

UNIVERSIDADE EVANGÉLICA DE GOIÁS - UNIEVANGÉLICA
ENGENHARIA DE SOFTWARE

GABRIEL SOUZA DE CASTRO E ROSA
JOSÉ INÁCIO FRANÇA FILHO

Gestão Devocional: REZZO

Anápolis, Goiás
Setembro, 2021

UNIVERSIDADE EVANGÉLICA DE GOIÁS - UNIEVANGÉLICA
ENGENHARIA DE SOFTWARE

GABRIEL SOUZA DE CASTRO E ROSA
JOSÉ INÁCIO FRANÇA FILHO

Gestão Devocional: REZZO

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. Ms. Alexandre Moraes Tannus.

Anápolis, Goiás
Setembro, 2021

UNIVERSIDADE EVANGÉLICA DE GOIÁS - UNIEVANGÉLICA
ENGENHARIA DE SOFTWARE

GABRIEL SOUZA DE CASTRO E ROSA
JOSÉ INÁCIO FRANÇA FILHO

Gestão Devocional: REZZO

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:

Prof. Ms. Alexandre Moraes Tannus.
(ORIENTADOR)

Nome completo do examinador, [Especialista | Mestre | Doutor], Sigla da universidade que atua
(AVALIADOR)

Anápolis, dia de mês de 2021.

FICHA CATALOGRÁFICA

FILHO, José Inácio. **Gestão Devocional: REZZO**. 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. Devoção católica. Dificuldade contemporânea. Gerenciador de rotina.

ROSA, Gabriel. **Gestão Devocional: REZZO**. 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. Devoção católica. Dificuldade contemporânea. Gerenciador de rotina.

REFERÊNCIA BIBLIOGRÁFICA

FILHO, José, ROSA, Gabriel. **Gestão Devocional: REZZO**. Anápolis, 2021. 41p. Monografia - Curso de Engenharia de Software Universidade Evangélica de Goiás - UniEVANGÉLICA.

CESSÃO DE DIREITOS

NOMES DOS AUTORES: José Inácio França Filho, Gabriel Souza de Castro e Rosa.

TÍTULO DO TRABALHO: Gestão Devocional: REZZO.

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.

José Inácio França Filho

Gabriel Souza de Castro e Rosa

Anápolis, dia de mês de 2021

AGRADECIMENTOS

Agradecemos primeiro a Deus por ter nos mantido na trilha certa durante este projeto com saúde e forças para chegar até o final, com a grande intercessão do patrono do presente projeto: o glorioso pai adotivo de Cristo, São José. Sem as graças necessárias, jamais teríamos conseguido chegar até aqui.

Agradecemos às nossas famílias pelo apoio que sempre nos deram durante toda a nossa vida.

Agradecemos aos amigos, companheiros de trabalhos e irmãos na amizade que fizeram parte de nossa vida e formação.

Deixamos um agradecimento especial ao nosso orientador pelo incentivo e pela dedicação do seu escasso tempo ao nosso projeto.

Também, agradecemos à Universidade Evangélica de Goiás e a todos os professores do nosso curso pela elevada qualidade do ensino oferecido.

RESUMO

A realização de práticas devocionais surge como proposta de aproximar o homem de si e de Deus, que por sua própria natureza decorre a capacidade de conhecer a verdade e de agir moralmente segundo o bem. O advento da revolução tecnológica com o seu ritmo acelerado, prevê a escassez de tempo e uma percepção de vida rasa com sua ordem para o homem do novo tempo que, embora tenha o acesso quase ilimitado a informações, esse fato deixa-o com a mente caótica, poluída, impaciente e com menos capacidade para pensamentos aprofundados. O desenvolvimento de um gerenciador que organize a rotina para esses fiéis, e pode ajudar em suas práticas espirituais no enfrentamento das dificuldades cotidianas coordenando suas devoções. Esse sistema será implementado utilizando JavaScript (ReactNative e Node.Js) por sua flexibilidade e alta capacidade de escala, integrado à elaboração de uma arquitetura de informações instigando a busca por Deus neste tempo tão caótico da indústria 4.0. O resultado desse trabalho pretende intensificar a rotina de orações do fiel católico gerenciando-a, e ajudando no enfrentamento das dificuldades atuais, para que possa garantir uma melhor percepção de desenvolvimento de sua espiritualidade cristã.

Palavras-chave: Devoção católica. Dificuldade contemporânea. Gerenciador de rotina. JavaScript.

ABSTRACT

The performance of devotional practices appears as a proposal to bring man closer to himself and to God, which, by its very nature, results from the ability to know the truth and act morally according to what is good. The advent of the technological revolution with its fast pace, predicts the scarcity of time and a shallow perception of life with its order for the man of the new time who, although he has almost unlimited access to information, this fact leaves him with the mind chaotic, polluted, impatient and less capable of deep thinking. The development of a manager who organizes the routine for these faithful, and can help in their spiritual practices in facing daily difficulties by coordinating their devotions. This system will be implemented using JavaScript (ReactNative and Node.Js) for its flexibility and high scalability, integrated to the elaboration of an information architecture instigating the search for God in this very chaotic time of industry 4.0. The result of this work intends to intensify the prayer routine of the Catholic faithful, managing it, and helping to face the current difficulties, so that they can guarantee a better perception of the development of their Christian spirituality.

Key words: Catholic devotion. Contemporary difficulty. Routine Manager. JavaScript.

LISTA DE ILUSTRAÇÕES

Figura 1. Plano de Ação 5W2H.....	19
Figura 2. Modelo Scrum	20
Figura 3. Modelo KanBan.....	21
Figura 4. Ciclo de Vida OpenUp	21
Figura 5. Dado da Pesquisa. 01.....	25
Figura 6. Dado da Pesquisa. 02.....	25
Figura 7. Dado da Pesquisa. 03.....	26
Figura 8. BPMN do REZZO.....	27
Figura 9. BPMN do MVP do REZZO.....	28
Figura 10. Caso de Uso do REZZO.....	28
Figura 11. Caso de Uso do MVP do REZZO.....	29
Figura 12. - Paleta de Cores da Identidade.....	30
Figura 13. Imagem Devocional da Identidade	31
Figura 14. Tela 1 de Carregamento da Identidade.....	31
Figura 15. Tela 2 de Carregamento da Identidade.....	32
Figura 16. Tela para Leituras e Orações da Identidade	32
Figura 17. Base de Navegação da Identidade	33
Figura 18. Splash Screen do REZZO.	33
Figura 19. Escolha de Interface do REZZO.....	34
Figura 20. Confirmação de Interface do REZZO	34
Figura 21. Preload de Interface Definida do REZZO.....	35
Figura 22. Tela da Agenda.....	35
Figura 23. Tela Estendida da Agenda	36
Figura 24. Tela com Status da Agenda	36
Figura 25. Código Retorno da lista de orações	37
Figura 26. Código armazenamento de dados	38
Figura 27. Código armazenamento na agenda	38
Figura 28. Código Instauração de dados.....	39
Figura 29. Código transferência para oração	39
Figura 30. Agenda sem atividades.....	40
Figura 31. Tela para adicionar uma oração na agenda.....	40
Figura 32. Agenda com atividades não concluídas.....	41
Figura 33. Agenda com atividades concluídas	41

