

**UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DE CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

GABRIEL RICARDO TUBIAS ROCHA

**DESENVOLVIMENTO DE APLICATIVO PARA AUXÍLIO NO
TRATAMENTO FISIOTERAPÊUTICO A DISTÂNCIA**

**CAXIAS DO SUL
2019**

GABRIEL RICARDO TUBIAS ROCHA

**DESENVOLVIMENTO DE APLICATIVO PARA AUXÍLIO NO
TRATAMENTO FISIOTERAPÊUTICO A DISTÂNCIA**

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação da Universidade de Caxias do Sul, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador Me. Alexandre Erasmo Krohn
Nascimento

**CAXIAS DO SUL
2019**

GABRIEL RICARDO TUBIAS ROCHA

**DESENVOLVIMENTO DE APLICATIVO PARA AUXÍLIO NO
TRATAMENTO FISIOTERAPÊUTICO A DISTÂNCIA**

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação da Universidade de Caxias do Sul, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovado(a) em

Banca Examinadora

Prof. Me. Alexandre Erasmo Krohn Nascimento
Universidade de Caxias do Sul - UCS

Prof. Me. Marcos Eduardo Casa
Universidade de Caxias do Sul - UCS

Prof^a. Dra. Scheila de Ávila e Silva
Universidade de Caxias do Sul - UCS

RESUMO

Diversos pacientes que consultam com fisioterapeutas recebem recomendações de exercícios fisioterápicos, que são atividades dadas pelos fisioterapeutas a seus pacientes, e as motivações por trás da escolha de cada exercício são as mais diversas. Os exercícios recomendados também são os mais diversos, e vários deles não necessitam de algum aparelho especial para serem executados, nem necessitam também de acompanhamento físico. Esse tipo de exercício, combinado com pacientes que possuem dificuldade de se deslocar até a clínica, despertou uma ideia de melhoria nesse processo. Este trabalho propõe o desenvolvimento de uma solução digital, através de um aplicativo, que permita aos fisioterapeutas recomendar com mais facilidade os exercícios aos seus pacientes, e que esses possam visualizar os exercícios recomendados também em seu aplicativo, onde estiverem. A proposta foi desenvolvida com base na metodologia de desenvolvimento ágil Iconix, utilizando-se de alguns de seus artefatos. Desta forma a documentação necessária para o desenvolvimento da solução proposta está redigida, sendo o próximo passo desenvolver de fato a solução com base na documentação aqui proposta.

Palavras-chave: Aplicativo. Fisioterapia. Gerenciamento de Conteúdo.

LISTA DE ILUSTRAÇÕES

Figura 1 – Participação no Mercado de Sistema Operacional Móvel no Mundo (Jan - Abr 2019)	21
Figura 2 – Processo Iconix	30
Figura 3 – Arquitetura da Solução	45
Figura 4 – Topologia	48
Figura 5 – Casos de Uso	50
Figura 6 – Modelo de Domínio	58
Figura 7 – Acesso ao aplicativo	59
Figura 8 – Criação e edição da conta	60
Figura 9 – Menus do aplicativo	60
Figura 10 – Fluxo para um paciente	61
Figura 11 – Questionário	61
Figura 12 – Visualização de exercícios para um fisioterapeuta	62
Figura 13 – Edição de exercícios para um fisioterapeuta	62
Figura 14 – Protocolos para um fisioterapeuta	63
Figura 15 – Edição de protocolos para um fisioterapeuta	63
Figura 16 – Pacientes	64
Figura 17 – Acesso ao aplicativo	65
Figura 18 – Criação e edição da conta	65
Figura 19 – Menus	66
Figura 20 – Fluxo para um paciente	66
Figura 21 – Questionário	66
Figura 22 – Exercícios para um fisioterapeuta	67
Figura 23 – Protocolos para um fisioterapeuta	67
Figura 24 – Pacientes	67
Figura 25 – Cronograma	69

LISTA DE QUADROS

Quadro 1 – Caso de Uso 01 - Criar Conta	51
Quadro 2 – Caso de Uso 02 - Acessar Conta	52
Quadro 3 – Caso de Uso 03 - Recuperar senha	53
Quadro 4 – Caso de Uso 04 - Cadastrar Exercícios	54
Quadro 5 – Caso de Uso 05 - Criar Protocolos	55
Quadro 6 – Caso de Uso 06 - Recomendar Protocolos	56
Quadro 7 – Caso de Uso 07 - Visualizar protocolos	56
Quadro 8 – Caso de Uso 08 - Visualizar exercício	57
Quadro 9 – Caso de Uso 09 - Responder questionário	57

LISTA DE TABELAS

Tabela 1 – Requisitos Funcionais	49
Tabela 2 – Requisitos Não Funcionais	50

SUMÁRIO

1	INTRODUÇÃO	15
2	OBJETIVOS	17
2.1	OBJETIVO GERAL	17
2.2	OBJETIVOS ESPECÍFICOS	17
3	METODOLOGIA	19
4	REFERENCIAL TEÓRICO	21
4.1	DESENVOLVIMENTO DE SOFTWARE PARA DISPOSITIVOS MÓVEIS	21
4.1.1	Desenvolvimento nativo	22
4.1.2	Desenvolvimento multiplataforma	24
4.1.2.1	React Native	26
4.1.2.2	Flutter	28
4.2	METODOLOGIA DE DESENVOLVIMENTO ÁGIL ICONIX	30
4.3	ARMAZENAMENTO E DISPONIBILIZAÇÃO DE INFORMAÇÕES	31
4.3.1	Armazenamento Local	31
4.3.2	Armazenamento em Servidor Próprio	32
4.3.3	Armazenamento em Nuvem	33
4.3.3.1	Firebase	35
4.3.3.2	Back4App	36
4.4	SISTEMA DE GERENCIAMENTO DE CONTEÚDO	37
4.4.1	Definição de Sistema de Gerenciamento de Conteúdo	37
4.4.2	Características de Softwares Gerenciadores de Conteúdo	38
4.4.3	Exemplos de Softwares Gerenciadores de Conteúdo	39
4.5	FISIOTERAPIA	40
4.5.1	Exercícios fisioterápicos	41
4.5.2	Acompanhamento do Paciente	41
4.5.3	Tecnologização na Fisioterapia	42
4.5.4	Avaliação de Resultados de Tratamentos	43
5	PROPOSTA DE SOLUÇÃO	45
6	PROJETO	47
6.1	TOPOLOGIA DO SOFTWARE PROPOSTO	48
6.2	REQUISITOS	49
6.2.1	Requisitos Funcionais	49

