

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

RODOLFO VIOLA CARVALHO

**Proposta de uma plataforma *codeless* para
implementação de apps de promoção da
saúde**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em
Engenharia da Computação

Orientador: Prof^ª. Dra. Érika Fernandes Cota

Porto Alegre
2021

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões

Vice-Reitora: Prof^a. Patricia Pranke

Pró-Reitora de Graduação: Prof^a. Cíntia Inês Boll

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Engenharia de Computação: Prof. Walter Fetter Lages

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

“Imagination is more important than knowledge.”

— ALBERT EINSTEIN

AGRADECIMENTOS

A Deus por ter me dado tudo que sempre precisei para alcançar este objetivo.

Aos meus pais e irmãos por todo apoio incondicional e irrestrito durante esta longa caminhada da graduação.

A minha namorada que me apoiou desde o início, por todo amor e parceria compartilhados e pela paciência.

Aos professores do Instituto de Informática e da UFRGS pelo auxílio prestado ao longo destes anos de graduação.

Agradeço à professora Erika, pela orientação deste trabalho, apoio e confiança.

A Universidade Federal do Rio Grande do Sul por proporcionar que um sonho se tornasse realidade.

RESUMO

O uso de dispositivos móveis por profissionais de saúde transformou diversos aspectos da prática clínica. Os dispositivos móveis tornaram-se comuns em ambientes de saúde. Vários aplicativos estão agora disponíveis para auxiliar os profissionais de saúde do inglês, HCPs *Health Care Professionals* em tarefas importantes. Este estudo parte da hipótese de que o *mobile health* (mHealth) não é largamente adotado ainda devido à dificuldade tecnológica para o desenvolvimento de aplicativos de saúde com elementos individualizados de informações de engajamento. Esta dificuldade surge do alto custo (financeiro e temporal) de desenvolvimento de tal solução usando as abordagens disponíveis. Embora os benefícios do mHealth já tenham sido demonstrados, seu uso no Brasil ainda é incipiente. Nota-se que as soluções disponíveis oferecem mecanismos genéricos de engajamento e informação, oferecendo um fluxo pré-determinado e único para todos os usuários, embora os fatores de adesão ou não a um tratamento de saúde sejam bastante particulares e individual para cada pessoa. Considerando a complexidade desta proposta, este trabalho apresenta uma prova de conceito da plataforma *codeless*, fazendo o uso dos conceitos de *flow based programming* (FBP) e *visual based programming* (VPL), focando na possibilidade de geração rápida e sem codificação de um aplicativo contendo mecanismos customizados de elementos de engajamento com a possibilidade de customização/individualização de elementos de informação e engajamento para grupos de pacientes. Assim sendo, como resultado desse trabalho foi possível criar um ecossistema de software, onde os profissionais de saúde conseguem programar um conjunto de elementos de engajamento a fim de aumentar a adesão dos pacientes no tratamento. Por fim, nota-se que ainda existem várias possíveis áreas de evoluções do sistema para contemplar um funcionamento que reflète um fluxo complexo criado pelos HCPs.

Palavras-chave: Engajamento do usuário. mHealth. *codeless*. *component based software*.

LISTA DE ABREVIATURAS E SIGLAS

HCPs	Health Care Professionals
FBP	Flow Based Programing
VPL	Visual Programing Language
LCDPs	Low-code Development Platforms

LISTA DE FIGURAS

Figura 3.1 Fluxo de interação médico e paciente com o software	23
Figura 3.2 Proposta software de FBP e VPL.	28
Figura 3.3 Operação do Frontend.	29
Figura 3.4 Exemplo de sequência de nodos a serem processados.	30
Figura 3.5 Proposta software mobile.	31
Figura 4.1 Dashboard com fluxos.	33
Figura 4.2 Exemplo de fluxo com dois nodos e uma conexão.	34
Figura 4.3 Funcionalidade arraste e puxe para adicionar novo nodo.	36
Figura 4.4 Página de construção de fluxos.	37
Figura 4.5 Edição de metadado de um nodo.	37
Figura 4.6 Lógica da execução de nodos com mural.	42
Figura 4.7 Fluxograma da execução dos nodos.	43
Figura 4.8 Ordem temporal da execução dos nodos.	44
Figura 4.9 Mural de atividades do usuário.	45
Figura 4.10 Atividade lembrete.	46
Figura 4.11 Atividade entrada de dados.	47

LISTA DE TABELAS

Tabela 3.1 Mecanismos de interação e engajamento	26
--	----

SUMÁRIO

1 INTRODUÇÃO	10
1.1 Motivação e escopo	12
1.2 Descrição do problema	13
1.3 Objetivos	13
1.4 Organização e trabalhos futuros	14
2 FUNDAMENTAÇÃO TEÓRICA E REVISÃO DA LITERATURA	15
2.1 mHealth	15
2.1.1 Trabalhos relacionados	17
2.1.2 Profissionais da área da saúde e dispositivos móveis.....	17
2.2 Mecanismos de engajamento	18
2.3 Arquitetura cliente-servidor	19
2.3.1 Cliente	19
2.3.2 Servidor.....	20
2.4 Programação baseada em fluxo	20
2.5 Programação visual	21
2.6 Plataformas <i>Codeless</i>	22
3 PROPOSTA	23
3.1 Mecanismos de interação e engajamento	23
3.1.1 Compartilhamento Social.....	24
3.1.2 Conteúdo direcional	24
3.1.3 Entrada de dados	24
3.1.4 Gamificação	25
3.1.5 Comparativo de funcionalidades.....	25
3.2 Tipos de nodos	26
3.2.1 Nodos de agendamento de execução (cronômetro)	26
3.2.2 Nodos com dependência de entrada de dados do criador do fluxo (HCPs).....	27
3.2.3 Nodos com independência de entrada de dados do criador do fluxo (HCPs).....	27
3.2.4 Nodos com dependência de entrada de dados dos pacientes	27
3.3 Software visual codeless para construção de fluxos	28
3.4 Fluxo da aplicação	28
3.5 Estruturas de software: backend e frontend	29
3.6 Interface com paciente	30
4 IMPLEMENTAÇÃO	32
4.1 Ferramenta FBP e VBP	32
4.1.1 Estruturas e programação de fluxos	33
4.1.2 Mecanismo de programação visual.....	35
4.2 Backend e APIs	38
4.2.1 Descrição do nodo para frontend	38
4.2.2 Execução e resultado da execução do fluxo.....	41
4.3 Software <i>Mobile</i>	44
5 CONSIDERAÇÕES FINAIS	48
REFERÊNCIAS	50

1 INTRODUÇÃO

O uso de dispositivos móveis por profissionais de saúde do inglês, HCPs *Health Care Professionals* transformou diversos aspectos da prática clínica. Os dispositivos móveis tornaram-se comuns em ambientes de saúde, levando a um rápido crescimento no desenvolvimento de aplicativos médicos para essas plataformas (WALLACE; CLARK; WHITE, 2012; AUNGST, 2013). Vários aplicativos estão agora disponíveis para auxiliar os HCPs em tarefas importantes, como: gerenciamento de informações e tempo, manutenção e acesso aos registros de saúde, comunicações e consultoria, referência e coleta de informações, gerenciamento e monitoramento de pacientes, tomada de decisão clínica e educação e treinamento médico (KISER, 2011; OZDALGA; OZDALGA; AHUJA, 2012; YOO, 2013; O'NEILL et al., 2013).

Atualmente, os sistemas de software fornecem inúmeros benefícios para HCPs, melhorando a tomada de decisões clínicas e trazendo melhores resultados para os pacientes (MURFIN, 2013; MICKAN et al., 2013). Eles permitem tomadas de decisões mais rápidas com menores taxas de erros, aumentando a qualidade do gerenciamento de dados, acessibilidade, eficiência da prática e conhecimento (WALLACE; CLARK; WHITE, 2012; MOSA; YOO; SHEETS, 2013; DIVALL; CAMOSSO-STEFINOVIC; BAKER, 2013; MICKAN et al., 2013). Destaca-se que esses benefícios demonstraram um efeito positivo nos resultados do atendimento ao paciente, evidenciando uma redução nos efeitos adversos e no tempo de internação hospitalar (DIVALL; CAMOSSO-STEFINOVIC; BAKER, 2013; MICKAN et al., 2013). A pandemia Covid-19 criou um ímpeto sem precedentes para o desenvolvimento da saúde móvel. Este desenvolvimento envolveu tanto o aprimoramento e padronização de soluções já consolidadas em saúde digital e a exploração de novos potenciais. (MING et al., 2020) (GIANSANTI, 2021)

A universalidade dos telefones celulares gerou uma oportunidade imensa para o fornecimento de soluções inovadoras para alguns dos problemas de saúde mais urgentes, incluindo a prevenção e o tratamento de doenças crônicas. O termo *mobile health* (mHealth) se refere ao uso de dispositivos móveis para melhorar a saúde e fornecer cuidados, concebido pela primeira vez há mais de uma década (BASHSHUR et al., 2011). A introdução do primeiro iPhone em 2007 marcou o início de uma década de avanços tecnológicos que têm proporcionado intervenções cada vez mais interativas para mudança de comportamento (ALI; CHEW; YAP, 2016). Entre 2011 e 2018, a utilização de *smartphones* cresceu de 35% para 77% dos americanos, e 95% possuem um telefone móvel de

algum tipo (SHEET, 2018). No Brasil, segundo um estudo de 2020 coordenado pela Fundação Getúlio Vargas (FVG), existem 424 milhões de dispositivos digitais em uso (MEIRELLES, 2020).

Os altos números de dispositivos sugerem que uma grande vantagem das iniciativas de saúde móvel é sua ampla acessibilidade. O mHealth não é apenas acessível, mas também possui um grande apelo para os consumidores. O número de instalações de aplicativos (apps) de bem-estar e saúde atingiu cerca de 3,35 bilhões em todo o mundo (AITKEN; CLANCY; NASS, 2017). À medida que a popularidade da saúde móvel aumentou, o número de tecnologias e recursos alavancados por intervenções móveis também aumentou. Essas interferências tornaram-se cada vez mais sofisticadas em sua tecnologia e diversificada em suas abordagens para mudar e manter a adesão aos tratamentos de saúde.

A saúde móvel possui um alto potencial de melhoria ao acesso à saúde, especialmente nos países em desenvolvimento socioeconômico. Desta maneira, os resultados das intervenções de saúde por meio do engajamento dos pacientes às condutas médicas é de grande relevância uma vez que, quanto maior a adesão ao tratamento, maior será a chance deste indivíduo se recuperar ou até mesmo prevenir problemas de saúde.

Uma pesquisa conduzida pela Intelligence (2017) afirmou que o envolvimento do paciente por meio de ferramentas mHealth é uma grande tendência na área da saúde dos Estados Unidos. A pesquisa revelou que 60% dos pacientes sentem que é importante o acompanhamento médico após uma consulta e 80% de todos os pacientes gostariam de ver seus resultados e diagnósticos online. Já no Brasil, existem poucos aplicativos publicados na área da saúde e uma alta demanda pelos serviços públicos de saúde.

Uma revisão sistemática e meta-analítica indicou que a intervenção mHealth em pacientes diabéticos e hipertensos pode melhorar muito os resultados clínicos, reduzir custos, melhorar a qualidade de vida e aumentar a autoeficácia entre pacientes, inclusive em países com diferentes níveis de desenvolvimento econômico. O estudo também enfatizou que a combinação da intervenção mHealth com a gestão profissional é fundamental para atingir a eficácia clínica ideal (MARCOLINO et al., 2018). Estas descobertas sugerem que os usuários apreciam recursos que economizam tempo em relação aos métodos atuais e identificam um aplicativo como valioso quando é simples e intuitivo de usar, fornece instruções específicas para gerenciar melhor uma condição e compartilhar dados com indivíduos designados.

O envolvimento em intervenções de mudança de comportamento em saúde eletrô-

nica é considerado importante para a eficácia da intervenção. Entretanto, para melhorar efetivamente o envolvimento do paciente com as ferramentas de saúde móvel, é necessário avaliar alguns aspectos que pertencem ao âmbito dos sistemas de software, como: notificações via *pushes*, *gamificação*, lembretes diários, registro e rastreamento de informações da saúde individual, fornecimento de conteúdo e até orientações de saúde personalizadas.

O sucesso de uma intervenção comportamental depende do envolvimento ativo dos participantes no tratamento e o comprometimento com a conduta possui aspectos comportamentais, afetivos e cognitivos que contribuem para maximizar os resultados positivos do procedimento. Portanto, a adesão é um componente chave de qualquer avaliação da eficácia do tratamento. Por isso, se faz necessário uma solução mHealth que permita ao profissional de saúde fazer individualizações de condutas com o paciente e para o paciente.