LISTA DE TABELAS

Marcos do OpenUP	23
Requisito Funcional 001	29
Requisito Funcional 002	29
Requisito não Funcional 001	30
Requisito não Funcional 002.....	30

SUMÁRIO

1. INTRODUÇÃO	15
2. OBJETIVOS	16
2.1. Objetivo Geral.....	16
2.2. Objetivos Específicos	16
3. JUSTIFICATIVA	17
4. FUNDAMENTAÇÃO TEÓRICA	18
4.1. Catolicismo, o fiel e a oração	18
4.2. 5W2H.....	18
4.3. Scrum e Kanban.....	19
4.4. OpenUp	21
4.5. Artefatos Digitais	22
5. METODOLOGIA DA PESQUISA	23
6. RESULTADOS	25
6.1. Pesquisa inicial.....	25
6.2. Product Overview	26
6.3. Plano de Ação - 5W2H.....	26
6.4. BPMN.	27
6.5. Caso de Uso.....	28
6.6. Requisitos Funcionais	29
6.7. Requisitos Não Funcionais	30
6.8. Protótipos.....	30
6.9. Codificação	36
7. CONCLUSÃO E RESULTADOS ESPERADOS	42
8. REFERÊNCIAS	43

1. INTRODUÇÃO

Na atualidade de quem vive conectado 24 horas por dia e têm a possibilidade de em tempo real acompanhar tudo que acontece em todo o mundo. A tecnologia e a inovação proporcionam evolução e revolução. As informações se movem instantaneamente e os conecta diretamente com as pessoas, sem qualquer limite de tempo e espaço. Passam pela quarta revolução industrial: a indústria 4.0. (FERREIRA, 2017).

A origem contemporânea com o advento de toda a revolução tecnológica traz ao indivíduo uma escassez de tempo e uma percepção de vida rasa com sua ordem, na sua necessidade de estar cada vez mais rápido, acaba se perdendo. Segundo Virilio (1999), o progresso tecno científico faz surgir um novo tempo com um ritmo acelerado ao máximo - e perde-se o tempo.

Não há perseverança, no livro *The shallows: what the internet is doing to our brains* (CARR, 2010), constata que, apesar desse acesso quase ilimitado a informações na grande rede, o homem perde a cada dia a capacidade de focar em apenas um assunto. A mente fica caótica, poluída, impaciente e, para o autor, com menos capacidade para pensamentos aprofundados. Um verdadeiro caos para a vida.

Tantos atritos remancham a vida das pessoas na modernidade, incluindo o nicho social cristão, na realização e perseverança de sua marcha espiritual. Então, será viável um *software* que gerencie a rotina para elas, e os ajude em suas práticas espirituais?

2. OBJETIVOS:

2.1. Objetivo Geral:

Desenvolver um *software* que gerencie a rotina de fiéis católicos dentro de suas práticas, e os ajude no enfrentamento das dificuldades atuais coordenando suas devoções.

2.2. Objetivos Específicos:

Realizar uma pesquisa para a compreensão das necessidades dos fiéis católicos, e a proposta como solução.

Construir conceitos de Gamificação na interface gráfica;

Desenvolver uma Experiência de Usuário cujo resultado auxilie a compreensão das práticas no contexto espiritual;

Desenvolver um *script* para tratar dados coletados;

Construção de *States* para armazenar dados e definir agendas no *software*.

3. JUSTIFICATIVA

O presente projeto de tecnologia tem por motivação o auxílio na caminhada espiritual dos fiéis católicos contemporâneos.

Este hiato motivou o projeto, sendo importante ressaltar o vínculo pessoal que os autores possuem com o tema, através de suas vivências com as dificuldades ressaltadas.

Levando em conta a atual fragilidade do cenário de vida advindas da indústria 4.0., enfatiza-se o desenvolvimento de soluções vivificas para o fiel, contribuindo desta maneira, com o desenvolvimento íntimo entre Deus e as pessoas.

4. FUNDAMENTAÇÃO TEÓRICA

Essa seção abordará os temas essenciais para o desenvolvimento do REZZO, tais como: habilidades, artefatos digitais e *frameworks* para o *software* e projeto.

4.1 Catolicismo, o fiel e a oração

O homem, de fato, é um ser social carregado de instintos, porém o homem é bem mais. O homem é filho de Deus, imagem e semelhança de seu Criador: “façamos o homem à nossa imagem e semelhança”, segundo Gênesis 1, 26 (BÍBLIA, 2018). A qual decorre a capacidade de conhecer a verdade e de agir moralmente segundo o bem de conhecer e amar, perceber, o intelecto e a razão. Entende-se então que, para compreender o ser humano, antes é necessário conhecer as pessoas Divinas. É olhando para Deus que entendemos o valor que cada ser humano possui (CATECISMO DA IGREJA CATÓLICA, 1993), não se entende a pessoa sem antes entender sua origem. Na categoria de criação, o que dá vida ao homem é o princípio de sua alma que é racional (ARISTÓTELES, 2011), o que a caracteriza é a racionalidade, a inteligência, o pensamento e a vontade - vendo que ela é espiritual. Assim como é necessária a água para as plantas viverem e não secarem, assim também, para nós é necessário um mantimento para viver com plenitude; assim como a alma dá vida ao corpo, precisamos dar vida à alma, e a nutrimos através da oração (CRISÓSTOMO, 2010).

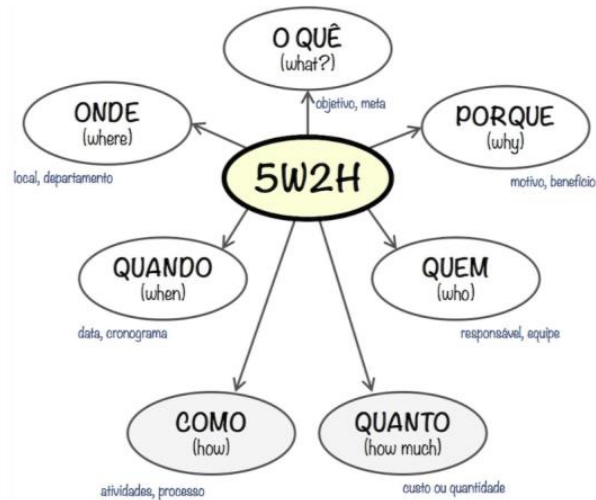
O catolicismo nasceu na era de Cristo quando o mesmo criou a Igreja, segundo Mateus 16, 18 (BÍBLIA, 2018). A partir deste princípio, é comum viver um relacionamento com o próprio Deus dentro da religião, pois ela, O une com o homem, e são religados através da necessidade espiritual do homem, o valor da oração, como o próprio Jesus pediu diversas vezes, a exemplo tomemos Mateus 26, 41 (BÍBLIA, 2018): “Vigiai e orai para que não entreis em tentação. O espírito está pronto, mas a carne é fraca”.