6.2.2	Requisitos Não Funcionais	50
6.3	CASOS DE USO	50
6.4	MODELO DE DOMÍNIO	58
6.5	PROTÓTIPOS DE TELA	59
6.5.1	Acesso	59
6.5.2	Dados Pessoais	60
6.5.3	Menus	60
6.5.4	Fluxo do Paciente	61
6.5.5	Questionário	61
6.5.6	Exercícios na visão de um fisioterapeuta	62
6.5.7	Protocolos na visão de um fisioterapeuta	63
6.5.8	Busca de Pacientes	64
6.6	DIAGRAMAS DE ROBUSTEZ	65
7	CONSIDERAÇÕES PARCIAIS	69
7.1	CRONOGRAMA	69
	REFERÊNCIAS	71

1 INTRODUÇÃO

A fisioterapia é uma área da saúde que atende necessidades físicas, oferecendo tratamento através de exercícios fisioterápicos. A pessoa que necessitar usufruir desses serviços, geralmente deverá se deslocar até um consultório ou local de atendimento de um fisioterapeuta, para que o mesmo possa lhe avaliar e assim poder lhe fazer as recomendações adequadas.

Porém, a necessidade do paciente geralmente não é alterada após a primeira consulta, e as consultas subsequentes são sempre realizadas para tratar da necessidade inicial. Dependendo da necessidade que se possui, as consultas podem servir apenas para o fisioterapeuta recomendar alguns exercícios para o paciente realizar, seja no consultório ou em outro lugar.

É possível que a necessidade do paciente seja o tratamento de alguma dor ou uma dificuldade de locomoção por exemplo, e mesmo assim ele precisa se deslocar até a clínica para poder receber uma nova lista de exercícios, geralmente impressa, para poder tê-la como orientação e seus exercícios realizar, o que pode ser uma tarefa muito difícil em decorrência de restrições físicas.

Analisando a situação por esse olhar propõe-se o desenvolvimento de um aplicativo que venha auxiliar no tratamento fisioterapêutico a distância, por onde os fisioterapeutas consigam ter acesso e incrementar uma lista de exercícios para depois recomendá-los aos seus pacientes, que por sua vez poderão acessar essas recomendações no mesmo aplicativo em seus dispositivos, onde estiverem.

O aplicativo visa dar mais qualidade de vida tanto aos fisioterapeutas quanto aos seus pacientes, oferecendo uma base de dados colaborativa de exercícios fisioterápicos para os fisioterapeutas, e possibilitando o paciente a ter acesso a recomendação do fisioterapeuta sem a necessidade de se deslocar fisicamente até o consultório ou similar.

Para concretização dessa solução, após uma pesquisa bibliográfica, se desenvolverá um aplicativo híbrido, para Android e iOS, que são os maiores detentores do mercado de sistemas de celular, com armazenamento de informações em nuvem e através da metodologia de desenvolvimento ágil Iconix, metodologia esta que proporciona uma documentação enxuta mas relevante quando aplicada. A solução também coletará periodicamente informações sobre a satisfação de seus usuários pacientes para com o aplicativo.

2 OBJETIVOS

As seções 2.1 e 2.2 a seguir se referem aos objetivos do projeto proposto.

2.1 OBJETIVO GERAL

Desenvolver uma solução digital para auxílio na execução de exercícios fisioterapêuticos.

2.2 OBJETIVOS ESPECÍFICOS

- a) Projetar um aplicativo que auxilie fisioterapeutas na criação de conteúdos a ser usados pelos pacientes;
- b) Projetar o aplicativo para que auxilie os pacientes na execução de exercícios fisioterapêuticos;
- c) Desenvolver o aplicativo projetado;
- d) Medir a satisfação dos usuários em relação ao aplicativo.

3 METODOLOGIA

A seguir está listado o passo a passo que irá ser executado no desenvolvimento desse projeto.

- a) Realizar a modelagem do software a partir dos requisitos levantados junto à clínica de fisioterapia.
- b) Efetuar o estudo das tecnologias de desenvolvimento envolvidas;
- c) Realizar o projeto do software através da metodologia de desenvolvimento ágil Iconix;
- d) Implementar a solução em forma de aplicativo com base no projeto elaborado;
- e) Medir o quanto o aplicativo colaborou na recuperação dos pacientes em tratamento na percepção dos envolvidos.

