utilizável do produto é criado, conforme mostrado na Figura 3. As sprints têm durações coerentes em todo o esforço de desenvolvimento. Existem 3 funções principais, elas são Product Owner, Scrum Master e Equipe de desenvolvimento.

O Product Owner é o responsável pelo backlog do produto (uma lista de itens sobre os quais a equipe de desenvolvimento trabalhará no decorrer do projeto), por ajustar funcionalidades e prioridades a cada sprint e decidir a data de liberação e conteúdo das atualizações.

Scrum Master é o responsável por facilitar o trabalho da equipe, é a pessoa que dita os prazos, além de ser quem facilita e auxilia, mas não é o gerente do projeto, garante a remoção dos impedimentos que dificultam o trabalho da equipe de desenvolvimento.

Equipe de desenvolvimento é um grupo multidisciplinar de pessoas, responsável pelo desenvolvimento do produto, a cada sprint a equipe de desenvolvimento gera um incremento do produto e que significa um avanço visível para os clientes do projeto.



Fonte: Autores

O Scrumban é a junção das melhores características do Scrum e do Kanban, usando o Scrum de base e o Kanban de complemento. O Scrumban é indicado a equipes que ainda não estão acostumadas a métodos ágeis de desenvolvimento, projetos de sustentação e que há muita ocorrência de histórias frequentes, projetos de manutenção e demandas inesperadas.

A principal diferença entre o Scrum e o Scrumban é a ausência das sprints e a inclusão do fluxo de trabalho contínuo com limitação do trabalho em progresso. Entre kanban e Scrumban, a diferença é que o Scrumban adiciona alguns detalhes do Scrum, como os papéis necessários (Product Owner, Scrum Master e Equipe de desenvolvimento) e as reuniões diárias.

Com o processo ágil do Scrumban, facilitando na identificação das tarefas e gestão das mesmas, será utilizado o TDD, que é o Desenvolvimento Orientado por Testes (Test Driven Development). Os desenvolvedores criam um teste que irá falhar de todo jeito. Afinal, o mesmo ainda não foi implementado. Em seguida, a equipe desenvolve um código que satisfaça esse teste e então reaplicar a ele. Se o resultado é satisfatório, a equipe implementa o novo recurso ao código, e então partem para o desenvolvimento de um novo teste. Essa primeira implementação deverá satisfazer imediatamente o teste que foi escrito no ciclo anterior;

Ele se baseia em um ciclo curto de repetições, no qual para cada funcionalidade do sistema, um teste é criado antes. Esse ciclo é repetido até o final do projeto, quando o programa ou aplicativo é finalizado.

Com o TDD os testes serão escritos antes do código de produção e para cada novo ciclo, haverá um teste. Isto garante mais qualidade no desenvolvimento do código, já que eles deverão ser sempre refeitos e melhorados. Também dá mais segurança à equipe de desenvolvimento, com menos chances de o software apresentar algum bug ao final do processo. Segundo Kaner

(2004), a automação mudou fundamentalmente a função dos testes em todo o processo de desenvolvimento.

O local para manter o repositório sempre atualizado e muito bem dividido, será utilizado o Git. Git é um sistema de gerenciamento e controle de versões. Desenvolvido em 2005 por Linus Torvalds, o mesmo criador do kernel do Linux. Um grande número de projetos utiliza o Git para versionar seus códigos. Com ele podemos criar projetos onde diversos usuários podem criar, editar, adicionar e excluir arquivos sem que tenham riscos de suas modificações serem sobrescritas.

O software de controle de versão armazena o registro de todas as alterações realizadas no código. Caso seja cometido um erro, os desenvolvedores podem “voltar no tempo”, restaurar uma versão mais antiga, onde não existia o erro e ajudar a corrigir o código e ao mesmo tempo, reduzir a interferência para todos os membros da equipe.

Segundo Martin (2020), o tempo de verificação usando o Git foi reduzido a quase nada. A ideia nem existe mais. Assim, podendo fazer commit em qualquer versão sempre que achar necessário. Os conflitos entre esses commits serão resolvidos quando os programadores bem entenderem. Com isso, é possível criar um conjunto de testes abrangente e rápido para testar quase tudo.