As orações são práticas de extrema importância na vida do fiel, elas devem ser tão elaboradas que se harmonizem com a sagrada liturgia, pois, de fato, por sua própria natureza aproximam o homem de si, da Verdade, de Deus (TRESE, 2014). Logo, nunca saberá viver bem quem não sabe rezar bem: não sabe viver corretamente quem não sabe rezar de forma correta. E a mesma coisa dizia São Francisco de Assis, ao afirmar que de nenhuma alma se pode esperar bons frutos sem a oração (CELANO, 2018).

4.2 5W2H

O plano de ação do projeto REZZO será feito com a utilização de uma ferramenta chamada 5W2H que é um *checklist* administrativo de atividades, prazos e responsabilidades que devem ser desenvolvidas com clareza e eficiência por todos os envolvidos no projeto. Tem como função definir o que será feito, porque, onde, quem irá fazer, quando será feito, como e quanto custará (LONGUINHO, 2017).

Figura 1 - Plano de Ação 5W2H



Fonte: Google Imagens, 2021.

Se fixa ao que será feito, objetivos e metas; Por que será feito, motivos e benefícios; onde será feito, local e departamento; Quando, datas e cronogramas; por quem será feito, responsáveis e equipe; como será feito, atividades, processos, desenvolvimento; por fim, quanto irá custar.

Pela simplicidade e eficiência de sua metodologia, haverá um controle maior das ações estratégicas nas quais se pretende realizar.

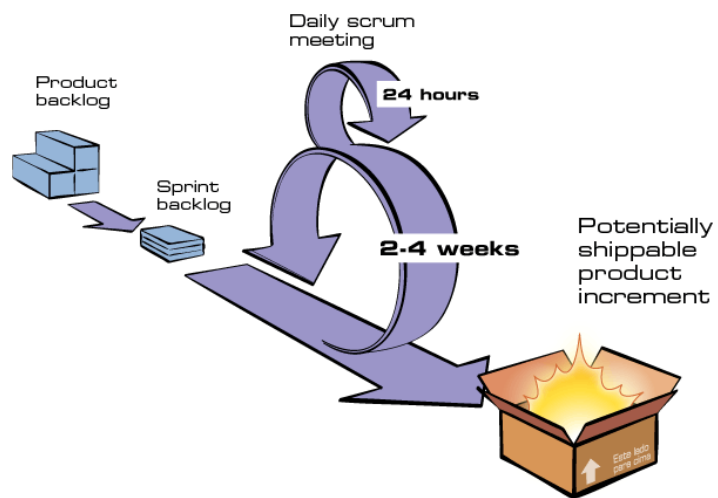
4.3 Scrum e Kanban

Na gestão do projeto, se utilizará o *framework* de processo ágil Scrum, uma metodologia usada para a gestão dinâmica de projetos que permite controlar de forma eficaz e eficiente o trabalho, potencializando as equipes que trabalham em prol de um objetivo em comum (SCHWABER e SUTHERLAND, 2017). No Scrum, os projetos são divididos em ciclos tipicamente mensais chamados de Sprints. O Sprint representa um *Time Box* dentro do qual uma parcela de atividades deve ser executada. Metodologias ágeis de desenvolvimento de software são iterativas, ou seja, o trabalho é dividido em iterações, que são chamadas de Sprints no caso do Scrum.

As funcionalidades a serem implementadas em um projeto são mantidas em uma lista que é conhecida como *Product Backlog*. No início de cada Sprint, faz-se um Sprint Planning Meeting, uma reunião de planejamento na qual o Product Owner prioriza os itens do Product Backlog e a equipe seleciona as atividades que ela será capaz de implementar durante o Sprint que se inicia. As tarefas alocadas em um Sprint são transferidas do Product Backlog para o Sprint Backlog.

A cada dia de uma Sprint, a equipe faz uma breve reunião (normalmente de manhã), chamada Daily Scrum. O objetivo é explanar o conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que se inicia.

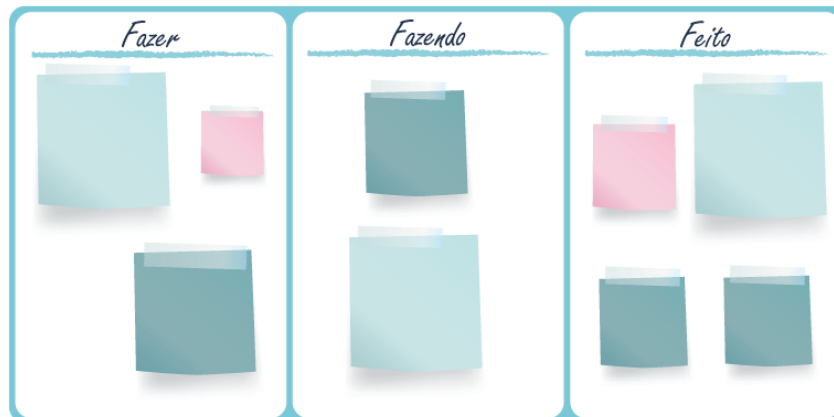
Ao final de um Sprint, a equipe apresenta as funcionalidades implementadas em uma Sprint Review Meeting. Finalmente, faz-se uma Sprint Retrospective e a equipe parte para o planejamento do próximo Sprint (THE DEFINITIVE GUIDE TO SCRUM, 2017). Assim reinicia-se o ciclo. Como no exemplo abaixo:

Figura 2 – Modelo Scrum

Fonte: Google Imagens, 2021.

Com o auxílio do modelo *Kanban*, uma estratégia para otimizar o fluxo de valor para partes interessadas através de um processo que utiliza um sistema visual que limita a quantidade de trabalho em andamento. Tornando mais eficaz o desenvolvimento como um todo (ANDERSON e ANDY CARMICHAEL, 2016). Ele funciona basicamente em três colunas “A Fazer”, “Em Execução” e “Feito” (mas nada impede que outras colunas sejam adicionadas). Cada coluna possui uma série de cartões que representam as tarefas que precisam ser executadas e a equipe vai “puxando” as tarefas conforme o fluxo de trabalho, por isso o nome produção: cada cartão é uma entrega realizada.

Figura 3 – Modelo Kanban



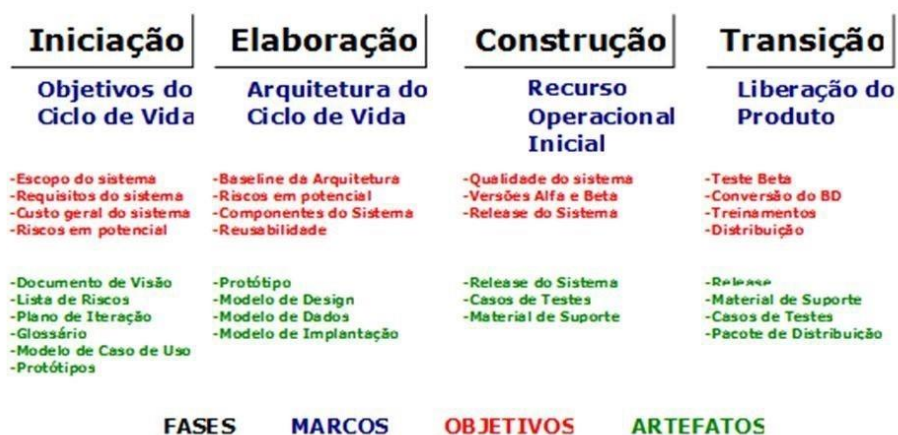
Fonte: Google Imagens, 2021.