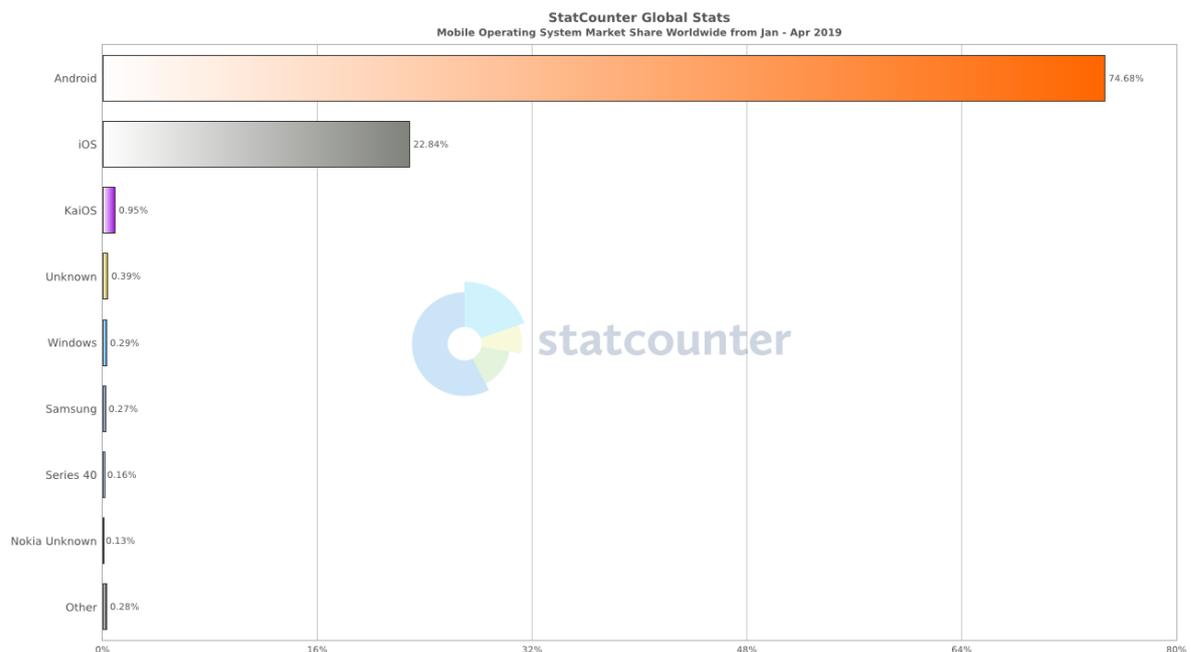
4 REFERENCIAL TEÓRICO

Neste capítulo são apresentadas as principais tecnologias e metodologias que estão relacionadas ao desenvolvimento do projeto.

4.1 DESENVOLVIMENTO DE SOFTWARE PARA DISPOSITIVOS MÓVEIS

Celulares e *tablets* são itens que passaram a ter total relevância no dia a dia das pessoas, tamanha é a versatilidade que esses pequenos dispositivos possuem. Uma oportunidade de mercado dessa grandeza com certeza é atrativa para as grandes empresas que podem atuar nesse setor, mas tratando-se do sistema operacional que a maioria dos dispositivos possuem temos 2 destaques: Android e iOS, que em abril de 2019 possuem 74,68% e 22,84% do mercado respectivamente, como mostra a Figura 1.

Figura 1 – Participação no Mercado de Sistema Operacional Móvel no Mundo (Jan - Abr 2019)



Fonte: StatCounter (2019)

O número de tarefas diversas que estes dispositivos podem realizar pode ser observado através da quantidade de aplicativos que estão publicados nas respectivas lojas de distribuição, aplicativos estes que vão desde jogos para entretenimento até ferramentas que auxiliam na gestão de um empresa. De acordo com AppBrain (2019) existem 2.603.349 aplicativos disponíveis para o sistema operacional Android em 21 de abril de 2019, e conforme a Steel Media (2019) são 3.344.967 os aplicativos que estão disponíveis para o sistema operacional iOS na mesma data.

Através de Gargenta (2010) pode-se observar que o sistema operacional Android pertence a empresa Google, é de código aberto e seus aplicativos são escritos na linguagem de programação Java. Já o sistema operacional iOS pertence a empresa Apple, e seus aplicativos podem ser escritos em Objective-C ou Swift, como vemos em Sinicki (2016).

Com estas diferenças apontadas, será considerado para este estudo as seguintes abordagens para desenvolvimento de aplicativos móveis:

- a) Desenvolvimento nativo;
- b) Desenvolvimento multiplataforma.

As duas abordagens são descritas com mais detalhes nas seções 4.1.1 e 4.1.2 do presente capítulo.

4.1.1 Desenvolvimento nativo

Desenvolver um aplicativo de forma nativa significa escrever uma aplicação exclusiva para determinada plataforma como visto em Rouse (2018b), também supondo o uso das linguagens de programação determinadas pelas detentoras das plataformas nas quais os aplicativos irão funcionar.

De acordo com Garbade (2018) podemos notar algumas vantagens na escolha do desenvolvimento nativo, que são:

- a) Alto desempenho;

Isto se dá devido a facilidade que o aplicativo nativo possui de se comunicar com as funcionalidades internas do sistema operacional e do dispositivo como mostra Garbade (2018), funcionalidades essas como microfone e câmera por exemplo. Tendo também acesso a novas funcionalidades, e melhorias sobre as preexistentes, no momento em que essas forem disponibilizadas.

b) Interface mais agradável;

De acordo com Garbade (2018) a interface é percebida como mais agradável pelos usuários pelo fato de que os aplicativos nativos se utilizam dos padrões de interface já padronizados e disponibilizados pelo sistema. Assim, os usuários que já estão acostumados a usar o sistema e seus aplicativos, reconhecem o novo aplicativo como algo mais familiar, algo que ele já esteja habituado, tornando mais intuitivo seu uso.

c) Melhor posicionamento nas lojas de aplicativos.

O posicionamento nas lojas de aplicativos é afetado também pela percepção da usabilidade que o usuário teve do aplicativo, pois é com base em sua experiência que o usuário é capaz de avaliar o aplicativo. Consequentemente, por causa do bom desempenho e da facilidade no uso, os aplicativos nativos costumam ser bem avaliados segundo Garbade (2018).

Como desvantagem dos aplicativos nativos pode se observar a sua característica de ser exclusivos para apenas uma plataforma, o que acaba os tornando mais custosos em dinheiro e tempo para desenvolvimento e manutenção, como mencionado em Garbade (2018). Caso uma empresa deseje desenvolver seu aplicativo de forma nativa, para mais de um sistema operacional, essa precisará desenvolver uma quantidade de aplicativos proporcional a quantidade de sistemas operacionais que deseja atender, gastando com isso um maior tempo e também recursos financeiros contratando programadores especializados nos diferentes sistemas operacionais.