Pensando em agilizar ainda mais o processo com o Git, foi decidido utilizar o Git Flow, que é um tipo alternativo de branch do Git que consiste em ramificações de recursos e diversas ramificações primárias. Podemos definir o GitFlow como uma metodologia de trabalho, com foco na entrega de projetos. Seus ideais não geram nada de novo ao Git. Este modelo usa branches fixas. A branch master é utilizada como ponte entre o código principal e o de desenvolvimento. A branch development é utilizada para inserir novas modificações no código de desenvolvimento. Caso não haja erros, a modificação é inserida na branch master.

5.3 Desenvolvimento Web

Desde o começo da web, vem crescendo de forma excepcional. Algumas décadas atrás, a web fornecia apenas sites estáticos, simples e construídos usando somente uma linguagem de marcação que é chamada de HTML. Esses sites possuíam unicamente textos e imagens. Ao longo dos anos, a internet continuou a progredir, seguindo-se o aparecimento de outras tecnologias que contribuíram para o desenvolvimento do site.

Atualmente, a web não é mais apenas um ambiente de páginas simples e passou a ser tornar uma plataforma de aplicações. Como a revolução da web e com surgimento de novas ferramentas para o desenvolvimento, as aplicações web passaram a se tornar mais difícil. Assim o número de usuários e servidores disponíveis crescia substancialmente, a busca por aplicações cada vez melhores se fez indispensável. conseqüentemente, com o aumento de novas tecnologias envolvidas, foram exigindo cada vez mais de profissionais na área. Segundo Coulouris (2013), a web evoluiu bastante sem mudar sua arquitetura básica que são divididas em três básicos componentes:

- HTML (*HyperText Markup Language*), que é uma linguagem de marcação aplicada para estruturar elementos da página, como textos, tabelas, imagens e até vídeos.
- URLs (*Uniform Resource Locators*), é o endereço virtual de uma página Web, como exemplo: www.unievangelica.edu.br.
- Arquitetura cliente e servidor que utilizam o padrão de comunicação através do protocolo HTTP (*HyperText Transfer Protocol*), que os navegadores usam para se comunicar e buscar informações, como documentos e outros recursos.

Portanto, para o desenvolvimento de sistemas web junto com os componentes citados, é também necessário o uso de linguagens de programação voltadas para web, o que faz com que se reparte em duas partes na sua construção, denominadas de Front-end (lado do cliente) e Back-end (lado do servidor).

5.3.1 Front-end e suas tecnologias

Andrade (2018) define front-end como sendo todo o código da aplicação responsável pela apresentação do software. Para Viana (2017) o front-end resultará na interação do usuário com o sistema. O front-end é o responsável por exibir e coletar as informações da interface gráfica do sistema. Dentro desta parte trabalha-se principalmente com HTML, CSS, JavaScript, entre outras tecnologias para a composição de interfaces gráficas (páginas web).

Segundo Miletto (2014), o HTML é uma linguagem de marcação utilizada para criação das páginas de internet. É através do HTML que é definido o formato que as informações devem se apresentar ao usuário nos navegadores de internet, ex Google Chrome, Mozilla Firefox, Internet Explorer. Conforme Eis e Ferreira (2012, p.26), o “HTML é baseado no conceito de Hipertexto, que define um conjunto de elementos ligados por conexões, e esses elementos

podem ser textos, vídeos, imagens, gifs e documentos”. Todos os elementos que compõem a página são inseridos por meio de comandos específicos da linguagem, nomeados de TAGs. Para Miletto (2014, p.72), “TAG como palavra específica, definida em HTML, envolta por sinais de “menor que” (<) e “maior que” (>). De um modo geral, as TAGs aparecem em pares, uma indicando o início e a outra indicando o fim da marcação”, conforme é mostrado na Figura 4:

Figura 4 - Início e fim da TAG



Fonte: Miletto (2014)