4.4 OpenUp

Para o *Software*, o uso da metodologia *OpenUp*, uma metodologia livre que cobre um conjunto diverso de perspectivas e necessidades de desenvolvimento de software. Seus processos são estruturados de forma a oferecer apenas seus elementos mais necessários; oferece todos os recursos necessários para um processo de software e se adapta de acordo com as reais necessidades da equipe e/ou, com as características inerentes de cada projeto (KROLL e MACISAAC, 2006).

Figura 4 – Ciclo de Vida OpenUp

Marcos do Open Up



Fonte: Google Imagens, 2021.

O OpenUP estrutura o ciclo de vida do projeto em **quatro fases**:

- **Iniciação** - onde os *stakeholders* e os membros da equipe colaboram para determinar o escopo e os objetivos do projeto, e determinar se o projeto deve ou não continuar.
- **Elaboração** - quando riscos arquiteturais significantes são tratados.
- **Construção** - foca no detalhamento dos requisitos, no desenho, na implementação e nos testes da maior parte do software.
- **Transição** - focada na transição do software para o ambiente do cliente e na obtenção da concordância dos *stakeholders* de que o desenvolvimento do produto está completo.

O ciclo de vida do projeto fornece aos *stakeholders* e à equipe, visibilidade e pontos de decisão durante o projeto. Isto permite uma efetiva supervisão para tomar decisões de “prosseguir ou parar” em momentos apropriados (SENE, 2010).

4.5 Artefatos Digitais

O *backend* do projeto será desenvolvido na linguagem *JavaScript* (JS) que é uma linguagem de programação interpretada estruturada, permitindo a implementação de diversos itens, como animações, mapas, gráficos ou informações que se atualizam em intervalos de tempo padrão, por exemplo (FLANAGAN, 2020), utilizando *Node.JS*, que é ambiente de execução *Javascript server-side* e o principal motivo de sua adoção é a sua alta capacidade de escala, sua arquitetura, flexibilidade e baixo custo, o tornam uma boa escolha para implementação (LENON, 2018).

Para o *front-end React Native*, um conjunto de bibliotecas utilizadas para criar uma base, um facilitador no desenvolvimento, possibilitando a criação de aplicações móveis multiplataformas (Android e iOS) utilizando apenas *Javascript* (REACT NATIVE), e suas dependências.

Para banco de dados utilizaremos *PostgreSQL*, sistema gerenciador de banco de dados objeto relacional (SGBD) que oferece todas as suas funcionalidades de forma gratuita, tendo também uma funcionalidade mais amigável e consistente (POSTGRESQL).

5. METODOLOGIA DA PESQUISA

Para entender profundamente as dificuldades experimentadas pelos fiéis católicos será feita uma pesquisa inicial, no empenho de melhor ajuda-los em suas necessidades, e entender suas experiências diárias com Deus. Chegar à compreensão do problema tecno-social e a proposta do *software* como solução. Feita e analisada através da ferramenta *Google-Forms*.

O presente projeto será feito com o plano de Ação 5W2H (LONGUINHO, 2016) em que serão identificados os componentes de administração, prazos, objetivos e responsabilidades necessárias para garantir o desenvolvimento e finalização do REZZO, caracterizando, dessa forma, a estrutura gerencial do *software*.

Utilizará o *framework* de processo ágil *Scrum* (SCHWABER e SUTHERLAND, 2017) na gestão do projeto para o gerenciamento. Controlando os processos, reduzindo as falhas, auxiliando no planejamento e por ser bastante adaptável. Terá o auxílio do modelo *Kanban* (ANDERSON e ANDY CARMICHAEL, 2016), para tornar mais eficaz o desenvolvimento do Projeto com a organização cotidiana mantendo o foco na equipe.

Na gestão do *Software*, terá o uso da metodologia *OpenUp* (KROLL e MACISAAC, 2006), nos baseando para os cronogramas e *releases* previstas no próprio modelo, trazendo as responsabilidades dos *Stakeholders* e qual a melhor forma deles trabalharem com a equipe de desenvolvimento para obter um *software* que atenda às suas necessidades. Seguindo a **Tabela 1** com o presente cronograma para o desenvolvimento:

	TCC1					TCC2						
ATIVIDADES	J A N	F E V	M A R	A B R	M A I	J U N	J U L	A G O	S E T	O T O	N O V	D E Z
Objetivo do Ciclo de Vida	X	X	X	X	X	X	X	X				
Arquitetura do Ciclo de Vida			X	X	X	X	X	X	X	X		
Recursos Operacional Inicial					X	X	X	X	X	X	X	X
Revisão e Correção do Documento de TCC			X		X	X		X			X	X

Tabela 1 – Marcos do OpenUp

Objetivo do Ciclo de Vida:

Desenvolvimento do Escopo do Sistema; dos Requisitos F. e N.F.; dos Riscos; o Modelo de Caso de Uso; o início da Prototipação com os conceitos de Gamificação; toda a parte inicial de documentação e a realização de uma pesquisa para compreender as necessidades dos fiéis.

Arquitetura do Ciclo de Vida:

Desenvolvimento de todos os Protótipos; os Modelos de UI e UX; o Modelo E.R. do Banco de Dados; a Revisão da Documentação do Projeto e do TCC.

Recurso Operacional Inicial:

Desenvolvimento da Versão MVP do *software*, que contará com as funcionalidades de Selecionar Interface, Definir Agenda, Escolher Orações e Personalizar Configurações; construção dos *States* que armazenarão os dados, e os *Scripts* que irão tratar os mesmos; construção do Banco de Dados.

Liberação do Produto:

Finalização do desenvolvimento da aplicação MVP.

O *Software* será desenvolvido na linguagem *JavaScript* (FLANAGAN, 2020), utilizando *Node.js* (LENON) no *backend*, também na construção de *States* para armazenar dados e definir agendas no *Software*. No *front-end* utilizará o *React Native* (REACT NATIVE) e outras dependências da própria biblioteca para a aplicação com os conceitos de gamificação na interface gráfica (a identidade visual do *software*) para engajar as pessoas, resolver os problemas e melhorar o aprendizado, motivando as ações e comportamentos no *software*. Para o banco de dados utilizará *PostgreSQL* (POSTGRESQL).

Por fim, será feita uma pesquisa, para entendermos os resultados que foram propostos à compreensão das práticas espirituais. As informações obtidas durante o processo de embasamento teórico serão suficientes para que a criação do *software* reflita a sua necessidade básica sobre o tema em discussão. A comparação à eficiência do *software* no processo diário do fiel católico deverá também ser testada, contudo, não serão realizados nesse trabalho, propondo-se que seja feita em um trabalho futuro.