4.1.2 Desenvolvimento multiplataforma

Desenvolver um aplicativo multiplataforma, em oposição ao desenvolvimento nativo, significa escrever uma aplicação em alguma linguagem de programação, que não necessita ser uma das determinadas pelas plataformas, e, como apontado em Rouse (2018a), ao final ter um aplicativo que funcione em mais de um sistema operacional.

Devido a ideia do desenvolvimento multiplataforma ser algo interessante, diversos *frameworks* surgiram com a proposta de ser a solução para quem quer realizar esse tipo de desenvolvimento, Rajput (2018) nos traz alguns exemplos:

- a) Ionic; (IONIC, 2019)
- b) React Native; (FACEBOOK INC., 2019)
- c) Xamarin; (MICROSOFT, 2019)
- d) Adobe PhoneGap; (ADOBE SYSTEMS INC, 2016)
- e) Flutter. (GOOGLE, 2019f)

Assim como o desenvolvimento nativo, uma abordagem de desenvolvimento multiplataforma também possui suas vantagens, como vemos em Maharana (2017):

- a) Velocidade de desenvolvimento;

Em Maharana (2017) vemos que a velocidade de desenvolvimento é superior a de desenvolvimento nativo, pois como é necessário escrever apenas código fonte na linguagem do *framework*, e esses se encarregam de fazê-lo funcionar nas diferentes plataformas, o processo todo se torna bem mais curto.

- b) Custo reduzido;

O custo se torna reduzido pelo fato de se necessitar apenas profissionais com conhecimento sobre os *frameworks* para desenvolver o aplicativo, e não um desenvolvedor para cada sistema operacional que se deseja atingir. A característica de possuir apenas códigos fontes do *framework*, e não específicos de cada plataforma, também aumenta a velocidade de correção de erros e melhorias, usando assim uma menor alocação de um desenvolvedor na tarefa.

c) Alcance.

A questão do alcance, segundo Maharana (2017), se refere ao fato do mesmo aplicativo poder atingir mais pessoas, onde ocorre que, por vezes, ao desenvolver softwares usando desenvolvimento nativo, o custo de contratar profissionais para desenvolver para plataformas menos populares não compensa. Já no desenvolvimento multiplataforma é possível atingir essas plataformas menos populares sem realizar nenhum esforço extra, pois alguns *frameworks* já oferecem essa opção por padrão.

Mas, de igual forma, Maharana (2017) também mostra algumas desvantagens desse modo de desenvolvimento:

a) Desenvolvedor e interfaces;

Essa ponto levantado por Maharana (2017) se refere ao desafio que o desenvolvedor tem de fazer seu aplicativo funcionar corretamente nas diferentes plataformas que ele tem como alvo, pois ele deve criar suas interfaces de modo que funcionem com fluidez em todas as plataformas que ele pretende atingir. A sua interface não pode funcionar no sistema Android mas não se adaptar ao sistema iOS, por exemplo.

b) Integração com a plataforma;

O desenvolvimento multiplataforma consegue tirar proveito dos recursos dos dispositivos aonde estão sendo executados, como câmera e microfone por exemplo, porém não é uma comunicação tão direta quanto a de um aplicativo nativo, o que segundo Maharana (2017) pode tornar o código um pouco mais confuso. Isso também faz com que nem sempre esteja disponível o uso, em um aplicativo multiplataforma, de um recurso nativo recém lançado, pois as vezes se faz necessário esperar pelo desenvolvimento, ou desenvolver, o *plugin* necessário para o aplicativo conseguir se comunicar com esse novo recurso.

c) Desempenho.

Essa questão da comunicação indireta com os recursos do dispositivo também influencia em outro ponto, o desempenho, pois é necessário mais esforço do dispositivo para entender o que o aplicativo quer fazer, e segundo Maharana (2017) o desempenho durante o uso pode ser bem ruim, dependendo de como for desenvolvido o *plugin* de comunicação entre o dispositivo e o *framework*.

A maneira como os *frameworks* transformam seus códigos fontes em algo que seja compreendido pelos diferentes sistemas operacionais varia, podemos observar alguns exemplos segundo NewGenApps (2018):

- a) O *framework* Ionic usa tecnologias de desenvolvimento *web* como HTML, CSS e JavaScript, e seus aplicativos funcionam de maneira similar a um site sendo executado em um navegador no celular;
- b) O *framework* React Native por sua vez, apesar de também utilizar JavaScript, transforma seu código fonte no código determinado para a plataforma, como por exemplo Java para Android e Objective-C/Swift para iOS. Dessa forma o aplicativo faz uso dos componentes nativos do dispositivo;
- c) O *framework* Flutter compila seus códigos fontes diretamente para código ARM, sendo assim executado nativamente pelas diferentes plataformas, porém fazendo uso de seus próprios componentes de interface, sem fazer uso dos componentes nativos.

As características dos *frameworks* React Native e Flutter são abordadas com mais detalhes nas seções 4.1.2.1 e 4.1.2.2.

4.1.2.1 React Native

O React Native é o *framework* desenvolvido pela empresa Facebook, apresentado em 2015, para a realização do desenvolvimento de seus aplicativos, como em Occhino (2015). Após a onda de sucesso dos *frameworks* baseados em tecnologias *web*, o React Native chegou com uma proposta similar, mas com diferenças bem significativas.

Eisenman (2015) traz algumas características do React Native, dentre as quais destacam-se sua forte influência no *framework* React para *web* e a sua capacidade de usar componentes verdadeiramente nativos para suas aplicações, essas características possibilitam que um *desenvolvedor web* seja capaz de produzir, com sua base de conhecimento, um aplicativo multiplataforma que passe a sensação de ser nativo.