1.1 Motivação e escopo

Segundo a Organização Mundial da Saúde (OMS), a não adesão aos tratamentos a longo prazo na população em geral está em torno de 50% (TAVARES et al., 2016). Já o Ministério da Saúde diz que a falta de elaboração de planos terapêuticos personalizados entre o profissional da saúde e o paciente gera uma baixa adesão aos tratamentos (CIÊNCIA; TECNOLOGIA, 2016), desencadeando a falha terapêutica e desperdício de recursos como consequências. Sabe-se que o maior benefício do mHealth é o auxílio na adesão ao tratamento, mantendo o paciente informado tanto sobre a doença quanto sobre o tratamento e o risco decorrente do seu abandono. Tendo em vista as aplicações já disponíveis para o público nas lojas de aplicativos, nota-se que tais aplicações se baseiam em três grandes pilares para o engajamento dos pacientes, sendo: conteúdo direcionado, compartilhamento social e entrada de dados, todos com ênfase nos pacientes.

Embora os benefícios do mHealth já tenham sido demonstrados, seu uso no Brasil ainda é incipiente. É possível observar a existência de poucas aplicações móveis para o público da área da saúde nas lojas de aplicativos nacionais, comparados com a mesma categoria em lojas internacionais. Além disso, nota-se que as soluções disponíveis oferecem mecanismos genéricos de engajamento e informação, oferecendo um fluxo pré-determinado e único para todos os usuários, embora os fatores de adesão ou não a um tratamento de saúde sejam bastante particulares e individualizados. Por fim, embora algumas soluções permitam a comunicação do paciente com os HCPs, o aplicativo de saúde

ainda é um elemento secundário de apoio ao tratamento, o que dificulta seu uso regular tanto pelo paciente como pelos HCPs.

1.2 Descrição do problema

Esse trabalho parte da hipótese de que o mHealth não é largamente adotado ainda devido à dificuldade tecnológica para o desenvolvimento de aplicativos de saúde com elementos individualizados de informações de engajamento. Esta dificuldade surge do alto custo (financeiro e temporal) de desenvolvimento de tal solução usando as abordagens disponíveis, quais sejam: a contratação de uma equipe de TI ou de uma *software house*. De fato, a construção de uma solução de qualidade nesta abordagem exige um alto envolvimento da equipe de saúde com a equipe de desenvolvimento e a manutenção de uma equipe de desenvolvimento dedicada para a evolução do produto. Nesse contexto, propõe-se a construção de uma plataforma *codeless* que permita aos HCPs a construção de aplicativos de saúde de maneira rápida e com a possibilidade de customização/individualização de elementos de informação e engajamento para grupos de pacientes.

Considerando a complexidade desta proposta, este trabalho apresenta uma prova de conceito da plataforma *codeless*, focando na possibilidade de geração rápida e sem codificação de um aplicativo contendo mecanismos customizados de elementos de engajamento. Dessa forma, busca-se responder a seguinte pergunta:

É possível construir uma plataforma *codeless* que possa ser usada por HCPs para construir, de maneira rápida e com pouco ou nenhum esforço de programação, um aplicativo móvel com elementos engajadores customizados?

1.3 Objetivos

A partir da pergunta definida acima, o objetivo deste trabalho é avaliar a possibilidade de se construir uma plataforma *codeless* que permita a construção (ou síntese) de um aplicativo móvel contendo funcionalidades específicas de engajamento a um tratamento de saúde. Importante notar que, embora o objetivo do final da solução proposta seja um aplicativo móvel para uso de um paciente, este trabalho tem como foco a construção da plataforma *codeless* que fará a síntese deste aplicativo. Tal plataforma tem como público alvo os HCPs, que deverão usá-la para instanciar as funcionalidades que farão parte do

aplicativo móvel de um ou mais pacientes. Busca-se a especificação e implementação de uma solução que, a partir de uma base de código e componentes de elementos engajadores já identificados (conteúdo direcionado, compartilhamento social e entrada de dados) e pré-programados, permitindo sua combinação livre sem a necessidade de programação. Esta solução deverá permitir, ainda, a síntese automática de um aplicativo móvel contendo esses elementos e permitindo a interação com o usuário final (paciente). Desse modo, pretende-se criar mecanismos da computação que permitam aos HCPs criar soluções MHealth para aumentar o engajamento dos pacientes aos tratamentos de saúde.

1.4 Organização e trabalhos futuros

A organização deste trabalho se manifesta na ordem de capítulos, sendo o Capítulo 2 a revisão dos principais tópicos estudados nesse trabalho e o Capítulo 3 uma revisão sobre os mecanismos de engajamento presentes no mHealth e uma proposta de arquitetura. O Capítulo 4 expõe a proposta de desenvolvimento com os principais elementos do desenho da solução. O Capítulo 5 contém as considerações finais, trabalhos futuros e conclusões. É possível identificar alguns padrões de melhorias no fluxo feito pelos HCPs e proposto nesse trabalho. A existência de uma lógica embutida no tratamento faz com que a personalização seja maior e mais dinâmica, esta lógica se baseia nas ações já efetuadas pelos usuários ou por fatores externos, podendo desviar do fluxo tradicional com uma condicional. Um exemplo cotidiano para os HCPs é: fazer ação A se ação B concluída. Tal tipo de customizações lógicas (maior, menor, se, igual, etc...) será concebida em trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA E REVISÃO DA LITERATURA

Este capítulo trata da fundamentação teórica afim de realizar o embasamento do estudo, trazendo os principais significados e termos para fundamentar o princípio do trabalho. Também, faz parte deste capítulo uma revisão da literatura, onde se faz necessário uma análise ampla dos estudos recentes desta área do conhecimento.

2.1 mHealth

MHealth é uma abreviatura de saúde móvel, do inglês *mobile health*, um termo usado para a prática da medicina e saúde pública suportada por dispositivos móveis, a aplicação mais comum do mHealth é o uso de dispositivos móveis para educar os usuários sobre os serviços de saúde. Com a finalidade de buscar elementos engajadores, práticas e algumas soluções já disponíveis no mercado, foi realizada uma análise das aplicações disponíveis no mercado nacional e internacional. O critério principal para seleção das aplicações foi a presença de pelo menos um elemento engajador e que o app estivesse no ranking das aplicações mais baixadas das lojas de aplicativos. É possível notar que em quase todos os casos, as aplicações fornecem conteúdos direcionados ao público alvo da aplicação, com o objetivo de educar e informar seus usuários. Todas as aplicações se direcionam para um grupo selecionado de pessoas, geralmente portadores de problemas crônicos, tais como: cardiopatias, diabetes, hipertensão arterial, câncer, entre outras. Por fim, todas as aplicações contam com mecanismos diferentes de entrada de dados que variam dos mais simples, como humor do dia, até mais robustos, como uma série de perguntas e respostas. As principais ferramentas analisadas serão discutidas a seguir.

- Sharecare: este software tem como principal característica fornecer informações, dicas, dietas e outros conteúdos digitais onde os próprios usuários consigam se inscrever nos tópicos para um acompanhamento mais detalhado de algumas das principais doenças. Também irá trazer notícias e informações relevantes para o público selecionado por temas. Os segmentos disponíveis na aplicação são: alergias, câncer, diabetes tipo 2, coronavírus, problemas de coração, hipertensão, saúde da mulher e tabagismo. A aplicação tem como objetivo atualizar os usuários, trazendo

conhecimento sobre os tópicos seguidos para o mesmo. Desta forma, existe uma funcionalidade em que o usuário pode cadastrar alguns dados relacionados à sua saúde, tais como: glicemia, pressão arterial, colesterol e alimentação. Estes dados serão armazenados em uma estrutura de dados para posteriormente realização de análises das informações coletadas.

- **Livongo:** é um software cujo principal objetivo é criar planos personalizados para atender alguns problemas de saúde e fornecer tópicos de prevenção e programas de manutenção da saúde. Possui como diferencial equipamentos especiais (hardware) que medem sinais e sintomas dos pacientes, enviados para a casa dos usuários através de aparelhos conectados à rede de internet. A partir destes dados coletados, o software irá construir métricas sobre a saúde do paciente, gerando relatórios para os HCPs, além de criar alguns alertas básicos sobre a saúde do utilizador. As principais categorias cobertas por ele são: prevenção de diabetes, manutenção de diabetes, pressão alta, controle de peso e saúde comportamental.
- **Medisafe:** com enfoque na parte medicamentosa, a Medisafe contempla programas para melhorar a adesão dos pacientes aos tratamentos. Sua principal funcionalidade é a de lembrete do medicamento, onde o usuário recebe notificações para lembrar de tomar as medicações nos horários prescritos. Além disso, possui outras utilidades, como a adição de notas e sinais de saúde, dentre eles: peso, gordura corporal, colesterol e etc. Por fim, através destas anotações e lembretes é possível gerar um relatório com o andamento das métricas coletadas.
- **Omada:** a fim de incentivar hábitos saudáveis, este software traz cinco programas digitais onde os pacientes conseguem: controlar a glicemia, controlar a hipertensão e melhorar a saúde mental através de terapia virtual. Além disso, o Omada possui uma ferramenta que permite ao usuário se consultar com um especialista para sanar suas dúvidas. Através da aplicação é possível criar metas de objetivos pessoais em que o usuário fornece informações diariamente, através de formulários, com o propósito de completar o objetivo do programa em que está inscrito.
- **Wellth:** esta aplicação possui como meta resolver os problemas da má adesão aos cuidados de saúde, trazendo três principais soluções: aderência à medicação, cui-

dados com a saúde pessoal e otimização nas métricas de qualidade de saúde, como lembretes para consultas e exames. Seu principal diferencial é que, além das notificações, os usuários poderão receber uma recompensa financeira pela boa adesão ao tratamento. Os programas disponíveis são: saúde mental, diabetes, câncer, problemas respiratórios, cardiopatias e doenças autoimunes.

- **Cuco:** é um software que auxilia pacientes a seguirem seu plano de cuidado de saúde. A adesão ao tratamento e monitoramento da patologia são seus principais pilares. A aplicação possibilita que os HCPs acompanhem a evolução dos pacientes através das métricas coletadas pelo software e enviadas periodicamente. Também apresenta recursos de agendamento de lembretes de medicamentos e um programa de educação sobre as patologias.

2.1.1 Trabalhos relacionados

Alguns estudos apontam que a personalização é a chave para a criação de aplicativos de saúde digital de sucesso (MADEIRA et al., 2018) Evidências recentes relacionam personalidade e preferência por experiência digital - sugerindo que as personalizações podem ser uma base promissora para serviços digitais de saúde. Com base em uma revisão da literatura, é possível categorizar os conceitos de personalização associados à mudança de comportamento no mHealth em 4 dimensões, usuários, funcionalidades do sistema, informações e propriedades do aplicativo (GOSETTO; EHRLER; FALQUET, 2020). A personalização da experiência do usuário pode ser particularmente importante para manter o paciente engajado com o aplicativo para um uso de longo prazo para aproveitar ao máximo os recursos do aplicativo (KHWAJA et al., 2021).

2.1.2 Profissionais da área da saúde e dispositivos móveis

Muitas revisões sistemáticas avaliaram a eficácia da saúde móvel para uma ampla gama de comportamentos e revelaram que mesmo intervenções simples como mensagens de texto, podem melhorar a cessação do tabagismo, perda de peso e adesão à medicação, e que a mudança de comportamento pode permanecer mesmo após a interrupção de uma intervenção (ARMANASCO et al., 2017; HALL; COLE-LEWIS; BERNHARDT,

2015; KITSIOU et al., 2017; MARCOLINO et al., 2018). Estudos revelam resultados igualmente promissores para o auto gerenciamento do diabetes, perda de peso e gerenciamento de outros problemas de saúde (WHITEHEAD; SEATON, 2016; BONOTO et al., 2017; CUI et al., 2016; MATEO et al., 2015).

A literatura destaca a diversidade nas intervenções de saúde móvel para melhorar o estilo de vida e os comportamentos de doenças crônicas. Essa diversidade só aumentará à medida que tecnologias mais sofisticadas permitirem a implementação de novas técnicas de mudanças de comportamento. As intervenções atuais de saúde móvel geralmente agrupam uma infinidade de técnicas de mudança de comportamento, tornando difícil discernir quais recursos levam ao sucesso ou ao fracasso. Esta pluralidade na diversidade das intervenções implica em alguns dos elementos se repetirem entre as soluções, mas também mostra que existe uma individualidade no tratamento de certas doenças.

Dugas et al. (2018), Hales et al. (2017) afirmam que indivíduos com características de personalidade distintas, incluindo idade, se beneficiam de maneiras diferentes com tratamentos de saúde móvel apontando para a necessidade de ir além das intervenções que são genéricas para todos os pacientes.