6. RESULTADOS

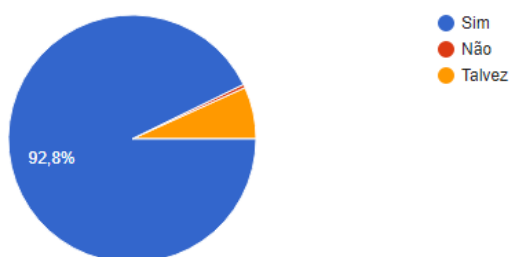
6.1. Pesquisa Inicial

A pesquisa foi realizada entre o mês de março e maio com 221 (duzentos e vinte e um) pessoas, através do aplicativo de gerenciamento de pesquisas *Google Forms*. Com o objetivo de profundamente entender as dificuldades que os fiéis católicos experimentam, no empenho de melhor ajuda-los em suas necessidades, entender suas experiências diárias com Deus e descobrir o que de fato necessitam, chegando a uma melhor compreensão do problema.

Foram feitas as seguintes perguntas com a intenção de validar os primeiros requisitos que haviam sido planejados, vendo se, os *stakeholders* realmente necessitam de tais funcionalidades. As **Figuras 5, 6 e 7** apresentam os resultados da pesquisa.

Figura 5 – Dado da Pesquisa.01

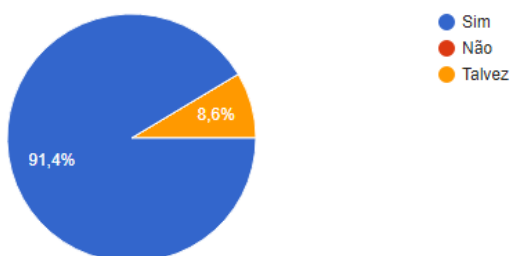
Um cronograma semanal com orações específicas, com calendário e alarme para te lembrar.
Te ajudaria na vida espiritual?
221 respostas



Fonte: Autor, 2021.

Figura 6 – Dado da Pesquisa.02

Você acha um sistema de auxílio à confissão necessário? (Com exame de consciência, data da última confissão, ato de contrição e etc)
221 respostas

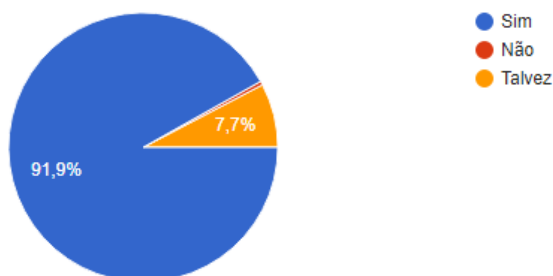


Fonte: Autor, 2021.

Figura 7 – Dado da Pesquisa.03

Além de orações óbvias como terços, novenas, ladainhas, orações aos santos. Um devocional reto para orações específicas, seria bom?

221 respostas



Fonte: Autor, 2021.

Foram coletados também respostas individuais ao que diz respeito sobre as dificuldades encontradas na vida de oração; tempo, concentração, perseverança, organização e ordem foram as variáveis mais apresentadas pelos *stakeholders*. Visando o desenvolvimento centrado no usuário, apresentamos um espaço aceitando sugestões para o sistema, totalizando 117 requisitos para futuras *releases* do REZZO.

6.2. Product Overview

Para pessoas que desejam progredir na vida espiritual, com acesso a orações e leituras a qualquer momento na palma da mão. O REZZO é um aplicativo mobile, que diferente de outros existentes, terá um planner específico para cada pessoa realizar as suas devoções no melhor horário e dia sem se perder; apoio para o sacramento da confissão; iteração com o devoto. Mecanismos para que o usuário possa evoluir e estar cada vez mais enraizado à fé em uma interface bonita, simples e efetiva. O projeto atuará como um meio não necessário, mas transformador para uma rotina entregue à Deus.

6.3. Plano de Ação - 5W2H

Definimos o que será feito, porque, onde, quem irá fazer, quando será feito, como e quanto custará.

O que será feito, objetivos e metas:

Será desenvolvido um aplicativo mobile cujo nome é REZZO, ele objetiva criar uma ponte entre o usuário e Deus, e que seja de fácil acesso para todos os usuários.

Porque será feito, motivos e benefícios:

Auxiliar a caminhada espiritual dos fiéis católicos contemporâneos. Com a atual fragilidade do cenário de vida, desenvolver soluções vivificas para o fiel, contribuindo na sua intimidade com Deus.

Onde será feito, local e departamento:

O projeto será desenvolvido na casa dos envolvidos, pela Universidade Evangélica de Goiás – UniEvangélica, no Curso de Engenharia de Software e nas disciplinas de TCC I/ TCC II.

Quando, datas e cronogramas?

A entrega do produto funcional está ESTIMADA para acontecer dia 15/11/2021.

Por quem será feito, responsáveis e equipe?

O projeto será desenvolvido por estudantes do 8º período de Engenharia de Software na Universidade Evangélica de Goiás – UniEvangélica.

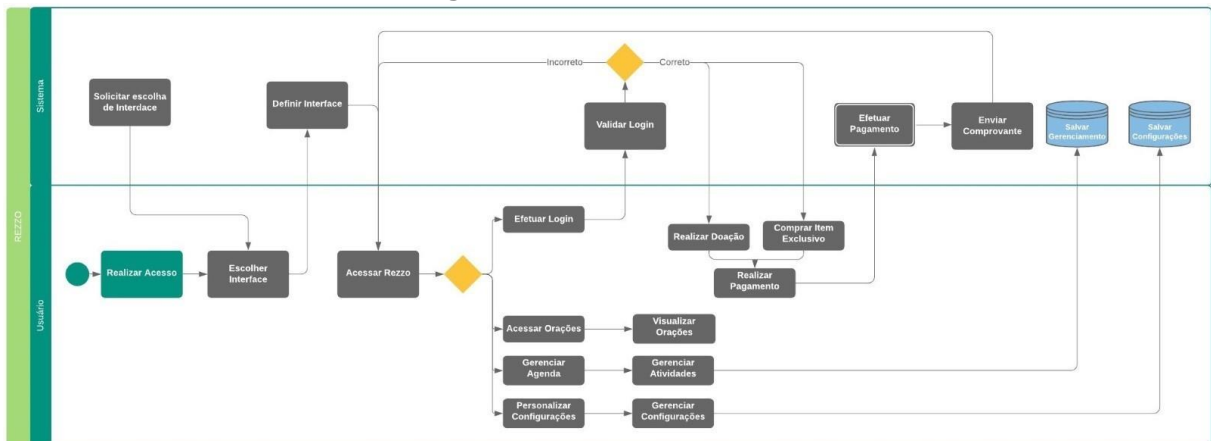
Como será feito, atividades, processo, desenvolvimento?

*Consultar Referencial Teórico e Metodologia do projeto.

6.4. BPMN

Na **Figura 8** é possível visualizar todas as operações do REZZO, trazendo maior facilidade na tomada de decisões. Através do início de evento, o usuário deverá escolher uma interface para que, de forma única, sua experiência seja mais íntima com o sistema. Com o acesso concluído, poderá personalizar suas configurações visuais, gerenciar sua agenda oracional, visualizar suas orações, e em eventual login, comprar itens exclusivos e realizar doações.