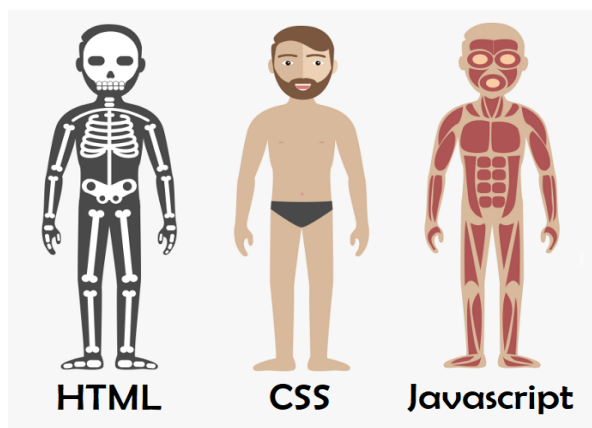
As páginas HTML não têm detalhes, cores ou estilos. Conforme Oliveira (2020, p.35) esta transformação pode acontecer usando “as Folhas de Estilo em Cascata (CSS ou Cascading Style Sheet), permitem customizar a formatação dos elementos HTML”. O CSS é o responsável por organizar todos os tipos de informações presentes na web, para que sejam exibidas da melhor maneira aos usuários. A tecnologia CSS tem uma função exclusiva de estilo de página, através da estilização pode ser atribuído alguns efeitos. Por exemplo, é o uso de atributos, como ao passar o mouse em uma estrutura ocorre a mudança da cor das letras.

Por último, temos o JavaScript que é uma linguagem de programação orientada a objetos semelhante às linguagens C++, C e Java. O nome oficial da linguagem, segundo a especificação ECMA-262, é ECMAScript, pois a linguagem foi padronizada e estabilizada pela associação European Computer Manufacturers Association (ECMA), e conta com diversas implementações do padrão (MILETTO, 2014). Por exemplo, movimentos do mouse, pressionar botão, arrastar e soltar etc.

O JavaScript conta com um interpretador que executa operações em tempo real sem a necessidade de enviar dados para o servidor, executando no próprio navegador. Portanto, por meio do JavaScript, podemos realizar a validação de dados e outras operações, esta possibilidade de processar informações no próprio navegador permite que o servidor não se sobrecarregue.

Portando, na Figura 5 a seguir comparando as tecnologias citadas (HTML, CSS e JavaScript) com o corpo humano:

Figura 5 - O corpo humano como uma página web



Fonte: Front-end - Alura (2021)

Para o desenvolvimento do front-end (interface do usuário) será utilizado a biblioteca complementar do JavaScript React. Ela foi criada em 2013 pela empresa Facebook, esta biblioteca é voltada para a construção de interfaces web, sendo capaz de manipular elementos visuais de forma que se possa criar componentes reutilizáveis.

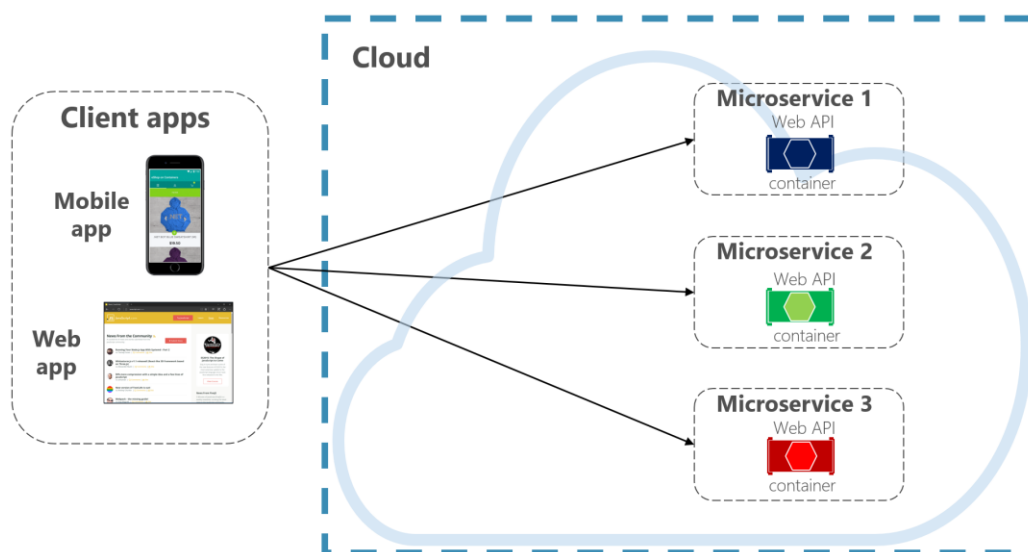
O motivo de utilizar a biblioteca React neste trabalho é a performance dessa tecnologia, pois trabalhar com Virtual Document Object Model (DOM), isto é, o React trabalha com uma cópia virtual do DOM da página web, sendo copiado para a memória do computador, e por meio dele é possível fazer a manipulação da árvore de elementos do HTML (REACT, 2021c).