Com esta perspectiva do panorama das aplicações disponíveis, é possível observar que tais sistemas de software focam nas necessidades exclusivamente dos pacientes e tem abordagens genéricas, ou seja, o mesmo que é usado por todos os pacientes e de maneira não interativa com o tratamento e equipe de saúde. Os programas apresentadas não solucionam os problemas apresentados neste trabalho.

2.2 Mecanismos de engajamento

Um dos grandes desafios no mercado de aplicativos móveis é manter o usuário engajado a longo prazo ou de forma contínua durante o tratamento, evitando a perda de interesse. O engajamento é uma consequência da profundidade da participação que um usuário pode alcançar ao interagir com os recursos disponíveis em um aplicativo (HARMARI et al., 2016).

Na área da saúde, a falta de comprometimento com o tratamento é uma preocupação constante dos HCPs, que vêm adotando a tecnologia como um fator motivacional para os pacientes (CECHETTI et al., 2019). Embora os aplicativos mHealth tragam benefícios e forneçam informações que podem auxiliar no tratamento do paciente, Ribeiro et al. (2016) argumenta que a maioria dos usuários não seguem as recomendações propostas

pelos médicos.

O uso de técnicas que aumentam o engajamento dos usuários é uma estratégia que vem sendo utilizada para motivar mudanças de comportamento e instigar os usuários a concluírem tarefas e atingirem metas de forma mais adequada. Os mecanismos de engajamento e interações são as principais funcionalidades que os mHealth disponibilizam para seus usuários.

2.3 Arquitetura cliente-servidor

O sistema cliente-servidor pode ser definido como uma arquitetura de software em que os clientes sempre enviam solicitações enquanto o servidor responde às solicitações enviadas (KRATKY; REICHENBERGER, 2013). Cliente-servidor fornece uma comunicação entre processos pois envolve a troca de dados do cliente e do servidor, em que cada um deles desempenha funções diferentes (SVOBODOVA, 1985).

Com o avanço da tecnologia, a Web está se tornando essencial no dia a dia das pessoas pois praticamente tudo envolve seu uso. Além disso, a aplicação da Web não se limita aos computadores mas está aberta a diferentes tipos de dispositivos digitais inteligentes, como por exemplo os móveis (ZHANG, 2013; GUIMARAES; IGBARIA, 1997). A arquitetura cliente-servidor tem minimizado o tempo de desenvolvimento de aplicativos, dividindo as funções de compartilhamento de informações no cliente e no servidor. O cliente é o solicitante enquanto o servidor é o provedor de serviço. Na maioria dos ambientes cliente-servidor, o processamento de dados é feito pelo servidor e os resultados são devolvidos aos clientes, a comunicação entre o cliente e o servidor é do estilo transacional e cooperativo. A natureza transacional significa que o servidor envia de volta para o cliente somente os dados relevantes (SVOBODOVA, 1985; FUNKHOUSER, 1995).

2.3.1 Cliente

O processo de cliente é ativo, ou seja, ele que faz a solicitação de serviços a outros programas, os servidores. Normalmente o cliente é dedicado à sessão do usuário, começando e terminando com a sessão. Um cliente pode interagir com um ou mais servidores, mas pelo menos um processo servidor é necessário. A nível de aplicação, o primeiro ponto a residir no cliente é a interface com o usuário. Algumas tarefas a serem realizadas

pelo cliente são: manipulação de tela, entrada e validação dos dados, etc. Para o usuário a impressão é que a aplicação está sendo rodada apenas localmente (BERTOCCO et al., 1998; SVOBODOVA, 1985).

2.3.2 Servidor

Servidores são programas que respondem às solicitações por serviços compartilhados. São um processo reativo disparado pela chegada de pedidos de seus clientes. Geralmente, o processo servidor roda o tempo todo, oferecendo serviços a muitos clientes. Em alguns sistemas, o processo servidor ao invés de responder diretamente, cria um processo filho exclusivamente para cada pedido de cliente. Um servidor processa a informação sem interagir com outros servidores. Os clientes que interagem com mais de um servidor tem a responsabilidade de ativá-los quando necessário. O processamento do servidor geralmente inclui: armazenar, organizar os dados compartilhados e atualizar dados (GUIMARAES; IGBARIA, 1997; FUNKHOUSER, 1995).

2.4 Programação baseada em fluxo

A programação baseada em fluxos (do inglês, FBP) é um paradigma de programação que decompõe os sistemas de software em uma rede de processos e modela esses sistemas como um grafo direcionado de processos predefinidos que são executados de forma assíncrona e trocam dados através de portas de entrada e saída (SOUSA, 2012). Esse paradigma foi definido por J. Paul Rodker Morrison no final dos anos 60, que usava uma metáfora de "fábrica de processamento de dados" para projetar e construir software (MORRISON, 1994). O FBP define aplicativos como redes de processos de "caixa preta" que se comunicam por meio de blocos de dados (chamados de Pacotes de Informação) que viajam por conexões predefinidas, especificadas externamente aos processos. Esses processos de caixa preta podem ser reconectados indefinidamente para formar diferentes software sem a necessidade de serem alterados internamente (WADGE; ASHCROFT et al., 1985; CULLER, 1986).

Um dos pilares fundamentais da programação baseada em fluxo é a ideia de um nó, processo ou módulo isolado. Isso pode ser pensado como uma parte do código-fonte do programa que não tem nenhuma dependência direta de nenhum outro módulo, quase

tornando o módulo uma parte autônoma do sistema que pode ser chamada sempre que necessário (SOUSA, 2012). Cada componente não depende de ser chamado sequencialmente com outro componente, portanto, os componentes de um software podem ser organizados e usados em qualquer ordem, permitindo a existência de vários fluxos de dados exclusivos à medida que as informações se movem pelo sistema (WADGE; ASHCROFT et al., 1985).

Cada um dos módulos individuais na programação baseada em fluxo aceita dados por meio de uma interface abstrata conhecida como porta, que opera de forma muito semelhante a uma porta de dados em uma rede de computadores. Os dados são enviados a uma porta por meio de uma ligação. Uma única porta pode estar relacionada a várias instâncias de um único componente, tornando a estrutura fácil de usar em um sistema distribuído ou para processamento paralelo (LEE; HURSON, 1994; CULLER, 1986).

Os dados contidos em uma mensagem são enviados e recebidos através dos módulos por um sistema de mensagem externo. O sistema de mensagens basicamente não tem conhecimento do que os módulos estão fazendo ou não e não há nenhuma preocupação real sobre quais dados estão contidos nas mensagens que os módulos estão processando (MORRISON, 1994).

Grande parte do código é encapsulado, portanto, o código-fonte tem um alto potencial de reutilização. Isso significa que atualizar ou dimensionar um aplicativo que usa programação baseada em fluxo pode ser mais fácil do que com um aplicativo mais integrado, porque o sistema de mensagens, os módulos e o sistema de portas podem ser alterados independentemente, sem afetar o programa maior. O FBP também promove um estilo funcional de alto nível de especificações que simplifica o raciocínio sobre o comportamento do sistema (SOUSA, 2012).

2.5 Programação visual

Linguagem de programação visual (do inglês, VPL) permite aos usuários criarem ilustrações para descrever diferentes tipos de processos (BURNETT; MCINTYRE, 1995). É uma técnica desenvolvida para funcionar com a capacidade do ser humano de explicar conceitos por meios visuais. A inclusão de elementos gráficos a torna acessível a novos utilizadores. Esses elementos podem ser manipulados para construir software (ERWIG; MEYER, 1995).

A programação visual não funciona com informações abstratas como classes, ins-

tâncias ou entrada e saída. Pelo contrário, permite que uma pessoa construa uma solução para um problema de forma que possa ser facilmente compreendida por outros humanos, com isso, a linguagem de programação visual permite o desenvolvimento de programas de software, eliminando o código de software textual com uma série de elementos gráficos visuais. O VPL incorpora esses elementos gráficos como o contexto principal da linguagem organizados de forma sistemática (BOSHERNITSAN; DOWNES, 2004).

Os gráficos ou ícones incluídos em um programa visual servem como entrada, atividades, conexões ou saída do programa. Como exemplo, muitas aplicações VPLs usando FBP são baseados na ideia de "caixas e setas", onde caixas ou outros objetos de tela são tratados como entidades, conectados por setas, linhas ou arcos que representam relações (BURNETT; MCINTYRE, 1995).

2.6 Plataformas *Codeless*

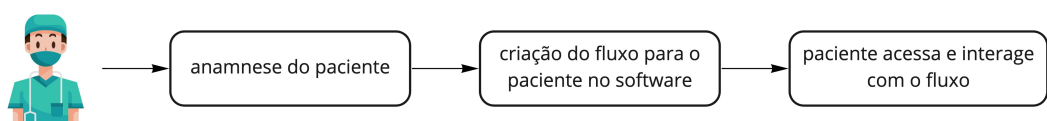
A programação sem código é também conhecida como "desenvolvimento de software sem código". Ela permite que qualquer pessoa, mesmo aquelas sem experiência em programação ou desenvolvimento de software, crie seus próprios aplicativos usando modelos e módulos em interfaces gráficas de usuário. As plataformas sem código permitem que os usuários criem aplicativos rapidamente por meio de uma ferramenta simples de arrastar e soltar. O uso de software sem código garante melhor segurança dos dados, melhora o gerenciamento e o controle dos dados e ajuda a comercializar novos aplicativos mais rapidamente com custos e riscos mais baixos (ITFORUM, 2020).

Com toda a flexibilidade das VPL juntamente com FBP é possível a criação de ferramentas completas com pouco código escrito pelo usuário final ou até mesmo sem nenhum código. Essas plataformas *codeless* são um conjunto de ferramentas que permitem o desenvolvimento visual de um sistema por meio de modelagem de elementos da interface gráfica (ALEXANDROVA; TATLOCK; CAKMAK, 2015; BURNETT; MCINTYRE, 1995).

3 PROPOSTA

Este trabalho sugere o uso de FBP e VPL para construção de uma plataforma que seja interativa, não requisite conhecimento de programação e viabilize a criação de fluxos de tratamentos, de um modo que seja intuitivo para os HCPs. Obtendo como proposta a interação dos pacientes com o conjunto de elementos de engajamentos pré selecionados pelos profissionais de saúde, considerando a complexidade e os problemas impostos, sendo assim, se faz necessário a construção de um conjunto de sistemas de software que irá compor desde a primeira interação e desenho do fluxo por parte dos HCPs até o meio onde irá acontecer as ações dos usuários. O foco deste estudo é viabilizar o conjunto de ferramentas que possibilita o fluxo apresentado na Figura 3.1 ser constituído de maneira digital e que possa ser individualizado ao nível do paciente. Fazendo uma correlação, quando os HCPs fazem uma anamnese e diagnosticam um problema no paciente, geram um mapa mental de ações para serem executadas em uma ordem cronológica e de sequência, sendo alguns exemplos: exames, interpretação de resultados de exames, novos atendimentos, tratamentos com medicamentos e outros. Dentre todos esses problemas, visamos resolver um deles, que é criar ações motivacionais ao longo do curso do tratamento a fim de aumentar a adesão do paciente aos procedimentos. Para isso, a solução proposta é elaborar todo um ambiente onde os HCPs possam criar e executar os mapas mentais de ações em formato de sequências ou cronológica para interagirem com seus pacientes.

Figura 3.1: Fluxo de interação médico e paciente com o software



miro

3.1 Mecanismos de interação e engajamento

Através desta revisão da literatura e análise de mercado, conclui-se que, atualmente, há quatro principais fatores utilizados no engajamento dos usuários, sendo eles: compartilhamento social, conteúdo direcionado, entrada de dados e gamificação. Estes

fatores, que serão detalhados abaixo, buscam com que os usuários utilizem o software de tempos em tempos, seja para completar uma tarefa ou fazer a leitura de um novo conteúdo, mantendo sempre ativa a interação com o aplicativo.

3.1.1 Compartilhamento Social

O compartilhamento social é quando o usuário compartilha intencionalmente uma informação sua com outros usuários, seja através de uma simples curtida em um conteúdo ou então, pelo compartilhamento de seu estado de saúde com o restante da comunidade presente no aplicativo ou nas redes. Estas ações executam um papel de prova social, gerando uma conexão entre os usuários da plataforma e promovendo a identificação entre as pessoas, aumentando o engajamento. Os exemplos mais notórios da aplicação dessa técnica são em aplicativos que promovem a interação entre os usuários como em blogs e fóruns, porém, a interação mais presente nas aplicações são as curtidas.