Os aplicativos desenvolvidos em React Native são programados em JavaScript, mas ao contrário de outros *frameworks* que se utilizam de tecnologias *web*, o React Native não utiliza WebViews. A estratégia usada pelo React Native, contada em Occhino (2015), revela que o JavaScript é usado para realizar a comunicação com os componentes nativos do celular, criando uma espécie de ponte com o dispositivo, ponte essa que não atrapalha o processamento principal por ser assíncrona.

Outros benefícios que essa abordagem traz e que vemos em Eisenman (2015) são:

- a) O aumento do desempenho comparado ao frameworks de WebViews, pois o que se exibe na tela são os componentes nativos possuindo já todas as suas animações e transições, essas últimas que ao se tentar imitar com tecnologias web acabam não ficando tão fluídas e agradáveis;
- b) A utilização do componente nativo mais atualizado que o dispositivo disponibilizar, pelo fato do React Native apenas realizar uma chamada ao componente nativo. Quando os componentes do dispositivo sofrem alguma alteração ou melhoria acabam por simplesmente funcionar nos aplicativos feitos com React Native.

Porém, além de seus benefícios, podemos observar algumas desvantagens em Malik (2018) sobre o uso de React Native:

- a) As vezes há a necessidade de se escrever código nativo. Parece meio contraditório para um *framework* de desenvolvimento multiplataforma, mas em alguns casos mais específicos, dependendo da necessidade de negócio que se tiver, talvez seja necessário escrever algumas linhas de código nativo no aplicativo;
- b) Performance ainda inferior a de aplicativos nativos. Apesar de desempenhar melhor que os aplicativos que funcionam nas WebViews, a sua “ponte” de JavaScript ainda o faz perder um pouco de desempenho, deixando-o para trás de um aplicativo nativo;
- c) Segurança do JavaScript. A insegurança que o JavaScript passa na *web* pode ser observada também nos aplicativos feitos com React Native, ou seja, ao se fazer um aplicativo com um pouco mais de criticidade no quesito segurança recomenda-se atenção a essa questão;
- d) Desempenho do JavaScript. Essa não é uma linguagem feita para computação intensiva, o que pode fazer com que aplicativos com essa necessidade consumam mais memória, e funcionem de maneira não tão eficiente, em alguns casos.

4.1.2.2 Flutter

O Flutter é o *framework* criado pela empresa Google para o desenvolvimento de aplicativos híbridos que foi lançado no final de 2018. Como mencionado em Google (2019c), o Flutter é um kit de desenvolvimento de software para dispositivos móveis, que promete ser capaz de produzir aplicativos com alto desempenho, fiéis ao estilo de interface nativa, dando agilidade durante o desenvolvimento através de um código fonte único.

Uma das características que diferencia o Flutter dos demais *frameworks* é a sua estratégia de renderização dos componentes na tela. Ao invés de imitar os componentes com tecnologias *web*, ou usar uma ponte para se comunicar com os componentes nativos, o Flutter desenha o que está presente na tela em um Canvas do dispositivo, como vemos em Leler (2017), e é baseado nessa característica que o Flutter promete entregar animações mais fluídas e mais responsivas.

Bellinaso (2018) apresenta alguns pontos positivos de se usar o Flutter:

- a) O fato do Flutter ter o controle e desenhar os componentes o torna previsível, assim o desenvolvedor sabe que o que ele enxerga nos testes é o mesmo que será renderizado em outros dispositivos;
- b) A funcionalidade de *Hot Reload*. O Flutter permite que durante o desenvolvimento do aplicativo se consiga enxergar as alterações que estão sendo realizadas na mesma hora, pois é capaz de realizar as alterações na pré-visualização do aplicativo sem precisar reiniciá-lo completamente e sem alterar seu estado, mantendo assim os efeitos colaterais causados por interações anteriores;
- c) O uso da linguagem Dart, que é simples, completa, e que é compilada em código ARM, ao contrário do JavaScript que é interpretado enquanto está sendo executado.

Mas Bellinaso (2018) também traz alguns pontos não tão bons ao se fazer essa escolha:

- a) O Flutter possuir controle sobre os componentes renderizados, o que, apesar de ser um ponto positivo, também pode ser considerado como negativo. Caso algum sistema operacional decida por alterar a aparência de um componente, um aplicativo usando Flutter está automaticamente desatualizado, até que seja desenvolvida a nova versão do componente para o Flutter;
- b) Poucos *plugins* e componentes da comunidade. Em comparação com as comunidades de outros *frameworks*, o número de itens disponíveis para o Flutter é bem inferior. Porém deve-se considerar que o Flutter é o *framework* mais novo, e que deverão surgir mais componentes da comunidade conforme sua popularidade for aumentando;
- c) Possível duplicidade no código das interfaces. Como tudo no Flutter é considerado um *Widget*, os blocos de construção de uma interface também são, porém os *widgets* do visual de Android são *widgets* diferentes dos que são do visual de iOS, ou seja, caso o aplicativo necessite ser diferente entre as plataformas será necessário escrever a interface das duas maneiras.