5.3.2 Back end e suas tecnologias

A camada de back-end é uma aplicação de abstração do relacionamento e da comunicação e controle de dados entre a interface de usuário (front-end) e o banco de dados. É uma parte que não é visível para os usuários, contendo a lógica de negócio relativo ao projeto solicitado que é considerado uma informação privada e restrita ao negócio (dados pessoais de usuários, informações de funcionários etc.) além de ser de suma importância para o funcionamento correto de software. O código fonte do back-end e o sistema de banco de dados geralmente residem e são processados em um servidor.

O back-end tem o papel de proporcionar diversos pontos chaves de comunicação com a camada front-end de acordo com a necessidade da aplicação de consumir e persistir dados do banco de dados para que o front-end possa recuperar (pegar) informações e exibi-las na interface de usuário de maneira apropriada, mostrado na Figura 6.

Figura 6 - Comunicação direta do cliente para o serviço

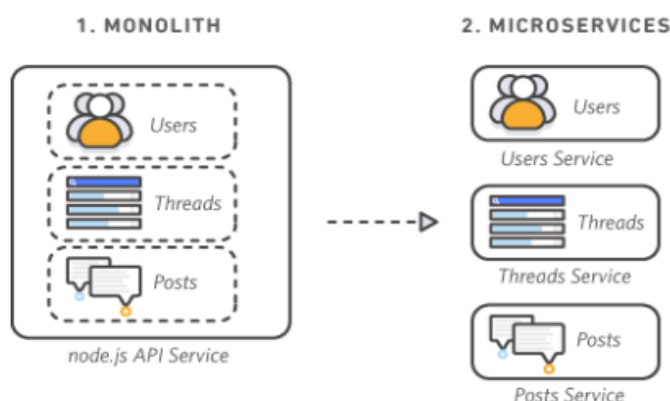


Fonte: Comunicação front-end de cliente - Microsoft (2021)

A Figura 6 é uma imagem representativa de um sistema de baixa confiabilidade de uma comunicação entre o front-end e microserviços de back-end utilizados para fazer a comunicação entre os mesmos, “Microserviços são uma abordagem arquitetônica e organizacional do desenvolvimento de software na qual o software consiste em pequenos serviços independentes que se comunicam usando APIs bem definidas.” (AWS, 2021).

Podemos definir um sistema entre arquiteturas monolítica e de microserviços, no monolítico todas as funções do negócio estão implementadas em um único processo e caso uma função tenha uma alta de acesso e necessite que seja escalada, será feita de todo processo por ter suas funcionalidades todas interligadas entre si em um único processo. Já os microserviços tem suas funcionalidades divididas em processos separados e independentes, possibilitando que seja escalado só a funcionalidade que está tendo uma demanda maior, na Figura 7 abaixo será mostrado a divisão de um aplicativo monolítico em microserviços.

Figura 7 - Divisão de um aplicativo monolítico em microsserviços



Fonte: Microsserviços - AWS (2021)

Os monólitos e microsserviços tem suas vantagens e desvantagens, as vantagens do um sistema monolítico e que o código tem uma curva de aprendizado melhor pois todas as suas funções estão vinculadas em um único sistema, além de sua manutenção ser feita em único local, porém caso algum problema todo o conjunto do processo será prejudicado e sairá do ar. Já os microsserviços têm sua curva de aprendizagem mais difícil por ter suas funções separadas em processos diferentes além de sua manutenção ser feita em vários locais, porém caso tenha algum problema um dos processos só conjunto continuará funcionando sem o mesmo e quando ele for corrigido tudo funcionará como antes.

No mercado há diversas linguagens de programação e plataformas de desenvolvimento para as tecnologias do back-end, já que é uma camada com implementação desacoplada do contexto da aplicação. Elixir é uma linguagem de programação funcional executada na máquina virtual do Erlang. “Erlang é uma linguagem de programação usada para construir sistemas soft em tempo real altamente escaláveis com requisitos de alta disponibilidade.” (WINBLAD, 2021), sendo utilizada em vários setores de TI como banco de dados, comércio eletrônico, setores de telecomunicações, telefonia por computador e mensagens instantâneas.