3.1.2 Conteúdo direcional

Os conteúdos desempenham uma função importante de chamar a atenção dos usuários e garantir o foco dos pacientes. Como estes já possuem interesse prévio no assunto, o conteúdo exerce um papel significativo na identificação do usuário com a aplicação. A promoção do engajamento pelo conteúdo pode ocorrer de diversas formas, através de vídeos, imagens, áudios e textos. Entretanto, esta última acaba sendo a mais utilizada na maioria das vezes. É válido destacar que a entrega desse conteúdo pode ser feita dentro da aplicação, no formato de um artigo longo, por exemplo, ou então como uma notificação push e um texto curto.

3.1.3 Entrada de dados

A entrada de dados é fundamental quando o assunto é mHealth. É deste modo que as aplicações coletam as informações. Os dados coletados são usados de diferentes maneiras, tanto para melhorar a experiência dos próprios usuários, quanto para gerar resultados para os pacientes e HCPs. Geralmente esse processo é feito por formulário, contudo, também pode ser feito de forma mais simplificada, como pela marcação de lei-

tura em um texto através de um botão ou qualquer tipo de informação proveniente de um hardware do usuário, como por exemplo, relógios inteligentes.

3.1.4 Gamificação

A Gamificação, é o uso de mecânicas e características de jogos para engajar, motivar comportamentos e facilitar o aprendizado de pessoas em situações reais, tornando conteúdos densos em um insumo mais acessível. O uso de elementos de jogos em aplicações convencionais está se tornando uma tendência amplamente aceita, indicando a eficácia da gamificação, especialmente em serviços relacionados à saúde (COTTON; PATEL, 2018). Um exemplo dessas mecânicas é a criação de recompensas virtuais para pacientes que tomarem os medicamentos nos horários corretos, ou então, praticarem atividades físicas semanalmente de acordo com o estipulado, por exemplo, no mínimo 30 minutos pelo menos 3 vezes na semana.

3.1.5 Comparativo de funcionalidades

Na Tabela 3.1 apresenta-se um comparativo entre todos os mecanismos discutidos acima com o intuito de trazer uma visão mais ampla do uso dos elementos nos sistemas de software mencionados. Nota-se que todos os mHealth apresentados implementam diversas maneiras de direcionamento do conteúdo e permitem a entrada de dados, ambos impulsionados pelo compartilhamento social. Os exemplos mais prevalentes derivados dos artifícios listados acima, consistem em: *pushes*, cursos, programas de conteúdo de saúde, formulários de entrada de dados, lembretes e hardware específico. A técnica menos popular é a de gamificação. Entretanto, apesar de ser menos utilizada, a gamificação possui usos interessantes como a possibilidade de oferecer recompensas virtuais (medalhas e pontos) e financeiras.

É notório que tais elementos de engajamentos são distintos, mas por outro lado, podem ser combinados de maneira que faça que a sequência deles represente um mapa mental do tratamento propostos pelos profissionais da saúde, desta forma, cada um dos tipos destes elementos podem virar um tipo de representação e assim essas representações possam ser combinadas a fim de criar um fluxo de tratamento. Para esquematizar o mapa mental dos HCPs é necessário criar uma estrutura que facilite computacionalmente

Tabela 3.1: Mecanismos de interação e engajamento

Aplicativo	Compartilhamento social	Conteúdo direcionado	Entrada de dados	Gamificação
Sharecare	SIM	SIM	SIM	NÃO
Livongo	SIM	SIM	SIM	NÃO
Medisafe	NÃO	SIM	SIM	NÃO
Omada	SIM	SIM	SIM	SIM
Wellth	SIM	SIM	SIM	SIM
Cuco	NÃO	SIM	SIM	NÃO

o processamento dos elementos, para isto, foi escolhido a representação de grafos. Nesta notação de grafos os elementos de engajamento irão ser representados pelos nodos e a sequência de execução irá ser representada pela ligação entre os nodos. Para ilustrar esta proposta, podemos exemplificar com um caso real de um paciente em tratamento de uma doença crônica, onde é necessário diariamente fazer a ingestão de todos os medicamentos e educar sobre os riscos, exercícios e alimentação. Neste exemplo se faz necessário a representação dos nodos: tempo ou recorrência, notificações e conteúdo. A combinação dos nodos de recorrência, notificação e conteúdo neste caso irá resultar em um fluxo de tratamento virtual onde o paciente poderá interagir com a equipe de saúde e se manter engajado com o progresso do procedimento.

3.2 Tipos de nodos

Observando a natureza dos elementos de engajamentos já vistos, é conhecido a distinção de seus mecanismos, porém conseguimos classificar e encaixar estes elementos em alguns tipos baseados na entrada de informação necessária para a execução do mecanismo de engajamento. Nas subseção abaixo estão listados os tipos e também relacionados com os elementos e as propostas de implementação destes mecanismos de engajamentos.

3.2.1 Nodos de agendamento de execução (cronômetro)

Responsável por criar as regras de execução do fluxo a partir do ponto em que ele inicia no grafo, o nodo cronômetro atua como um agendador de tarefas recorrentes de execução, a partir do momento em que um novo fluxo é salvo, ele fica responsável pela ativação do fluxo no momento mais adequado e início da execução.

3.2.2 Nodos com dependência de entrada de dados do criador do fluxo (HCPs)

Nodos que tem como dependência a entrada de dados do criador do fluxo podem ser agrupados e podem gerar uma gama grande de mecanismo de interação com os pacientes. O maior exemplo desta categoria são os nodos responsáveis pelo lembrete de alguma atividade, como a ingestão de um medicamento. Estes nodos podem também representar um formato de conteúdo direcional, onde os HCPs fazem a alimentação das informações para serem consumidas pelos pacientes. Uma característica destes nodos é que são idempotentes.

3.2.3 Nodos com independência de entrada de dados do criador do fluxo (HCPs)

Os nodos que são independentes de dados de entrada do criador utilizam sua lógica interna para conseguir gerar um resultado. Neste caso, representam o segmento da gamificação e podem ser exemplificados como: *pushes*, mensagens geradas automaticamente, baseadas no comportamento dos pacientes e textos motivacionais para aumentar o engajamento. Uma característica destes nodos é que as operações executadas destes nodos podem variar conforme o tempo, usuário, interação e outros.

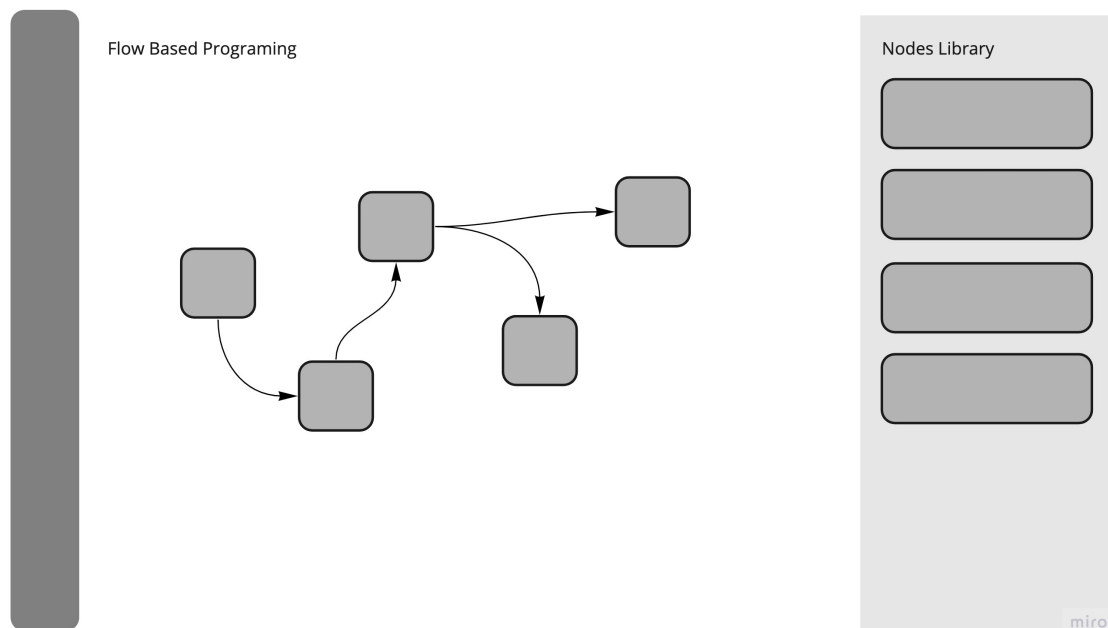
3.2.4 Nodos com dependência de entrada de dados dos pacientes

A interação com os pacientes gera informações e estas podem ser alimentadas nos fluxos por meio da entrada de dados. Fez-se necessário a criação de uma categoria de nodos que criam uma dependência de entrada de dados dos pacientes. Os nodos que são dependentes da entrada de dados dos pacientes só conseguem ser executados quando o paciente adiciona os requisitos de entrada do nodo. Neste caso, podemos exemplificar os nodos que buscam informações dos pacientes como por exemplo: quantidade de glicose diária, pressão arterial diária dentre outras informações. Uma característica destes nodos é que eles só serão executados quando os requisitos de entrada são preenchidos.

3.3 Software visual codeless para construção de fluxos

Analisando os principais fatores de engajamento estudados, propõe-se a elaboração de uma ferramenta baseada em FBP e VPL que contenha alguns destes fatores além de abranger as principais funcionalidades das aplicações FBP, VPL e codeless, criando uma biblioteca de elementos gráficos onde profissionais da área da saúde conseguirão criar fluxos codeless genéricos baseados em elementos visuais para cumprir com as demandas específicas necessárias de seus pacientes, como mostrado na Figura 3.2. Essa ferramenta possibilitará a criação de programas por meio de linguagem visual que poderá ser executada através de uma lógica de programação criada com fluxos. A ferramenta baseada em FBP e VPL será o frontend e sua lógica será desconhecida, sendo totalmente controlada pelo backend e API que conterá o mecanismo de processamento.

Figura 3.2: Proposta software de FBP e VPL.



3.4 Fluxo da aplicação

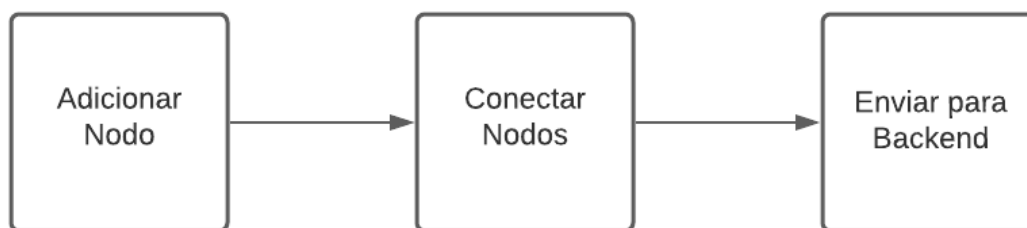
O domínio do software será dividido para duas personas, HCPs e pacientes, porém com enfoque principalmente nos profissionais da saúde. Desta forma, o sistema tem três grandes escopos de implementação, são eles: frontend, backend e aplicação móvel. Partindo do fluxo de interação médico paciente visto na Figura 3.1, o backend tem que ser

responsável por conectar toda a lógica do desenvolvimento do frontend com a aplicação móvel, o frontend fica somente com a lógica de construção de grafos e a aplicação móvel nos moldes das aplicações que já existem no mercado. Os HCPs serão responsáveis pela criação do FBP por meio de VBP. A criação dos fluxos pode ser feita a partir de um fluxo já existente ou através da elaboração de novos.

3.5 Estruturas de software: backend e frontend

Do ponto de vista do backend, é necessário a implementação de um software que irá fazer a tradução da linguagem produzida visualmente para código a ser processado e executado. Neste caso, propõe-se uma arquitetura cliente-servidor, onde a ferramenta de VPL (cliente) irá comunicar com o servidor (backend), enviando a estrutura de dados e fluxo criado para armazenamento e execução do fluxo. Utilizando-se os conceitos do FBP os nodos são como "caixas pretas". Desta forma o cliente não fica responsável pela lógica de execução, sendo possível o backend ficar responsável por definir quais os nodos disponíveis e enviar para o frontend *on the fly*. Assim, o frontend desconhece a lógica de execução e, por sua vez, o software visual fica possibilitado em montar uma sequência de nodos e com a responsabilidade apenas de enviar a programação baseada em fluxo criada pelo usuário para o backend, como mostrado na Figura 3.3.