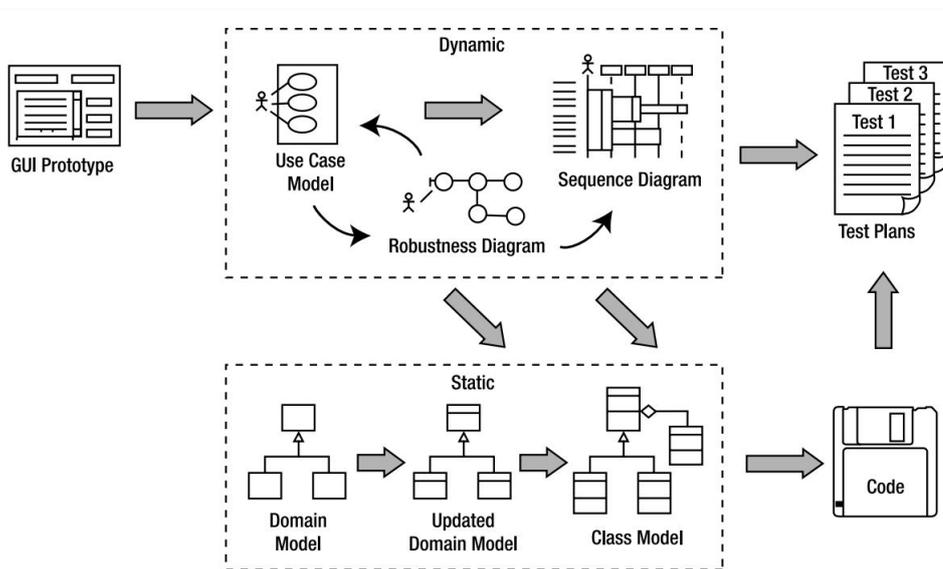
Para se realizar um bom desenvolvimento geralmente se adota uma metodologia, para que a mesma guie os passos que devem ser executados no decorrer do desenvolvimento. A Seção 4.2 aborda um pouco sobre uma metodologia de desenvolvimento ágil chamada Iconix.

4.2 METODOLOGIA DE DESENVOLVIMENTO ÁGIL ICONIX

Ao se utilizar a metodologia Iconix pode-se dizer que o que se está fazendo é “destilar os requisitos de comportamento para uma forma mais completa, mais precisa e menos ambígua. [...] estamos desambiguando os requisitos de comportamento e, em seguida, direcionando o design de software para esses requisitos não ambíguos” (ROSENBERG, 2005, p. 42, tradução nossa).

A Iconix é uma metodologia que propõe algumas etapas para o desenvolvimento da solução, essas etapas podem ser vistas na figura 2 abaixo.

Figura 2 – Processo Iconix



Fonte: Rosenberg (2005, p. 45)

De forma sucinta, a entrada do processo é uma interface que o usuário vê, e os artefatos gerados são os casos de uso, o diagrama de robustez, o diagrama de sequência e o modelo de domínio que depois evolui para um modelo de classe, em cima disso o código é escrito e depois testado.

Independente da forma que se escolha para desenvolver um aplicativo, e sua metodologia, também deve-se pensar em como manipular e armazenar os dados que são necessários para o correto funcionamento da aplicação. Tendo já visto as diferentes formas de se desenvolver a aplicação, a seção 4.3 aborda mais detalhadamente a parte de armazenamento.

4.3 ARMAZENAMENTO E DISPONIBILIZAÇÃO DE INFORMAÇÕES

Um aplicativo geralmente é uma interface feita para se manipular dados. Para que seu funcionamento seja adequado e agradável ao usuário, é necessário que o aplicativo possua acesso a todos os dados que precisar, na hora em que precisar.

Há algumas possibilidades no quesito armazenamento de dados, e nenhuma delas é excludente, podendo-se até usar todas em conjunto. Algumas opções de armazenamento são:

- a) Armazenamento Local
- b) Armazenamento em Servidor Próprio
- c) Armazenamento em Nuvem

Essas opções serão abordadas respectivamente nas seções 4.3.1, 4.3.2 e 4.3.3.

4.3.1 Armazenamento Local

A possibilidade de se armazenar itens interna e diretamente no dispositivo já é padrão dos sistemas mais utilizados, o iOS da Apple oferece a classe `NSUserDefaults` Apple (2019) e o Android da Google também oferece suas opções, dentre as quais temos `SharedPreferences`, por exemplo Google (2019g). Esse tipo mais simples de armazenamento se resume a armazenar os dados dentro do dispositivo no qual o aplicativo está rodando.

Como os dados estão localizados no mesmo dispositivo que o aplicativo, a leitura e a escrita dos mesmos não exige nenhuma comunicação externa, facilitando assim esse processo. Porém os dados precisam chegar até o dispositivo de alguma maneira, sendo assim:

- a) Ou o aplicativo já possui todas as informações que necessita, e, ao ser instalado, salva esses dados no dispositivo;
- b) Ou o aplicativo também depende de uma outra fonte de armazenamento externa, que provê os dados para o aplicativo e o esse se encarrega de passar ao dispositivo.

Caso a situação seja a primeira, o aplicativo não possui dados dinâmicos, ou seja, a informações serão sempre as mesmas provenientes da instalação, ou variações dessas. Caso a situação seja a segunda, o aplicativo pode possuir um conjunto vivo de informações, que pode ir se alterando com o passar do tempo por influências externas.

A segunda situação é encontrada com maior facilidade, pois possibilita uma abordagem de que o aplicativo sempre tenha algum dado para exibir ao usuário. Por exemplo, no primeiro acesso do usuário o aplicativo busca os dados externamente, salva no dispositivo e exibe para o usuário, já no segundo acesso, o aplicativo exibe para o usuário os dados que já possui dentro do armazenamento interno, e só depois tenta buscar os dados externamente. Caso o aplicativo não consiga buscar novas informações, ao menos o usuário terá a informação mais recente possível.

4.3.2 Armazenamento em Servidor Próprio

Se for necessário ter um maior dinamismo nos dados, ou não se puder tê-los disponíveis no momento de envio do aplicativo para sua respectiva loja, será necessário armazenar os dados em algum lugar externo ao aplicativo. O armazenamento em servidor próprio é uma dessas alternativas.

As informações necessitam ser armazenadas em algum lugar físico, para que depois sejam enviadas e existam posteriormente dentro do aplicativo. Esse lugar físico, que são dispositivos de armazenamento, podem estar dentro de uma empresa terceirizada ou dentro da própria empresa do aplicativo, esse segundo caracterizando armazenamento em servidor próprio.

De acordo com SysGen (2018) pode-se observar as seguintes vantagens nessa abordagem:

- a) A empresa possui controle total do backup, tanto no que se refere ao momento quanto a periodicidade;
- b) Dados sensíveis não saem da empresa. Como o equipamento fica dentro das dependências da empresa, não há o risco das informações sensíveis caírem nas mãos de terceiros;
- c) Ausência de necessidade de conexão com a internet. Não necessita de tráfego externo, colaborando com o ponto acima.