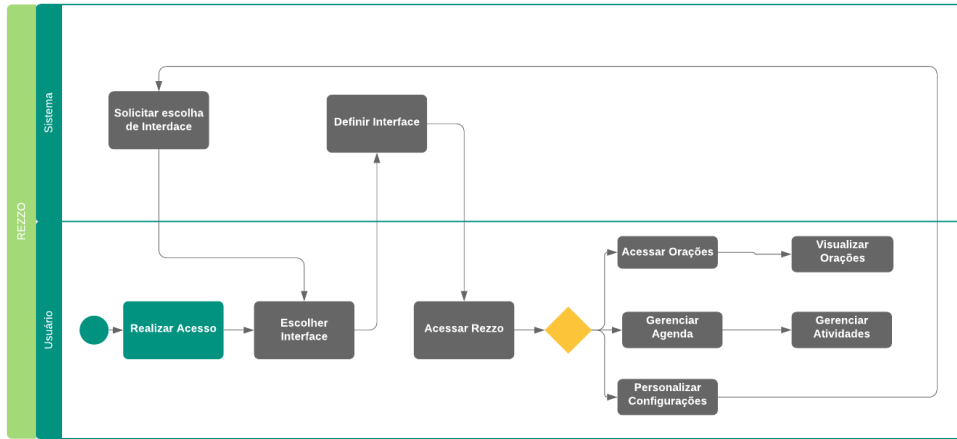
Figura 8 – BPMN do REZZO



Fonte: Autor, 2021.

Na **Figura 9** é possível visualizar todas as operações do MVP do REZZO. Através do início de evento, o usuário deverá escolher uma interface para que, de forma única, sua experiência seja mais íntima com o sistema. Com o acesso concluído, poderá personalizar suas configurações visuais, gerenciar sua agenda oracional e visualizar suas orações.

Figura 9 – BPMN do MVP do REZZO

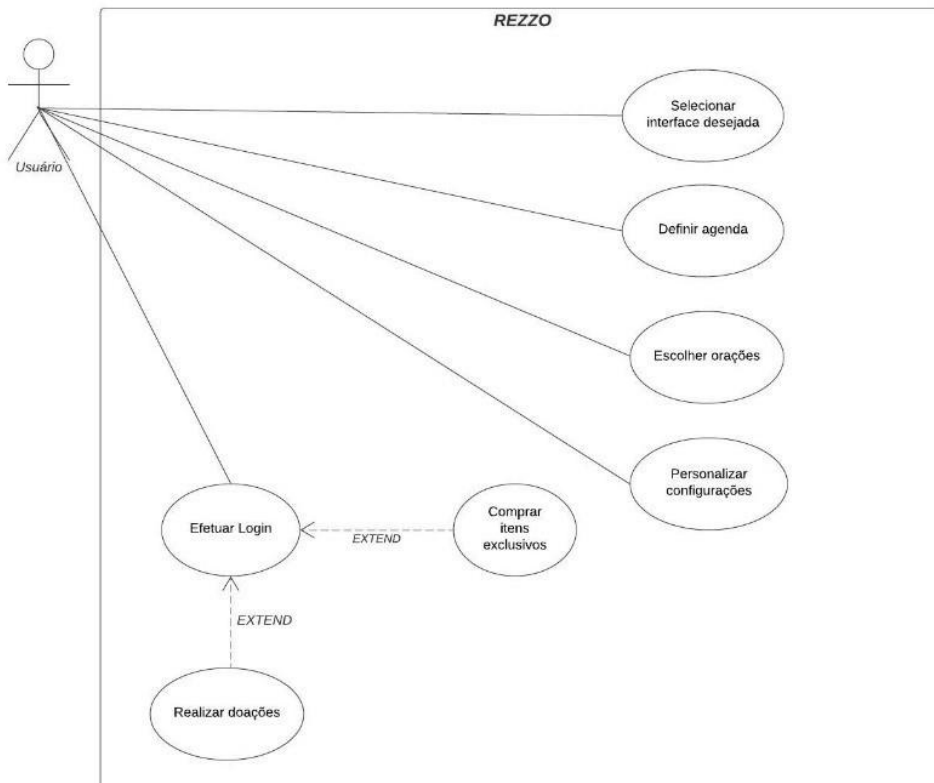


Fonte: Autor, 2021.

6.5. Caso de Uso

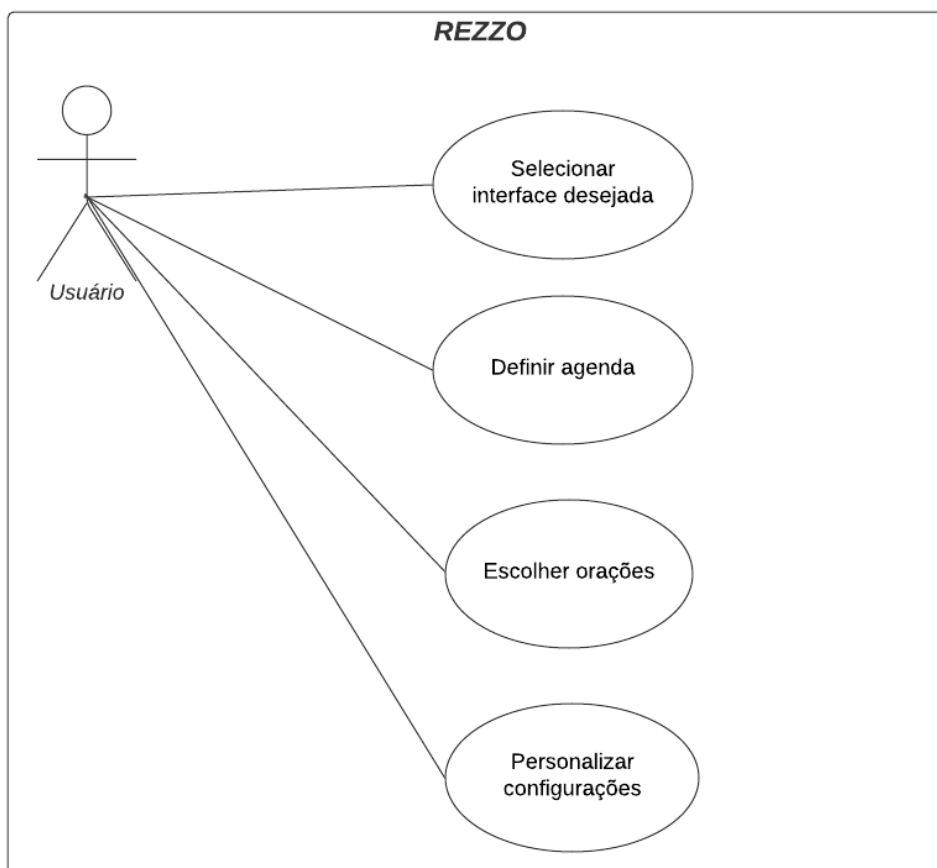
Na **Figura 10** é possível visualizar todas as interações do Usuário com o sistema REZZO, trazendo maior facilidade para especificar o contexto e os requisitos do sistema. Através do ator Usuário, pode-se ver o fluxo básico dos eventos que ele terá acesso.

Figura 10– Caso de Uso do REZZO



Fonte: Autor, 2021.

Na **Figura 11** é possível visualizar todas as interações do Usuário com o sistema REZZO em sua versão MVP, trazendo maior facilidade para especificar o contexto e os requisitos do sistema. Através do ator Usuário, pode-se ver o fluxo básico dos eventos que ele terá acesso nessa versão.