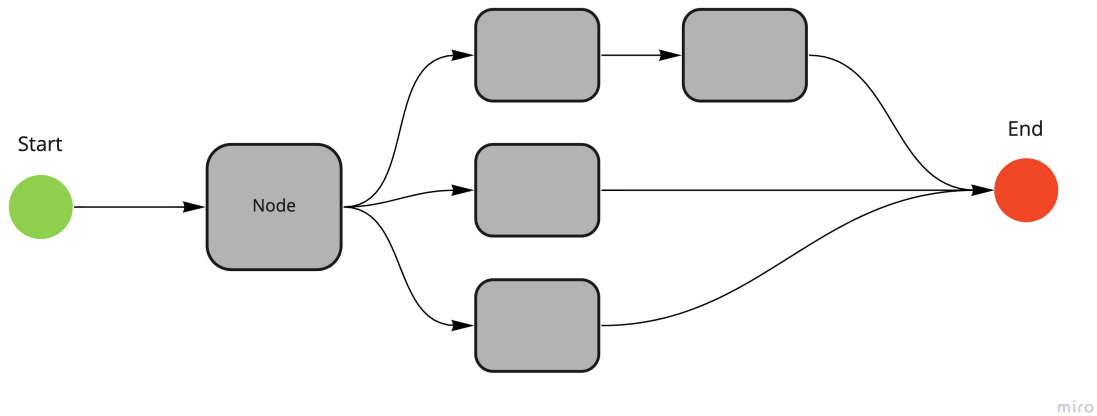
Figura 3.3: Operação do Frontend.



O processamento de nós irá acontecer de um nó para o outro, seguindo a sequência determinada pelo nó executado, previamente ligados pelos usuários. O início da execução é dado por um nó de tipo especial, que é o nó de início (cronômetro). Após o reconhecimento dos nós de início, irá acontecer um processamento recursivo, onde o resultado do nó pai será a entrada do nó filho e assim sucessivamente até os nós finais. Este processo pode levar ao processamento de nós simultâneos, a única dependência de execução será

o dado do nó pai, como mostrado na Figura 3.4. O final da execução por nodos pode ser guardado em uma estrutura de dados a fim de ser consumida posteriormente pelos usuários.

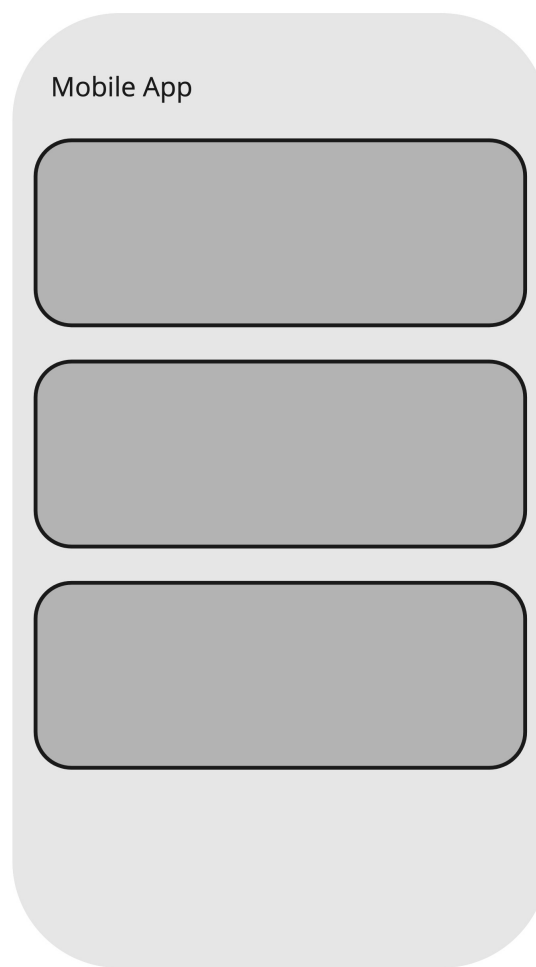
Figura 3.4: Exemplo de sequência de nodos a serem processados.



3.6 Interface com paciente

Toda interação do paciente será por meio de um aplicativo móvel, onde este poderá se inscrever ou ser inscrito em algum programa para fazer a entrada de dados, receber os mecanismos de engajamentos, leitura de textos e todas as outras possíveis interações com o fluxo previamente criado. O objetivo da aplicação móvel é facilitar a interação do HCP com o paciente, proporcionando uma abstração do fluxo projetado com intenção de melhoria da prática da conduta dos HCPs. A fim de facilitar o entendimento do paciente, é necessário propor uma estrutura onde as tarefas poderão ser exibidas para os pacientes em uma ordem cronológica de objetivos a serem cumpridos previamente definido pelos nodos e fluxos criados pelo profissional da área da saúde, assim como mostrado na Figura 3.5. Toda a responsabilidade de criação deste mural com as tarefas e mecanismos de engajamento ficará com o backend e API, a responsabilidade do software *mobile* fica em criar e manter o engajamento do usuário com o fluxo.

Figura 3.5: Proposta software mobile.



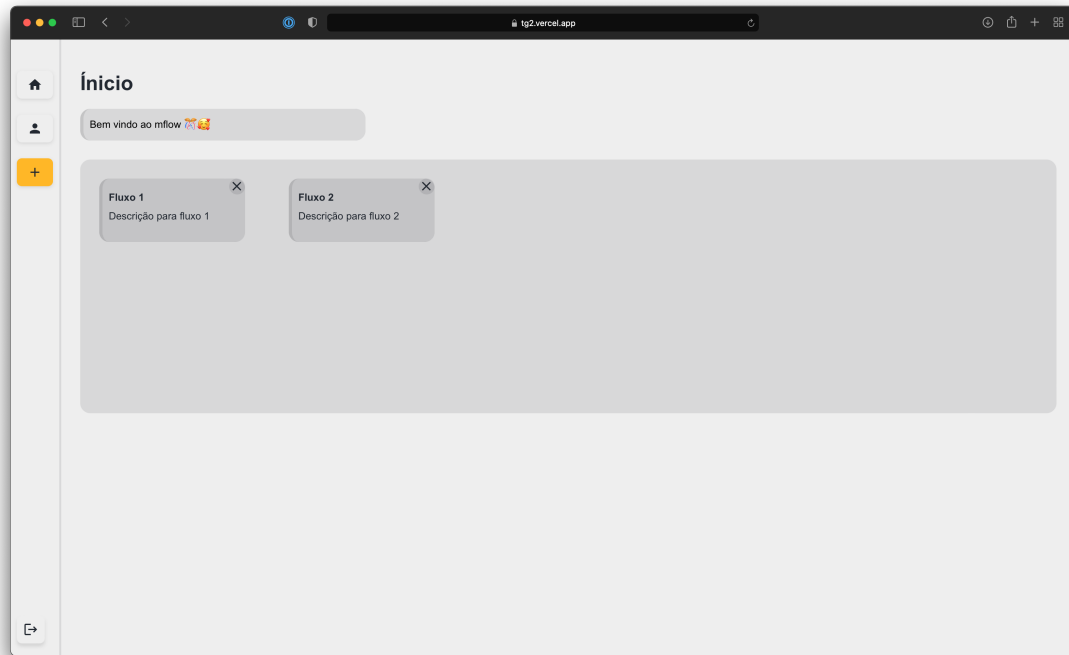
4 IMPLEMENTAÇÃO

A partir da proposta de solução descrita, foram definidos alguns requisitos para o desenvolvimento dos sistemas: todos eles foram versionados usando uma plataforma de versionamento, a linguagem base escolhida foi o Javascript e o padrão de comunicação entre os serviços é REST feita por meio do HTTP. Por fim, por decisão de projeto e com as vantagens de componentização e modularização do FBP, foi utilizado também o conceito de *Server Driven UI*, onde toda a interface, tanto do sistema de VPL quanto da aplicação *mobile* é controlado pelo backend. A preferência destas tecnologia foi adotada afim de facilitar o desenvolvimento da plataforma, pois com uma mesma linguagem será possível o desenvolvimento tanto do frontend, backend e da aplicação móvel, tornando a manutenção e evolução da plataforma mais acessível.

4.1 Ferramenta FBP e VBP

O desenvolvimento do software de frontend (Figura 4.1) para a criação dos fluxos baseado em linguagem visual é realizado através do React, escolhido como *framework* para facilitar o desenvolvimento. Esta é uma biblioteca de Javascript com foco em criar interfaces Web. Todos os componentes que compõem esse ambiente foram desenvolvidos modularmente pensados para o conceito de *Server Driven UI*, desta maneira podem ser reutilizados. Este sistema é constituído por três grandes partes: o mecanismo de autenticação com suas páginas, o mecanismo de programação visual e as estruturas e aplicações da programação de fluxos.

Figura 4.1: Dashboard com fluxos.



4.1.1 Estruturas e programação de fluxos

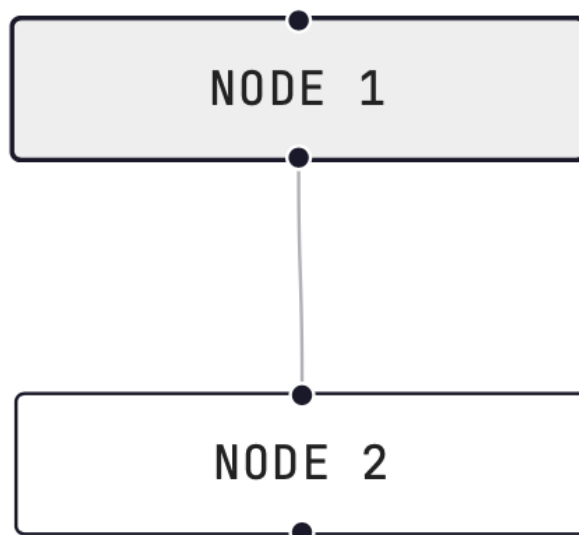
Para garantir a coesão dos dados e informações foi necessário criar uma estrutura de dados que representasse a lógica, hierarquia e conexões dos fluxos. O modelo escolhido para essa representação foi um modelo de grafos constituído por os nodos e as conexões. Os nodos seriam como "caixas pretas" onde toda lógica de execução é programada previamente pelo backend, tornando necessária somente a descrição das entradas. As conexões representam a linha de execução do fluxo e elas são independentes ao processamento e aos dados.

A descrição de um nodo é basicamente composta por um identificador, tipo do nodo, quantidade de entrada de conexões, quantidade de saída de conexões e, em alguns casos, interface (descriptor) de entrada de dados. Esse comportamento pode ser generalizado e repetido para todos os possíveis nodos já que o frontend não conhece os mecanismos internos do nodo e a lógica de execução fica no backend. Por conta da abstração total do FBP o frontend implementa somente um nodo genérico e todos os requisitos dos nodos são enviados pelo backend e renderizados "on the fly" pelo frontend. Quando necessário a entrada de dados é salva em um *payload* dentro da hierarquia da estrutura de dados do nodo. A descrição de uma conexão é composta por um identificador, um nodo fonte e um

nodo alvo. Essa estrutura, desconhece qualquer tipo de informação do nodo fonte ou do nodo alvo, seu principal papel é fazer a interligação de dois nodos. Tendo como base a estrutura proposta, alguns métodos de computação são possíveis de serem implementados somente manipulando a estrutura de dados, tal como nodos raízes e nodos folhas.

A estrutura do grafo é representada somente por um *Array*, no qual é possível encontrar outras duas estruturas de dados do nodo e das conexões. Esse padrão cria uma flexibilidade para representar a estrutura de grafos em estrutura de dados. Novos nodos podem ser adicionados na estrutura assim como novas conexões. Deste modo, cria uma facilidade computacional e de estrutura de dados, onde é possível guardar os nodos e suas conexões, sendo possível criar todos os tipos de interligações entre os nodos. Podemos exemplificar com um fluxo de dois nodos e uma conexão como mostrado na Figura 4.2.

Figura 4.2: Exemplo de fluxo com dois nodos e uma conexão.



Listing 1: Exemplo de estrutura de dados para dois nodos e uma conexão.

```
1  [
2    { "id": "1", "type": "GENERIC_INPUT" },
3    { "id": "2", "type": "GENERIC_INPUT" },
4    { "id": "e1-2", "source": "1", "target": "2" }
5  ]
```

4.1.2 Mecanismo de programação visual

O objetivo da programação visual é facilitar a manipulação dos nodos e conexões mencionadas acima, para isso, esse software une três grandes conceitos em uma arquitetura de *Server Side UI*, *FBP* e *VPL*. A junção destes mecanismos cria uma grande habilidade de manipulação de interfaces de fluxos e conexões para o frontend. Como cada nodo é uma "caixa preta", tendo conhecimento somente da entrada de dados, é possível generalizar todos os nodos e criar uma implementação dinâmica onde todos os nodos implementam um nodo genérico, sendo que este pode ser totalmente controlado pelo backend. A implementação de um nodo para o frontend é genérica e, para especificar suas características que irão distinguir um nodo de outro, é necessário criar um esquema de representação do nodo, com dados que serão usados para a renderização da interface e entrada de dados. Deste modo, ao iniciar a criação de um novo fluxo, o frontend tem como pré requisito receber do backend uma lista dos esquemas dos nodos que podem ser renderizados, possibilitando a criação de diferentes tipos de nodos alterando apenas os esquemas de definições dos nodos, já que toda a lógica de execução fica no backend acessível por meio da API. Sendo assim, uma listagem da biblioteca de nodos é exibida no momento em que o usuário entra na tela de criação de um novo fluxo.