Mas SysGen (2018) também traz desvantagens dessa mesma abordagem:

- a) Necessita de um investimento inicial em equipamento. Para se ter toda a estrutura de armazenamento internamente na empresa, primeiro é necessário adquirir os equipamentos. Dessa forma seu custo inicial é maior que o de outras alternativas;
- b) Ocupa espaço nas instalações da empresa. O equipamento necessário vai precisar de um local adequado para ser colocado dentro das dependências da empresa.

- c) Suscetibilidade à perda. A empresa também necessita preocupação com qualquer desastre natural ou acidente que possa ocorrer em suas dependências, pois qualquer evento pode ocasionar a perda dos dados ali armazenados.

Em resumo o armazenamento em servidor próprio é uma alternativa que pode trazer segurança aos dados armazenados, porém também cobra um preço equivalente a isso.

4.3.3 Armazenamento em Nuvem

Em oposição ao armazenamento em servidor próprio, caso o armazenamento dos arquivos seja feito por um empresa terceira, e não se possuir os equipamentos dentro da própria empresa, temos caracterizado um armazenamento em nuvem.

Trikha (2010) mostra de maneira simplificada como essa modalidade de armazenamento funciona:

- a) Os arquivos que necessitam de armazenamento são enviados da sua origem até os servidores do provedor de armazenamento, via internet;
- b) Ao receber os arquivos, os servidores se encarregam de salvá-los em diferentes lugares de seu sistema;
- c) Com os arquivos salvos, assim que a origem (ou qualquer outro dispositivo autorizado) solicitar novamente os arquivos, os servidores procurarão uma cópia dos mesmos e enviarão ao solicitante através da internet.

Do ponto de vista da origem/solicitante não há diferença visível em relação ao armazenamento em servidor próprio. A decisão sobre qual estratégia de armazenamento utilizar influencia muito mais a quem está tomando essa decisão, juntamente com o responsável pelo tipo de informação que ficará armazenada.

Pensando em um negócio, ou no responsável por essa decisão, pode-se observar alguns benefícios dessa estratégia de armazenamento, como se vê em Pritchett (2018):

- a) **Acesso aos arquivos:** De qualquer lugar com internet é possível ter acesso ao que se armazenou previamente, não se limitando assim ao alcance interno de um armazenamento em servidor próprio;
- b) **Baixo investimento inicial:** Por não se ter os gastos de aquisição de *hardware* de armazenamento e de se montar toda a estrutura necessária, o armazenamento em nuvem acaba sendo uma solução muito mais barata para se alcançar o objetivo;

- c) **Escalabilidade:** O futuro de um negócio é incerto, pois o negócio pode tanto crescer como ser reduzido, e assim por consequência sua necessidade de armazenamento. O armazenamento em nuvem torna isso mais fácil de ser contornado, possibilitando rápidos ajustes tanto na questão do espaço necessário quanto nos custos envolvidos;
- d) **Ausência de manutenção:** Assim como todo componente de um computador, um dispositivo de armazenamento pode vir a estragar, mas no armazenamento em nuvem isso se torna responsabilidade do fornecedor, tanto a reposição do dispositivo quanto a manutenção em todo o sistema de armazenamento.

Em contrapartida, com Pritchett (2018) pode-se observar também as desvantagens dessa solução de armazenamento:

- a) **Privacidade dos dados:** Como os dados são enviados para um terceiro, é muito importante que seja levado em conta o quanto essas informações são sensíveis e secretas, exigindo assim a escolha de um fornecedor seguro e confiável;
- b) **Vulnerabilidade:** Há uma abertura para ataques externos visando acesso os dados, visto que esses dados são recebidos e enviados através da internet, que pode ser um meio não muito seguro;
- c) **Custos a longo prazo:** Por se tratar de um serviço há custos envolvidos com os terceiros, custos esses que podem durar por todo o tempo de fornecimento do serviço. E como também se faz necessário o uso da internet para utilização, pode ser que haja um aumento dos custos relacionados a internet também.

Tendo sido estas características abordadas, as seções seguintes detalham duas soluções de armazenamento que fazem uso do armazenamento em nuvem, Firebase na seção 4.3.3.1 e Back4App na seção 4.3.3.2.

4.3.3.1 Firebase

Firestore é uma plataforma de serviços para desenvolvimento *web* e desenvolvimento móvel que foi adquirida pela Google em 2014, como visto em Tamplin (2014). Esplin (2016) descreve o Firestore como algo que permite ao desenvolvedor poder focar apenas no seu objetivo principal, pois com o Firestore não é necessário gerenciar servidores e nem escrever *APIs*, já que o Firestore se responsabiliza por essa parte, sendo também um sistema de armazenamento de dados, tudo feito de maneira genérica para que seja facilmente adaptável a maioria das necessidades.

Em 1 de maio de 2019 o Firestore oferecia 18 produtos, como visto em Google (2019e), dentre esses há exemplos como:

- a) **Cloud Firestore**, que armazena dados e sincroniza os mesmos entre diferentes dispositivos, usando um banco de dados não relacional e oferecendo suporte *offline*;
- b) **Hospedagem**, fornecida para aplicações *web*, já incluindo certificado SSL, segurança e velocidade;
- c) **Autenticação**, dispo de diversos métodos para se gerenciar os usuários, incluindo integração com provedores de identidade terceiros como Facebook e Google;
- d) **Cloud Storage**, que é um serviço de armazenamento de arquivos na nuvem, oferecendo segurança no *download* e no envio dos arquivos, independente da qualidade da conexão.