A escalabilidade é gerada graças ao elixir ser executada dentro de threads leves de execução (chamados de processos) trocando informações por meio isolado utilizando mensagens. Possibilitam uma máquina fazer execução de centenas de milhares de processos simultaneamente sem pausas em todo sistema podendo utilizar a de forma mais eficiente o possível graças ao isolamento de seus processos além de trabalhar com a imutabilidade diminuindo possíveis conflitos de informação. A imutabilidade apresenta como conceito algo

que não pode ser alterado como objetos e variáveis que precisam estar em um estado imutável após serem criados, porém não garante que os valores serão sempre os mesmos, na Figura 8 abaixo temos um exemplo de um processo.

Figura 8 - Processo Elixir

```
current_process = self()

# Spawn an Elixir process (not an operating system one!)
spawn_link(fn ->
  send(current_process, {:msg, "hello world"})
end)

# Block until the message is received
receive do
  {:msg, contents} -> IO.puts(contents)
end
```

Fonte: Recursos da plataforma - Elixir (2021)

Com a necessidade de fazer uma definição de um banco de dados para fazer a armazenagem, modificações e requisições de informações para que possa ter uma interação completa com o back-end e front-end. Os sistemas de bancos de dados SQL são tradicionalmente utilizados, por consistir em uma estrutura de dados de forma relacional apresentando os dados em tabelas no modelo linhas e colunas.

O sistema de banco de dados escolhido para o desenvolvimento deste back-end utilizado neste trabalho foi o PostgreSQL, ele é um sistema que consegue lidar bem com altos volumes de solicitações e com cargas de trabalho grandes, funcionando bem com sites que tenham uma intensa demanda de acesso, trazendo um desempenho otimizado para um sistema que recebem simultâneos acessos, apresentando uma boa escolha para se trabalhar com a linguagem Elixir que trabalha com milhares de processos simultâneos.

Além de todos esses aspectos o PostgreSQL é um sistema de banco de dados relacional de objeto de código aberto. De acordo com o site <https://www.postgresql.org/>, o sistema tem mais de 30 anos e possui uma ótima gama de documentos e informações na internet. Atualmente, existem vários profissionais com conhecimento na área, o que facilita a busca de

informações para um melhor desempenho dentro das necessidades de cada projeto a ser desenvolvido.

6. METODOLOGIA

Esta pesquisa possui uma natureza aplicada, pois busca gerar o conhecimento para a aplicação prática e dirigida à solução de um problema específico. Tem como objetivo o desenvolvimento de uma aplicação web, que poderá contribuir para causas humanitárias, trazendo um alcance maior de visibilidade para pessoas que visam arrecadar ou realizar doações.

Esta pesquisa busca solucionar um problema da sociedade, por meio de estudos sistemáticos, pesquisas e investigação. Nesse sentido, a classificação do ponto de vista dos objetivos, é de uma pesquisa exploratória, visto que sua finalidade é levantar dados sobre como é arrecadado e distribuído às vestimentas

Utilizar-se-á o método de levantamento de dados e, a partir dessas informações serão realizados estudos significativos a fim de analisar a sua facilidade ou não de adesão ao produto.

Primeiramente será realizada a definição dos papéis para essa escolha, tem-se como base os tipos aos quais os autores as classificam em duas modalidades: Tradicionais - possuem foco na documentação de cada etapa do desenvolvimento; e Ágeis - podendo ser considerada um novo paradigma de desenvolvimento de Software, visando a redução de formalidades e priorizando a resposta rápida as mudanças de requisitos e ambiente.

Seguindo pela linha ágil, ela utiliza de seus artifícios e práticas para a satisfação de todos os envolvidos no projeto. Dentro desta modalidade, existe um framework bastante popular, cujo nome é Scrum, a qual apresenta uma abordagem sustentada pela experiência e pela observação, aplicando alguns conceitos da teoria de gerenciamento e controle de processos. Considerada pertinente para o projeto e pela experiência dos integrantes da equipe, foi escolhido o Scrum para gerir as etapas do projeto.

De acordo com os papéis dentro do Scrum, foi determinado para cada integrante dentro da equipe um papel. A seleção foi feita também utilizando apenas dois papéis, sendo eles o Scrum Master e Product Owner. A separação de papéis ficou como pode ser visto na tabela 1.