O mecanismo de criação de fluxo foi planejado para ser dinâmico, permitindo que o usuário interaja diretamente com o fluxo, seja arrastando os nodos para criar um arranjo de posições, clicar e puxar uma ligação entre os dois nodos, ou até mesmo arrastando um novo nodo da biblioteca para a interface dos fluxos, como é possível ver na Figura 4.3. A lógica do fluxo pode ser configurada pelo usuário selecionando, arrastando e conectando os nodos.

Figura 4.3: Funcionalidade arraste e puxe para adicionar novo nodo.



A página de criação de fluxo (Figura 4.4) foi desenvolvida para comportar diversos nodos e conexões pois estes não são fixos na tela e podem mudar de posição conforme a preferência do usuário. Toda a estrutura de nodos e conexões compõem a estrutura de dados mencionada na seção anterior, deste modo, quando um nodo é arrastado da biblioteca para o quadro, uma nova entrada no *Array* da estrutura de dados é inserida para representar a adição do novo nodo, assim como uma nova conexão entre dois nodos. Caso um nodo implemente algum tipo de interface de entrada é possível interagir com o nodo clicando em cima do mesmo, abrindo uma janela com as características especificadas pelo nodo, como mostrado na Figura 4.5. Após toda a customização do fluxo e interligações de nodos, a estrutura de dados é enviada para o backend onde irá ser executada.

Figura 4.4: Página de construção de fluxos.

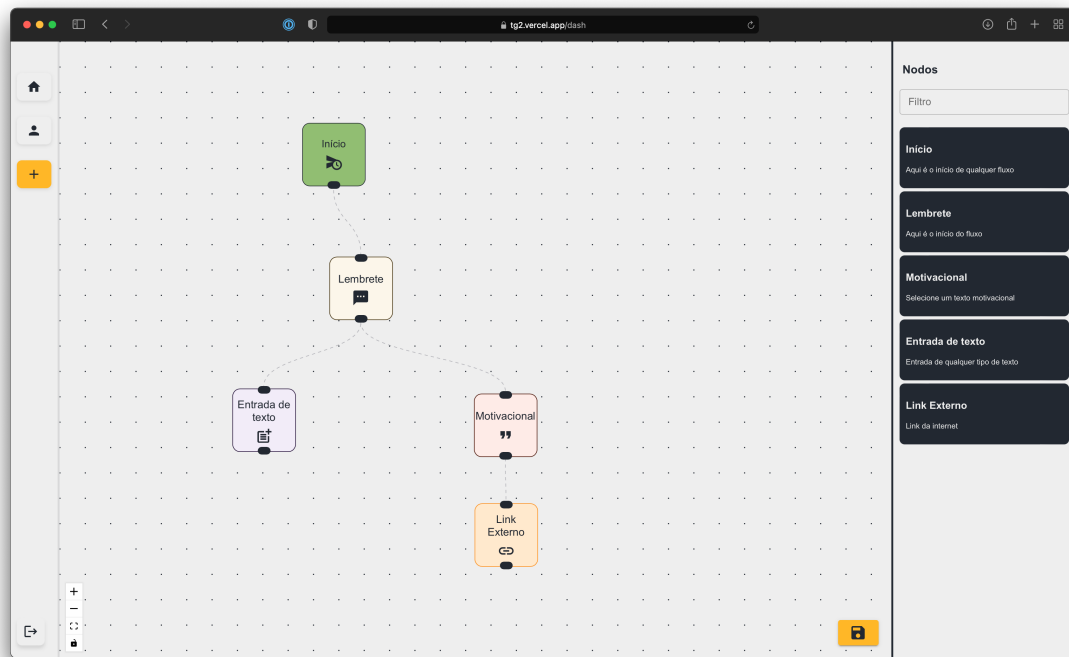


Figura 4.5: Edição de metadado de um nodo.

The screenshot shows the same flow builder interface as in Figure 4.4, but with a modal dialog open for editing the metadata of a 'Lembrete' node. The dialog has a title 'Lembrete' and a subtitle 'Aqui é o início do fluxo'. It contains three input fields: the first is labeled '[BOARD] - Nome' with the prompt 'Do que se trata o lembrete?'; the second is labeled '[BOARD] - Descrição' with the prompt 'Adicione a descrição do lembrete'; and the third is labeled 'lembrete' with the prompt 'escreva aqui seu lembrete'. At the bottom right of the dialog are two buttons: 'SALVAR' (Save) and 'FECHAR' (Close). The background of the interface is dimmed.

4.2 Backend e APIs

O desenvolvimento do backend foi realizado por meio da plataforma NodeJS que possibilita a execução de códigos Javascript. O principal objetivo do backend é expor algumas comunicações com o frontend, aplicação mobile e também ficar responsável pela implementação dos nodos e execução dos fluxos. Para desempenhar esse papel foi utilizado o ExpressJS para facilitar a criação do servidor e também foi escolhido o MongoDB como banco de dados. A comunicação entre o frontend, aplicação móvel e o backend é realizada por meio da arquitetura Rest, contendo três grandes domínios: comunicação para aplicação frontend, comunicação para aplicação móvel e autenticação de usuários.

As rotas que contemplam a comunicação com o frontend tem como principais responsabilidades garantir a comunicação entre as aplicações, tendo como rotas: listagem de todos os esquemas de nodos da biblioteca, armazenamento de estado da criação dos fluxos, listagem de fluxos do usuário, dentre outras. Já as rotas da aplicação móvel têm como objetivo garantir a listagem de tarefas a serem executadas pelo usuário, o reconhecimento das execução das tarefas e em alguns casos, a entrada de dados dos nodos. A autenticação dos usuários também é de responsabilidade do backend, onde são implementadas rotas de criação e recuperação de usuários, tais como a autorização e identificação de requisições dos usuários para o backend.

4.2.1 Descrição do nodo para frontend

O nodo para o frontend é genérico e toda a sua descrição parte do backend, tendo como requisito a implementação de uma interface que garante as especificidades de cada nodo. Para garantir a interface, foi criado um protocolo que estabelece um esquema de comunicação entre a execução lógica do nodo e suas entradas. Este mesmo esquema é enviado para o frontend renderizar os nodos genéricos com os atributos específicos de cada nodo. Um esquema de nodo é representado por uma estrutura notação de objeto em Javascript (do inglês, JSON) que contém algumas chaves, tais como: identificador, tipo, nome, descrição, ícone, estilo *css* do nodo, descritor de entradas e descritor de alças. Destas chaves, destacam-se: descritor de entradas e descritor de alças. O descritor de entradas é responsável por garantir a conexão da entrada dos dados na criação com a execução, já o descritor de alças é responsável por criar e estabelecer as possíveis conexões de entrada ou saída com outros nodos. Como exemplo, o código abaixo representa o esquema de um

nodo que tem como entrada um campo de texto e duas alças uma de entrada e outra de saída.

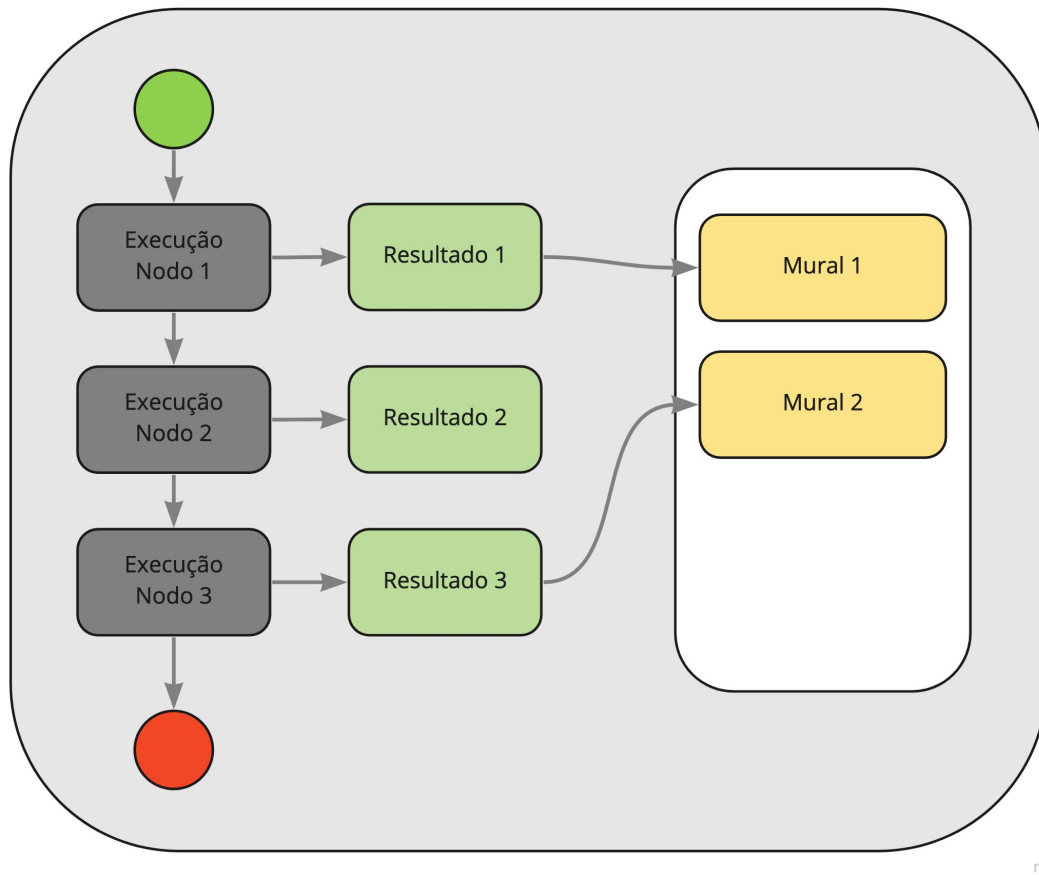
Listing 2: Exemplo de esquema de um nodo com uma entrada e duas alças.

```
1  {
2      "id": 4,
3      "type": "GENERIC_NODE",
4      "data": {
5          "type": "EXTERNAL_LINK",
6          "label": "Link Externo",
7          "description": "Link da internet",
8          "icon": "link",
9          "style": {
10             "backgroundColor": "#ffe9cc",
11             "borderColor": "#ff9000"
12         },
13         "inputFields": [
14             {
15                 "slug": "link",
16                 "name": "link",
17                 "description": "link da internet",
18                 "type": "RAW_TEXT_INPUT"
19             }
20         ],
21         "handles": [
22             {
23                 "type": "target",
24                 "id": "four_target",
25             },
26             {
27                 "type": "source",
28                 "id": "four_source",
29             }
30         ]
31     }
32 }
```


4.2.2 Execução e resultado da execução do fluxo

A implementação da lógica de cada nodo é independente de lógicas externas, tendo como único pré-requisito as entradas obrigatórias, estabelecidas no esquema do nodo para execução do mesmo. O nodo cria uma dependência forte com a entrada e depois pode ser executado sem depender de outros estados da aplicação, permitindo que o resultado da execução possa ser utilizado como entrada para um outro nodo secundário. A partir do momento em que um nodo é acionado sua execução começa de forma recursiva, executando o primeiro nodo pai até o último nodo filho. O resultado da execução de um nodo depende muito da lógica criada por ele, porém, toda execução conta com um resultado e este pode criar também um item no mural. Os itens no mural são aqueles consumidos pelos usuários (pacientes) na aplicação móvel e podem ser tanto itens de leitura quanto de execução, assim como é possível observar na Figura 4.6. Um exemplo de fluxo pode ser o lembrete de alguma atividade com apenas um nodo, o resultado da execução deste nodo será o lembrete em si e um painel também será criado com o lembrete para o usuário. O mecanismo de acionamento do fluxo foi desenvolvido pensando em recorrência e todos os fluxos possuem cronômetros que são iniciadores de fluxos, podendo haver mais de um para o mesmo grafo. Os cronômetros têm como mecanismos de ativação expressões em comando de sistema operacional *unix-like* (do inglês, CRON) que ficam salvas no banco de dados e há uma rotina de monitoramento de execução que periodicamente recupera os fluxos que devem ser executados e inicia o processamento.