Figura 11– Caso de Uso do MVP do REZZO

Fonte: Autor, 2021.

6.6. Requisitos Funcionais

Nas **Tabelas 2 e 3** é possível visualizar dois exemplos da especificação de um requisito funcional do REZZO, trazendo atributos de qualidade para o projeto. Visando as seguintes métricas: código para rastreabilidade do requisito funcional, a nome do requisito funcional, descrição do requisito funcional através da ótica da equipe de desenvolvimento, o relacionamento do requisito funcional com o caso de uso e a prioridade do requisito funcional.

RF001	Definir Agenda
Descrição	O Sistema deve possibilitar o acesso a uma agenda, para o usuário definir as suas atividades.
Caso de Uso	UC001
Prioridade	Alta

Tabela 2 – Requisito Funcional 001

RF002	Escolher Oração
Descrição	O Sistema deve possibilitar orações (previamente fornecidas ao sistema) para a visualização do usuário.
Caso de Uso	UC002
Prioridade	Alta

Tabela 3 – Requisito Funcional 002

6.7. Requisitos Não Funcionais

Nas **Tabelas 4 e 5** é possível visualizar dois exemplos da especificação de um requisito não funcional do REZZO, trazendo atributos de qualidade para o projeto. Visando as seguintes métricas: código para rastreabilidade do requisito não funcional, a característica do requisito não funcional, descrição do requisito não funcional através da ótica da equipe de desenvolvimento e o relacionamento do requisito não funcional com o caso de uso.

RNF001	Portabilidade
Descrição	O tempo de retorno das consultas (isto é, o intervalo de tempo entre qualquer consulta e seu resultado) não pode ser maior que 3 segundos.
Caso de Uso	Relacionado em todos

Tabela 4 – Requisito não Funcional 001

RNF002	Usabilidade
Descrição	A interface do sistema deverá ser intuitiva, agradável e objetiva, que permitirá sua utilização com toda sua potencialidade em um curto espaço de tempo, beneficiando todos os usuários.
Caso de Uso	Relacionado em todos

Tabela 5 – Requisito não Funcional 002

6.8. Protótipos

Para um design centrado no usuário, arquitetou-se identidades visuais diferentes para que, cada pessoa com sua devoção particular, possa deixar a experiência do aplicativo mais individual. Cada identidade é formada por sua paleta de cores (**Figura 12**), imagem devocional (**Figura 13**), duas telas para carregamentos (**Figuras 14 e 15**), tela para leituras e orações (**Figura 16**) e base para a navegação do usuário (**Figura 17**):

Figura 12- Paleta de Cores da Identidade



Fonte: Autor, 2021.

Figura 13 – Imagem Devocional da Identidade



Fonte: Autor, 2021.

Figura 14 - Tela 1 de Carregamento da Identidade



Fonte: Autor, 2021.

Figura 15 - Tela 2 de Carregamento da Identidade



Fonte: Autor, 2021.

Figura 16 - Tela para Leituras e Orações da Identidade



Fonte: Autor, 2021.

Figura 17 – Base de Navegação da Identidade



Fonte: Autor, 2021.

Nas Figuras 18, 19, 20 e 21 podemos ver o curso inicial, que só virá no primeiro uso do sistema, com o exemplo de São José, para realizar a seleção da identidade visual:

Figura 18 - Splash Screen do REZZO



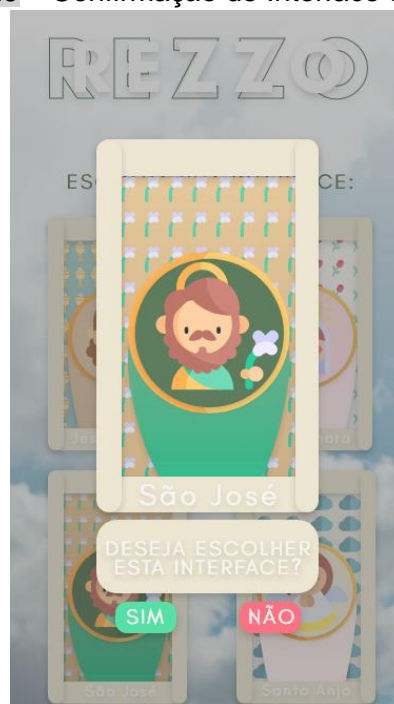
Fonte: Autor, 2021.

Figura 19 – Escolha de Interface do REZZO



Fonte: Autor, 2021.

Figura 20 – Confirmação de Interface do REZZO



Fonte: Autor, 2021.

Figura 21 – Preload de Interface Definida do REZZO



Fonte: Autor, 2021.

Inicia-se então o uso do sistema. Veremos o exemplo de um dos requisitos, o “Definir Agenda” nas Figuras 22, 23 e 24:

Figura 22 – Tela da Agenda



Fonte: Autor, 2021.

Figura 23 – Tela Estendida da Agenda



Fonte: Autor, 2021.

Figura 24 – Tela com Status da Agenda



Fonte: Autor, 2021.

6.9. Codificação

Para o desenvolvimento do aplicativo, o código foi construído usando o editor de texto *Visual Studio Code*. Conforme visto na metodologia, realizamos a codificação na linguagem *JavaScript*, permitindo que o sistema se torne mais dinâmico na troca mais rápida das telas

da interface e organização das orações na agenda, já que o processamento é todo local, não sendo então, necessário qualquer tipo de requisição ao servidor.

Para exemplo, dentro do que foi definido no MVP, utilizaremos os códigos das funções que toram a funcionalidade “Definir Agenda” possível (para ver todo o código, acesso no repositório: <https://github.com/gabrielcastr0/rezzo-app>).

Na Figura 25, vemos a função que é responsável por retornar a lista de orações, após fazer a requisição na API quando o usuário realizar o pedido para uma nova atividade em sua agenda.

Figura 25 - Código Retorno da lista de orações

```
const findActivities = async () => {
  setModalVisible(true);

  const req = await fetch('http://localhost:4000/api/prayers/', {
    method: 'GET',
    headers: {
      Accept: 'application/json',
      'Content-Type': 'application/json',
    },
  });

  const json = await req.json();

  if (json) {
    setPrayers(json.list);
    setFilterdData(json.list);
  }
};
```

Fonte: Autor, 2021.

Com o modal aberto apresentando então toda a lista de orações, após clicar em qualquer uma das orações presentes, a função (**Figura 26**) então armazena os dados de id, title, body e wasRead no state activity.

Figura 26 - Código armazenamento de dados

```

const handleClickActivity = item => () => {
  setActivity([
    ...activity,
    {
      id: item.id,
      title: item.title,
      body: item.body,
      wasRead: false,
    },
  ]);
};

```

Fonte: Autor, 2021.

Após realizar a seleção das orações desejadas, clicando em salvar, ativamos a função responsável por salvar os dados dentro da agenda do usuário (**Figura 27**).