Tabela 1: Stakeholders

CARGO	RESPONSABILIDADE	NOME
Scrum Master	Responsável pelo projeto	Gabriel Henrique
Product Owner	Responsável pelos requisitos do projeto	Vinicius Reis
Team	Responsável pelo desenvolvimento front-end	Júlio César
Team	Responsável pelo desenvolvimento do back-end	Henrique Valério

Vale ressaltar que independente do cargo dos integrantes, todos participarão da escrita do código fonte do software. Finalizando as nomeações, parte-se para a criação e distribuição das tarefas a serem desenvolvidas. Essa atividade será realizada com a utilização do software conhecido como Trello, permitindo a criação de um *board*, *columns* e *cards*, assim sendo, possibilitando uma organização das tarefas de acordo com o estado que ela se encontra. A Figura 9 mostra o quadro no Trello definido inicialmente.

Figura 9 - Quadro Trello

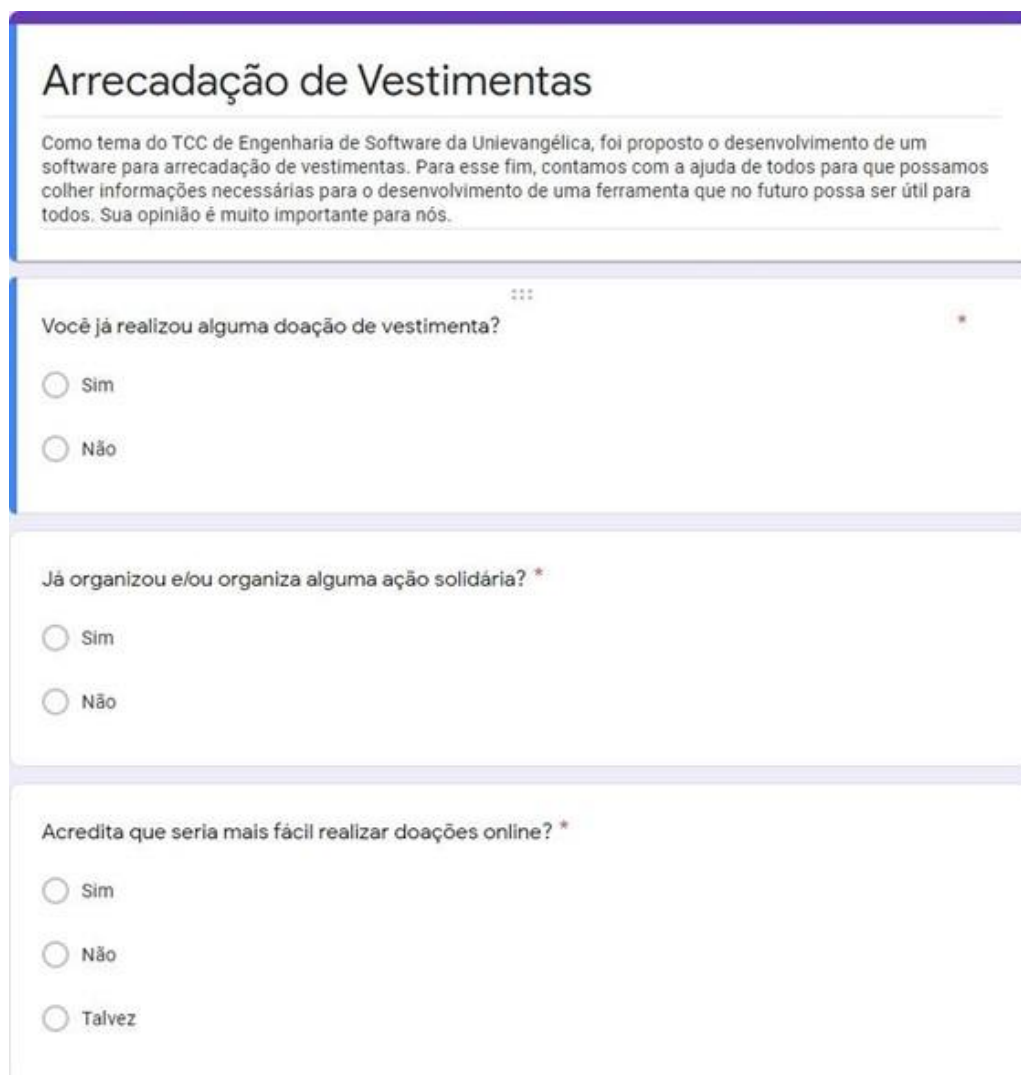


Fonte: Autores

Em seguida, foi desenvolvido um formulário para levantamento de requisitos dando entrada na concepção do projeto, foi realizada a concepção de um formulário web para obter uma direção de qual o caminho a tomar para o desenvolvimento do MVP. Foram levantadas seis questões genéricas para obter uma visão geral esperada pelos usuários em potencial. Dentre as questões, quatro são objetivas e as demais subjetivas, conforme a Figuras 10 e 11.