Figura 4.6: Lógica da execução de nodos com mural.



miro

Para entender o mecanismo e as ordens de execução, foi criado um exemplo hipotético, mostrado na Figura 4.7, com quatro nodos: A, B, C e D, conectados entre si A-B, A-C, A-D e B-D. Neste exemplo, existe a dependência de D com o resultado das execuções de B e A. A execução do grafo é recursiva e após a execução do nodo raiz, o nodo A executa as conexões a1, a2 e a3 e em seguida são executados os nodos filhos B e C. Desta forma, os nodos filhos são executados quando não dependem de nenhuma entrada pendente e assim por diante, recursivamente. A ordem da execução é assíncrona e paralela sempre que possível, como demonstrado na Figura 4.8. Nem toda execução cria murais mas o resultado de uma execução pode criar um mural que posteriormente irá para o feed dos usuários. Somente o resultado da execução de C criou o Mural C e o resultado da execução de D criou o Mural D.

Figura 4.7: Fluxograma da execução dos nodos.

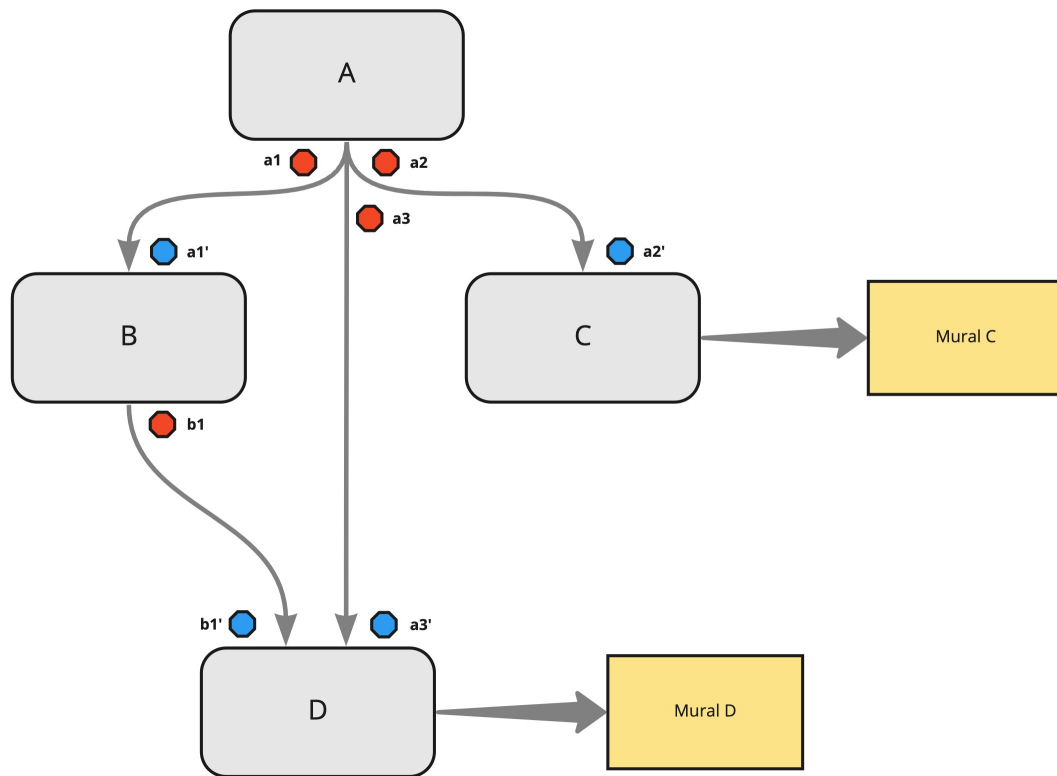
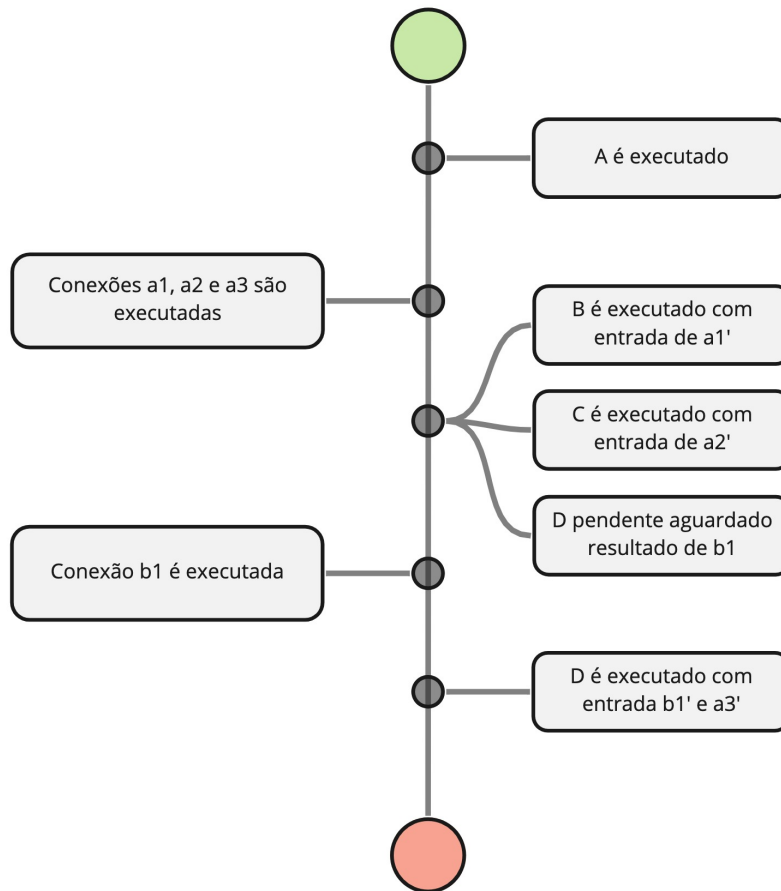


Figura 4.8: Ordem temporal da execução dos nodos.



miro

4.3 Software *Mobile*

O desenvolvimento *mobile* foi realizado utilizando o framework React Native para aplicações móveis e tem como objetivo facilitar o desenvolvimento tanto para o sistema operacional *IOS* quanto para *Android*. Uma mesma base de código é compartilhada entre os sistemas operacionais da Apple e do Google. Deste modo, foi criada uma aplicação móvel com a finalidade dos usuários interagirem com os fluxos por meio dos murais. A aplicação móvel consiste em uma área logada e uma área de dados (murais). Os fluxos são abstraídos para uma apresentação mais harmoniosa para os usuários e os resultados da execução criam murais que compõem um *feed* de tarefas exibido para os usuários, como mostrado na Figura 4.9. A exibição do *feed* e seus itens é controlada pelo backend que, por sua vez, é responsável por controlar a ordem, quantidade e tipos de tarefas por mural.

A tarefa do usuário é o resultado final da execução de um fluxo ou de um nodo.

O conceito de "conclusão" foi adicionado no software com o propósito de criar uma maior interatividade com a aplicação. Cada tipo de tarefa mostrada para o usuário pode ser marcada como concluída pelo utilizador do aplicativo, como mostra a Figura 4.10. Em alguns casos a conclusão da tarefa se dá pela adição de dados (Figura 4.11).

Figura 4.9: Mural de atividades do usuário.

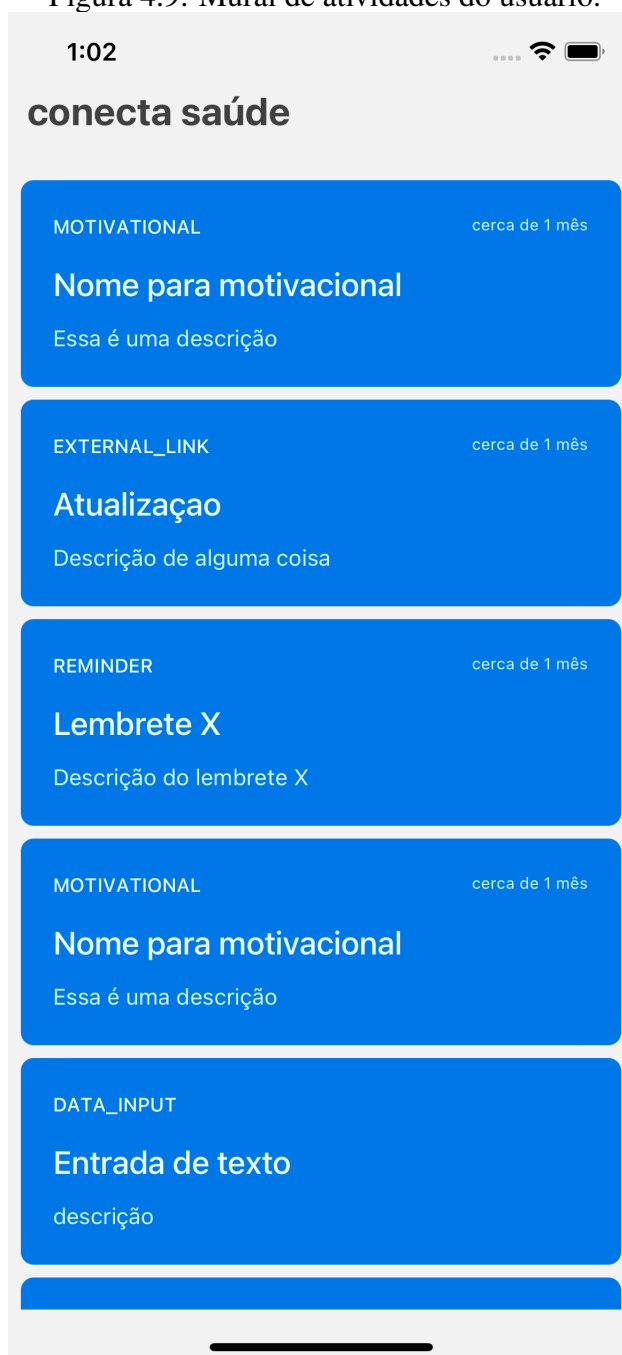


Figura 4.10: Atividade lembrete.

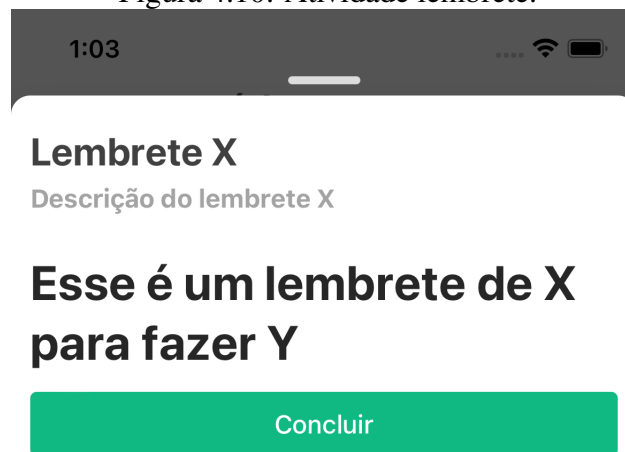


Figura 4.11: Atividade entrada de dados.



The image shows a mobile application interface for text input. At the top, there is a dark grey status bar with the time '1:03' on the left and icons for signal strength, Wi-Fi, and battery on the right. Below the status bar, the title 'Entrada de texto' is displayed in bold black text, followed by the subtitle 'descrição' in a smaller, lighter font. The main heading is 'Como você está hj?' in large, bold black text, with the subtitle 'Me diz como esta?' below it. A text input field with a light grey border contains the placeholder text 'Como você está hj?'. Below the input field is a prominent green button with the white text 'Concluir'. A thick black horizontal line is positioned at the bottom of the page, below the app's content area.

5 CONSIDERAÇÕES FINAIS

No mercado atual é possível notar o surgimento de aplicações móveis direcionadas às soluções de um público alvo cada vez mais específico. Neste âmbito, aplicativos referentes aos portadores de uma determinada doença podem aumentar a adesão dos usuários aos seus tratamentos médicos. Dessa forma, a colaboração dos profissionais de saúde é de fundamental importância no desenvolvimento destas aplicações, pois são eles que determinam os fluxos adequados a serem seguidos no tratamento de cada paciente.