Figura 27 - Código armazenamento na agenda

```

const savePrayerData = async () => {
  setModalVisible(false);
  const data = {
    date: `${todayDate}-${actualMonth}`,
    list: {
      prayers: activity,
    },
  };

  try {
    let res = await AsyncStorage.getItem('dataSave');
    let resObject = res ? JSON.parse(res) : [data];

    let found = resObject.find(element => element.date === data.date);
    if (found) {
      resObject.forEach((element, index) => {
        if (element.date === data.date) {
          resObject[index] = data;
        }
      });
    } else {
      resObject.push(data);
    }

    await AsyncStorage.setItem('dataSave', JSON.stringify(resObject));
    restoreData();
  } catch (err) {
    console.log(err);
  }
};

```

Fonte: Autor, 2021.

É necessário então que haja uma função responsável por instaurar os dados salvos no dia específico, que foi selecionado pelo usuário na agenda. Assim, após clicar em uma data, a função retorna todos as orações salvas naquele dia (**Figura 28**).

Figura 28 - Código Instauro de dados

```
const restoreData = async () => {
  let datas = await AsyncStorage.getItem('dataSave');
  let datasConverted = datas ? JSON.parse(datas) : [];
  let result = datasConverted.find(
    element => element.date === `${todayDate}-${actualMonth}`,
  );

  let prayers = result?.list?.prayers || [];

  setActivity(prayers);
};
```

Fonte: Autor, 2021.

Para acessar a oração salva na agenda, há uma função responsável por levar o usuário para a tela de leitura da oração, além, de marcar a oração como feita, mudando o ícone de *check* na tela principal da agenda (**Figura 29**).

Figura 29- Código transferência para oração

```
const handleClickAccessPrayer = item => () => {
  item.wasRead = !item.wasRead;
  setActivity(activity);
  savePrayerData();
  navigation.navigate('ReadScreen', {
    id: item.id,
    title: item.title,
    body: item.body,
  });
};
```

Fonte: Autor, 2021.

Podemos ver também de exemplo, previsto dentro do MVP definido, a funcionalidade “Definir Agenda” já codificada, nas **Figuras 30, 31, 32 e 33**.

Figura 30– Agenda sem atividades.



Fonte: Autor, 2021.

Figura 31– Tela para adicionar uma oração na agenda.



Fonte: Autor, 2021.

Figura 32– Agenda com atividades não concluídas.



Fonte: Autor, 2021.

Figura 33– Agenda com atividades concluídas.



Fonte: Autor, 2021.

7. CONCLUSÃO E CONSIDERAÇÕES FINAIS

O desenvolvimento do presente estudo possibilitou uma análise de como as pessoas se relacionam com Deus, tendo presente diversas as dificuldades contemporâneas. Com a necessidade de estar cada vez mais à frente do tempo, surgem dificuldades como a concentração e constância. Isso, nos alavancou à procura de ferramentas e métodos que poderíamos fazer uso para atingir a dor dos nossos usuários, e vimos que a atratividade, a experiência única e diferente a qual inserimos o nosso usuário, o ajuda no enfrentamento de suas deficiências comportamentais.

A prática efetiva e real de todo o processo de desenvolvimento de um software, do pensamento de escopo até sua codificação. Vivência real com todas as metodologias e processos visto no período estudantil, nos trouxe bastante conhecimento dentro de todo o processo da construção do software. Tendo prática e conhecimento de como funcionam os processos, nos leva a um patamar a cima dentro do contexto profissional.

Nesse sentido, vemos que a utilização de recursos digitais permite aos fiéis realizarem suas orações de forma mais eficiente, de forma concreta. Pensamos em diversas funcionalidades que podem vir a ser trabalhadas fora do escopo MVP, nos centralizando dentro do mercado, frente aos softwares com propostas semelhantes à nossa.

8. REFERÊNCIAS

ARISTÓTELES. **Da Alma – De Anima**. Edipro, 1 de Fevereiro de 2011.

ANDERSON, David J., CARMICHAEL, Andy. **Kanban Essencial**. Lean Kanban University, 28 de Julho de 2016.

BÍBLIA. Português. **Bíblia - Ave Maria**. Tradução de Padre Gerardus Jarussi, CMF. São Paulo, 2018. Edição Claretiana.

BUTLER, Alban. **Vida dos Mártires**. Publicação Minha Biblioteca Católica, 2020.

CARR, Nicholas. **The shallows: what the internet is doing to our brains**. W.W Norton & Company, 2010. Disponível em: http://www.postgrowth.ca/uploads/8/4/9/4/84946882/the_shallows_-_carr.pdf.

CELANO, Tomás. **A vida de São Francisco de Assis**. Publicação Minha Biblioteca Católica, 2018.

CRISÓSTOMO, São João. **Patrística São João Crisóstomo - Comentários às Cartas de São Paulo/1**. Paulus Editora, 2010.

FERREIRA, Paulo. **O avanço da tecnologia e as transformações na sociedade**. CNI, 10 de Outubro de 2017. Disponível em: <https://noticias.portaldaindustria.com.br/artigos/paulo-afonso-ferreira/o-avanco-da-tecnologia-e-as-transformacoes-na-sociedade/#>.

FLANAGAN, David. **JavaScript: The Definitive Guide 7th ed**. O'Reilly Media, 14 de Maio de 2020.

KROLL, Per, MACISAAC, Bruce. **Agility and Discipline Made Easy: practices from OpenUp and RUP**. Pearson, 2006. Disponível em: https://books.google.com.br/books?hl=pt-BR&lr=&id=xzU8aw4HiAsC&oi=fnd&pg=PT22&dq=KROLL,+P.+e+MCISAAK,+B.+Agility+and+Discipline+Made+Easy.+Pearson,+2006&ots=v_xjmnVxYa&sig=cR4tKljNSLsFi1I5TCrHd2PXvpQ#v=onepage&q&f=false.

LENON. **Node.js - O que é, como funciona e quais as vantagens**. Opus Software, 05 de Setembro de 2018. Disponível em: <https://www.opus-software.com.br/node-js/>.

LONGUINHO, João P. **O que é e Como aplicar a ferramenta 5W2H**. Blog Eu Administrador, 6 de abril de 2016. Arquivado do original em 15 de outubro de 2017. Disponível em: <https://web.archive.org/web/20171015095109/http://euadministrador.com/o-que-e-e-como-aplicar-a-ferramenta-5w2h/>.

POSTGRESQL .Disponível em: <https://www.postgresql.org/>.

REACTNATIVE. Disponível em: <https://reactnative.dev/>.

SCHWABER, Ken, SUTHERLAND, Jeff. **The Definitive Guide to Scrum: The Rules of the Game**. The Scrum Guide, Novembro de 2017. Disponível em:

<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>.

SENE, Rafael Peria de . **OpenUp: Uma visão geral**. Tiespecialistas, 2010. Disponível em: <https://www.tiespecialistas.com.br/openup-uma-visao-geral/>.

TRESE, Leo J. **A fé Explicada, 14° Edição**. Quadrante, 2014.

VATICANO. **CATECISMO DA IGREJA CATÓLICA**. 3ª. ed. Petrópolis: Vozes; São Paulo: Paulinas, Loyola, Ave-Maria, 1993.

VIRILIO, Paul. (1999). **A bomba informática**. (L. V. Machado, Trad.). São Paulo: Estação Liberdade, Obra original publicada em 1998.