O formulário poderia ser encontrado até o dia 30 de novembro no link <https://forms.gle/qP1BWs4G9qGxRLz26>. Sua divulgação tem como foco o mês de novembro de 2021, entre os dias 06 e 30 e com base nele serão extraídas as histórias de usuários, dando continuidade no projeto.

Figura 10 - Formulário para levantamento de requisitos



Arrecadação de Vestimentas

Como tema do TCC de Engenharia de Software da Unlevangélica, foi proposto o desenvolvimento de um software para arrecadação de vestimentas. Para esse fim, contamos com a ajuda de todos para que possamos colher informações necessárias para o desenvolvimento de uma ferramenta que no futuro possa ser útil para todos. Sua opinião é muito importante para nós.

Você já realizou alguma doação de vestimenta? *

Sim

Não

Já organizou e/ou organiza alguma ação solidária? *

Sim

Não

Acredita que seria mais fácil realizar doações online? *

Sim

Não

Talvez

Fonte: Autores

Figura 11 - Continuação do formulário para levantamento de requisitos

Gostaria de possuir um mapa exibindo locais de arrecadação próximo de você? *

Sim

Não

Talvez

Como você imagina um site para distribuição de vestimentas? *

Texto de resposta longa

O que te chamaria a atenção em um site e/ou app para realizar doações? *

Texto de resposta longa

Fonte: Autores

7. RESULTADOS

Com o decorrer do desenvolvimento do projeto, alguns resultados esperados foram alcançados. O desenvolvimento de toda estrutura organizacional para iniciar o desenvolvimento do MVP, bem como o sucesso do questionário para validar se seria viável desenvolver um site que suprisse a necessidade das pessoas em sua maioria.

Os dados coletados na pesquisa de campo através do questionário, mostrou que dos 47 participantes do formulário foi retratado a possibilidade de desenvolver um MVP mostrando um interesse de utilizar o site proposto. 38,3% das pessoas acreditam que com um sistema online irá ajudar e facilitar que as pessoas doem vestimentas, já 76,6% acham que um mapa que exiba os locais de arrecadação próximos, aumentaria ainda mais.

Obteve-se também um feedback, ao qual 48,9% das pessoas ficaram indecisas e acabaram escolhendo a opção talvez, nos dando uma margem de uma futura aceitação. Com essa quantidade elevada, existe a possibilidade de converter isso em satisfação. Para isso, deve-se procurar focar na experiência de usuário, em uma interface chamativa e simples.

Com o grande avanço na estruturação da equipe e com o desenvolvimento pessoal de cada integrante, foi definido e iniciado o desenvolvimento do MVP, sendo ele o objetivo do próximo desenvolvimento do projeto.

REFERÊNCIAS

ANÁPOLIS, Prefeitura de. **Ala 2 – Base Aérea de Anápolis realiza a entrega de quase 500 itens arrecadados na Campanha do Agasalho**, 2021. Disponível em <https://www.anapolis.go.gov.br/ala-2-base-aerea-de-anapolis-realiza-a-entrega-de-quase-500-itens-arrecadados-na-campanha-do-agasalho/>. Acesso em: 06 set. 2021.

ANDRADE, Thiago Faria de. **Programação: Back-end vs Front-end vs Fullstack: Escolha o seu futuro como programador!** 2018. Disponível em: [https://blog.algaworks.com/back-end\[1\]front-end-full-stack/](https://blog.algaworks.com/back-end[1]front-end-full-stack/). Acesso em: 01 nov. 2021.

AWS, Amazon. **Microserviços**, 2021. Disponível em <https://aws.amazon.com/pt/microservices/> Acesso em: 06 nov. 2021

BALLERINI , Rafaella. **HTML, CSS e Javascript, quais as diferenças?**, 25 fev. 2021. Disponível em: <https://www.alura.com.br/artigos/html-css-e-js-definicoes>. Acesso em: 21 out. 2021.