Sendo assim, as aplicações móveis relacionadas à área da saúde favorecem uma maior adesão ao tratamento correto das doenças, resultando em uma melhor resposta clínica dos pacientes e, conseqüentemente, melhora da qualidade de vida e redução da morbimortalidade, promovendo melhoria na saúde coletiva. Ademais, conclui-se que a plataforma desenvolvida consegue executar com seu propósito de criação de fluxos interativos *codeless* para os profissionais de saúde, sendo assim, cumprindo com o objetivo de engajamento dos pacientes no tratamento, validando a pergunta questionada no capítulo da introdução. Contudo, observou-se a necessidade do desenvolvimento de mais funcionalidades, tal como nodo lógico para uma maior personalização dos fluxos, também existe a demanda de métricas de adesão do paciente. Por outro lado, foi possível verificar a validade do uso de FBP e VPL para construção de sistemas onde existem os requisitos de transferência de dados para execução entre os módulos do sistemas. Desta forma, atingindo um patamar de possibilidade de programação sem codificação.

Portanto, este trabalho viabilizou inicialmente os mecanismos básicos de engajamento apresentados, por meio do desenvolvimento de uma plataforma interativa, customizável e escalável, para que possam ser utilizados amplamente a fim da promoção de saúde.

Propõe-se para trabalhos futuros a inclusão de nodos de lógicas, com o objetivo de poder fazer condicionais na linha de execução dos fluxos. Deste modo, será possível controlar a execução com condicionais, podendo por exemplo customizar e individualizar cada vez mais o fluxo baseado em fatores externos, como: entrada de dados dos usuários, resultado de exames e outros. Além disso, também é sugerido a possibilidade da criação de nodos baseados em fluxos com outros diversos nodos, deste modo, criando abstrações para possíveis fluxos em comum entre os tratamentos e possibilitando o reúso e compartilhamento. Por fim, é necessário fazer a validação do aumento e eficácia de engajamento nos tratamentos de saúde propostos pelos HCPs, proporcionado pelos sistemas

de software apresentado nesse trabalho.

REFERÊNCIAS

- AITKEN, M.; CLANCY, B.; NASS, D. The growing value of digital health: evidence and impact on human health and the healthcare system. **IQVIA Institute for Human Data Science**, v. 1, 2017.
- ALEXANDROVA, S.; TATLOCK, Z.; CAKMAK, M. Roboflow: A flow-based visual programming language for mobile manipulation tasks. In: **IEEE. 2015 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.], 2015. p. 5537–5544.
- ALI, E. E.; CHEW, L.; YAP, K. Y.-L. Evolution and current status of mhealth research: a systematic review. **BMJ Innovations**, BMJ Specialist Journals, v. 2, n. 1, 2016.
- ARMANASCO, A. A. et al. Preventive health behavior change text message interventions: a meta-analysis. **American journal of preventive medicine**, Elsevier, v. 52, n. 3, p. 391–402, 2017.
- AUNGST, T. D. Medical applications for pharmacists using mobile devices. **Annals of Pharmacotherapy**, SAGE Publications Sage CA: Los Angeles, CA, v. 47, n. 7-8, p. 1088–1095, 2013.
- BASHSHUR, R. et al. The taxonomy of telemedicine. **Telemedicine and e-Health**, Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA, v. 17, n. 6, p. 484–494, 2011.
- BERTOCCO, M. et al. A client-server architecture for distributed measurement systems. **IEEE transactions on instrumentation and measurement**, IEEE, v. 47, n. 5, p. 1143–1148, 1998.
- BONOTO, B. C. et al. Efficacy of mobile apps to support the care of patients with diabetes mellitus: a systematic review and meta-analysis of randomized controlled trials. **JMIR mHealth and uHealth**, JMIR Publications Inc., Toronto, Canada, v. 5, n. 3, p. e4, 2017.
- BOSHERNITSAN, M.; DOWNES, M. S. **Visual programming languages: A survey**. [S.l.]: Citeseer, 2004.
- BURNETT, M. M.; MCINTYRE, D. W. Visual programming. **COMPUTER-LOS ALAMITOS-**, IEEE INSTITUTE OF ELECTRICAL AND ELECTRONICS, v. 28, p. 14–14, 1995.
- CECHETTI, N. P. et al. Developing and implementing a gamification method to improve user engagement: A case study with an m-health application for hypertension monitoring. **Telematics and Informatics**, v. 41, p. 126–138, 2019. ISSN 0736-5853. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0736585318312656>>.
- CIÊNCIA, B. M. da; TECNOLOGIA, T. Síntese de evidências para políticas de saúde: adesão ao tratamento medicamentoso por pacientes portadores de doenças crônicas. In: . [S.l.: s.n.], 2016.

- COTTON, V.; PATEL, M. Gamification use and design in popular health and fitness mobile applications. **American Journal of Health Promotion**, v. 33, p. 089011711879039, 07 2018.
- CUI, M. et al. T2dm self-management via smartphone applications: a systematic review and meta-analysis. **PloS one**, Public Library of Science San Francisco, CA USA, v. 11, n. 11, p. e0166718, 2016.
- CULLER, D. E. Dataflow architectures. **Annual review of computer science**, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 1, n. 1, p. 225–253, 1986.
- DIVALL, P.; CAMOSSO-STEFINOVIC, J.; BAKER, R. The use of personal digital assistants in clinical decision making by health care professionals: a systematic review. **Health informatics journal**, Sage Publications Sage UK: London, England, v. 19, n. 1, p. 16–28, 2013.
- DUGAS, M. et al. Individual differences in regulatory mode moderate the effectiveness of a pilot mhealth trial for diabetes management among older veterans. **PloS one**, Public Library of Science San Francisco, CA USA, v. 13, n. 3, p. e0192807, 2018.
- ERWIG, M.; MEYER, B. Heterogeneous visual languages-integrating visual and textual programming. In: **Proceedings of Symposium on Visual Languages**. [S.l.: s.n.], 1995. p. 318–325.
- FUNKHOUSER, T. A. Ring: A client-server system for multi-user virtual environments. In: **Proceedings of the 1995 symposium on Interactive 3D graphics**. [S.l.: s.n.], 1995. p. 85–ff.
- GIANSANTI, D. The role of the mhealth in the fight against the covid-19: Successes and failures. In: MULTIDISCIPLINARY DIGITAL PUBLISHING INSTITUTE. **Healthcare**. [S.l.], 2021. v. 9, n. 1, p. 58.
- GOSETTO, L.; EHRLER, F.; FALQUET, G. Personalization dimensions for mhealth to improve behavior change: A scoping review. **Integrated Citizen Centered Digital Health and Social Care**, IOS Press, p. 77–81, 2020.
- GUIMARAES, T.; IGBARIA, M. Client/server system success: Exploring the human side. **Decision sciences**, Wiley Online Library, v. 28, n. 4, p. 851–876, 1997.
- HALES, S. et al. Trading pounds for points: Engagement and weight loss in a mobile health intervention. **Digital health**, SAGE Publications Sage UK: London, England, v. 3, p. 2055207617702252, 2017.
- HALL, A. K.; COLE-LEWIS, H.; BERNHARDT, J. M. Mobile text messaging for health: a systematic review of reviews. **Annual review of public health**, Annual Reviews, v. 36, p. 393–415, 2015.
- HAMARI, J. et al. Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. **Computers in Human Behavior**, v. 54, p. 170–179, 2016. ISSN 0747-5632. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S074756321530056X>>.

INTELLIGENCE, K. S. **Mobile Health (mHealth) App Market - Industry Trends, Opportunities and Forecasts to 2023**. 2017. <<https://www.researchandmarkets.com/reports/4435917/mobile-health-mhealth-app-market-industry#description>>. Accessed: 2017-09-10.

ITFORUM. **Low-code: as vantagens e desvantagens do tal código baixo**. 2020.

KHWAJA, M. et al. Personality and engagement with digital mental health interventions. In: _____. **Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization**. New York, NY, USA: Association for Computing Machinery, 2021. p. 235–239. ISBN 9781450383660. Disponível em: <<https://doi.org/10.1145/3450613.3456823>>.

KISER, K. 25 ways to use your smartphone. physicians share their favorite uses and apps. **Minnesota medicine**, v. 94, n. 4, p. 22–29, 2011.

KITSIOU, S. et al. Effectiveness of mhealth interventions for patients with diabetes: an overview of systematic reviews. **PloS one**, Public Library of Science San Francisco, CA USA, v. 12, n. 3, p. e0173160, 2017.

KRATKY, S.; REICHENBERGER, C. **Client/server development based on the apple event object model**. [S.l.]: Atlanta, 2013.

LEE, B.; HURSON, A. R. Dataflow architectures and multithreading. **Computer**, IEEE, v. 27, n. 8, p. 27–39, 1994.

MADEIRA, R. N. et al. Personalising the user experience of a mobile health application towards patient engagement. **Procedia Computer Science**, v. 141, p. 428–433, 2018. ISSN 1877-0509. The 9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2018) / The 8th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2018) / Affiliated Workshops. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1877050918318234>>.

MARCOLINO, M. S. et al. The impact of mhealth interventions: systematic review of systematic reviews. **JMIR mHealth and uHealth**, JMIR Publications Inc., Toronto, Canada, v. 6, n. 1, p. e23, 2018.

MATEO, G. F. et al. Mobile phone apps to promote weight loss and increase physical activity: a systematic review and meta-analysis. **Journal of medical Internet research**, JMIR Publications Inc., Toronto, Canada, v. 17, n. 11, p. e253, 2015.

MEIRELLES. **Brasil tem 424 milhões de dispositivos digitais em uso, revela a 31ª Pesquisa Anual do FGVcia**. 2020.

MICKAN, S. et al. Evidence of effectiveness of health care professionals using handheld computers: a scoping review of systematic reviews. **Journal of medical Internet research**, JMIR Publications Inc., Toronto, Canada, v. 15, n. 10, p. e212, 2013.

MING, L. C. et al. Mobile health apps on covid-19 launched in the early days of the pandemic: Content analysis and review. **JMIR Mhealth Uhealth**, v. 8, n. 9, p. e19796, Sep 2020. ISSN 2291-5222. Disponível em: <<https://mhealth.jmir.org/2020/9/e19796>>.

- MORRISON, J. P. Flow-based programming. In: **Proc. 1st International Workshop on Software Engineering for Parallel and Distributed Systems**. [S.l.: s.n.], 1994. p. 25–29.
- MOSA, A.; YOO, I.; SHEETS, L. A systematic review of health care apps for smartphones. **BMC Med Inform Dec Mak. Jul**, v. 12, p. 67, 2013.
- MURFIN, M. Know your apps: an evidence-based approach to evaluation of mobile clinical applications. **The journal of physician assistant education: the official journal of the Physician Assistant Education Association**, v. 24, n. 3, p. 38–40, 2013.
- OZDALGA, E.; OZDALGA, A.; AHUJA, N. The smartphone in medicine: a review of current and potential use among physicians and students. **Journal of medical Internet research**, JMIR Publications Inc., Toronto, Canada, v. 14, n. 5, p. e128, 2012.
- O'NEILL, K. M. et al. Applying surgical apps: Smartphone and tablet apps prove useful in clinical practice. **Bull Am Coll Surg**, v. 98, n. 11, p. 10–18, 2013.
- RIBEIRO, N. et al. Guidelines for a cancer prevention smartphone application: a mixed-methods study. **International journal of medical informatics**, Elsevier, v. 94, p. 134–142, 2016.
- SHEET, M. F. Pew research center: Internet. **Science & Tech**, v. 5, 2018.
- SOUSA, T. B. Dataflow programming concept, languages and applications. In: **Doctoral Symposium on Informatics Engineering**. [S.l.: s.n.], 2012. v. 130.
- SVOBODOVA, L. Client/server model of distributed processing. In: **Kommunikation in Verteilten Systemen I**. [S.l.]: Springer, 1985. p. 485–498.
- TAVARES, N. U. L. et al. Fatores associados à baixa adesão ao tratamento farmacológico de doenças crônicas no brasil. **Revista de Saúde Pública**, SciELO Public Health, v. 50, p. 10s, 2016.
- WADGE, W. W.; ASHCROFT, E. A. et al. **Lucid, the dataflow programming language**. [S.l.]: Academic Press London, 1985. v. 303.
- WALLACE, S.; CLARK, M.; WHITE, J. 'it's on my iphone': attitudes to the use of mobile computing devices in medical education, a mixed-methods study. **BMJ open**, British Medical Journal Publishing Group, v. 2, n. 4, 2012.
- WHITEHEAD, L.; SEATON, P. The effectiveness of self-management mobile phone and tablet apps in long-term condition management: a systematic review. **Journal of medical Internet research**, JMIR Publications Inc., Toronto, Canada, v. 18, n. 5, p. e97, 2016.
- YOO, J.-H. The meaning of information technology (it) mobile devices to me, the infectious disease physician. **Infection & chemotherapy**, Korean Society of Infectious Diseases, v. 45, n. 2, p. 244, 2013.
- ZHANG, H. Architecture of network and client-server model. **arXiv preprint arXiv:1307.6665**, 2013.