COULOURIS, George .; DOLLIMORE, Jean .; KINDBERG, Tim .; BLAIR, Gordon. **Sistemas Distribuídos**, 2013. 9788582600542. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788582600542/>. Acesso em: 23 out. 2021.

DEVMEDIA. **Guia de PostgreSQL**, c2021. Disponível em: <https://www.devmedia.com.br/guia/guia-de-postgresql/34328>. Acesso em: 06 nov 2021.

EIS, Diego; FERREIRA, Elcio. **HTML5 e CSS3: com farinha e pimenta**. São Paulo: Clube de Autores, 2012. P. 219

ELIXIR. **Elixir is a dynamic, functional language for building scalable and maintainable applications**, 2021. Disponível em: <https://elixir-lang.org/>. Acesso em: 14 out. 2021.

GOIÂNIA, Prefeitura de. **Aquecendo Corações é exemplo para demais municípios**, 2021. Disponível em <https://www.goiania.go.gov.br/aquecendo-coracoes-e-exemplo-para-demais-municipios/>. Acesso em: 06 set. 2021

HEFLO. **Não confunda mais: Agile, Scrum e Kanban**, 2017. Disponível em: <https://www.heflo.com/pt-br/agil/agile-scrum-e-kanban/>. Acesso em: 21 out. 2021.

KANER, C. **The ongoing revolution in software testing**, 2004. Disponível em: <http://www.kaner.com/pdfs/TheOngoingRevolution.pdf>>. Acesso em: 05 nov. 2021.

MARTIN, Robert. C. **Desenvolvimento Ágil Limpo**. Editora Alta Books, 2020. 9788550816890. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788550816890/>. Acesso em: 05 nov. 2021.

MICROSOFT. **Comunicação front-end de cliente**, 2021. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/architecture/cloud-native/front-end-communication>. Acesso em: 15 set. 2021.

MILETTO, Evandro. M .; BERTAGNOLLI, Silvia.de. C. **Desenvolvimento de Software II** . Grupo A, 2014. 9788582601969. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788582601969/>. Acesso em: 01 nov. 2021.

PARANÁ, G. do Estado do. **Paraná Solidário já cadastrou 460 doadores e 240 entidades, 2019**. Disponível em <https://www.aen.pr.gov.br/modules/noticias/article.php?storyid=104774>. Acesso em: 06 set. 2021.

POSTGRESQL. **The World's Most Advanced Open Source Relational Database**, 2021. Disponível em: <https://www.postgresql.org/>. Acesso em: 06 nov. 2021.

PRIKLADNICKI, Rafael.; WILLI, Renato.; MILANI, Fabiano. **Métodos Ágeis para Desenvolvimento de Software**. Grupo A, 2014. 9788582602089. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788582602089/>. Acesso em: 04 nov. 2021.

REACT. **Virtual DOM and Internals**, 2021. Disponível em <https://reactjs.org/docs/faq-internals.html#gatsby-focus-wrapper>. Acesso em: 06 nov. 2021

RIES , Eric. **A Startup Enxuta: Como os empreendedores atuais utilizam inovação contínua para criar empresas extremamente bem-sucedidas**. São Paulo: Leya, 2011.

RIES, Eric - Caroli. **Minimum Viable Product**, 2009. Disponível em <https://www.caroli.org/en/eric-ries-on-mvp-2009/>. Acesso em: 06 set. 2021.

SERASA EMPREENDEDOR. **O que você precisa saber sobre MVP**, 2018. Disponível em: <https://empresas.serasaexperian.com.br/blog/o-que-e-mvp-ou-melhor-produto-minimo-viavel-e-como-utiliza-lo/>. Acesso em: 15 set. 2021.

SOUZA, Ivan de - Rockcontent. **PostgreSQL: saiba o que é, para que serve e como instalar**, 2020. Disponível em <https://rockcontent.com/br/blog/postgresql/>. Acesso em: 06 nov. 2021

VIANA, Daniel. **O que é front-end e back-end?** 2017. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-front-end-e-back-end/>. Acesso em: 01 nov. 2021.

WINBLAD, Kjell - Elang. **Practical functional programming for a parallel world**, 2021. Disponível em <https://www.erlang.org/>. Acesso em: 06 nov. 2021