Ainda em Google (2019e) pode-se ver outros produtos relacionados mais diretamente com aplicativos, como o **Crashlytics** que reporta erros em tempo real e os organiza para melhor gerenciamento, o **Monitoramento de desempenho** que verifica o quanto o aplicativo está usando de cada recurso enquanto está sendo usado, e o **Test Lab** para a realização de testes automatizados que descobrem *bugs* e inconsistências.

Como se pode ver em Google (2019d), o Firestore libera seus produtos para uso de forma gratuita até o atingimento de certos limites, o que pode acabar suprimindo a necessidade de um pequeno negócio, ou de um negócio em expansão, após isso cobra-se baseado no uso do produto.

Seus produtos mais utilizados são os que atuam como bancos de dados, o **Cloud Firestore** e o **Realtime Database**, e é com base nesses que Jamin (2016) relata algumas desvantagens:

- a) **Possibilidade de inconsistências:** Como o Firebase dá suporte ao uso *offline*, pode ser que mais de um usuário esteja operando *offline* a mesma informação, o que pode causar um conflito quando os dois conseguirem conexão com a internet e enviarem as suas operações;
- b) **Banco de dados não relacional:** Por causa da escolha do Firebase de guardar os dados de maneira não relacional, migrações de dados se tornam mais difíceis, e também o relacionamento entre os dados é mais complexo do que em um banco de dados relacional.

4.3.3.2 Back4App

O Back4App é uma plataforma de *backend* baseada no *framework* Parse. Como descrito em Back4App (2019b) o Parse é um *framework* de *backend* desenvolvido pelo Facebook, e o Back4App adiciona melhorias sobre o Parse Server, que é a versão de código aberto do Parse, oferecendo um serviço de hospedagem de primeira classe.

“O Parse Server trabalhou nas deficiências que estavam presentes no Parse e reteve as coisas boas que estavam presentes no Parse” (MELO, 2016, tradução nossa), possuindo características como:

- a) Facilidade de implantação;
O Parse precisa apenas de uma infraestrutura que suporte Node.js, e necessita de poucas alterações no código assim que os dados já estejam no banco de dados.
- b) Armazenamento de dados;
O Parse se utilizava de MongoDB e da Amazon S3 para salvar arquivos, já o Parse Server permite a livre escolha de qual sistema de arquivos usar, podendo também salvar os arquivos em JSON como forma de *backup*.
- c) Painel de controle;
O painel de controle do Parse continua presente, onde os usuários podem configurar seus aplicativos e enviar notificações *push*.
- d) Consultas rápidas;
O Parse Server busca os dados em tempo real e de acordo com o que o usuário solicitar.

Melo (2016) cita um grande diferencial do Parse Server, que é o fato de possuir código aberto, ou seja, sua utilização não cria vínculo com nenhum fornecedor, como o Firebase cria com a Google, por exemplo. Melo (2016) também aposta na longevidade dos *frameworks* baseados no Parse Server, dizendo que esses devem se manter estáveis no longo prazo muito mais que soluções vinculadas a algum fornecedor.

O Parse Server precisa ser hospedado em algum lugar, e além das alterações sobre o *framework* o Back4App também oferece essa hospedagem. Como se vê em Back4App (2019a), o Back4App possui disponibilidade gratuita de seus serviços até um limite determinado, o que colabora para que novos desenvolvedores e interessados possam conhecer a plataforma antes de ter que pagar.

Ao conceder a liberdade para um usuário inserir novas informações em um sistema, armazená-las, e passar a se utilizar dessas informações dentro do sistema, exibindo-as por exemplo, cria-se a necessidade de tornar isso fácil e gerenciável. A seção 4.4 a seguir detalha mais o assunto.

4.4 SISTEMA DE GERENCIAMENTO DE CONTEÚDO

As seções a seguir discorrem em mais detalhes sobre um sistema de gerenciamento de conteúdo.

4.4.1 Definição de Sistema de Gerenciamento de Conteúdo

Um Sistema de Gerenciamento de Conteúdo (SGC) pode ser estudado em 3 níveis para se facilitar o entendimento:

- a) Conteúdo;
- b) Gerenciamento de Conteúdo;
- c) Sistema de Gerenciamento de Conteúdo.

Barker (2016, p. 31) levanta a questão sobre o que de fato é conteúdo, e o que deve ser classificado como dado ou informação por exemplo, mas também responde esta dúvida com um conceito muito claro, de que “Conteúdo é informação produzida através de processo editorial e finalmente destinada a consumo humano via publicação” (BARKER, 2016, p. 35, tradução nossa), ou seja, o conteúdo tem o fator humano envolvido, tem influência do sentimento do editor do conteúdo no momento de sua criação.

A respeito de gerenciamento de conteúdo, Barker (2016, p. 29) menciona o erro que geralmente se comete ao se pensar que gerenciamento de conteúdo é algo digital, pois desde que os seres humanos começaram a criar conteúdo já começou a busca por formas de organizá-los. O que há de maneira digital hoje para se gerenciar conteúdo é apenas uma transposição do que já se fazia manualmente tempos atrás.

E por fim, tratando de um SGC, Barker (2016, p. 36) aborda isso como um programa que automatiza de alguma maneira as tarefas de gerenciamento de conteúdo. Como características de um SGC observa-se que “[...] permite aos editores criar novo conteúdo, editar conteúdo existente, realizar processos editoriais sobre o conteúdo e, finalmente, disponibilizar esse conteúdo [...]” (BARKER, 2016, p. 36, tradução nossa)

Posteriormente Barker (2016, p. 36) também comenta que um SGC é composto por várias partes, como a interface de edição, o armazenamento do conteúdo e o sistema de publicação por exemplo, mas para quem se utiliza do SGC e não possui uma visão técnica, tudo se torna um coisa única chamada SGC.

4.4.2 Características de Softwares Gerenciadores de Conteúdo

Encontra-se no mercado os mais diversos sistemas para gerenciamento de conteúdo, cada um com suas particularidades e propostas, porém pode-se estipular o que deve ser o mínimo para um sistema ser considerado como gerenciador de conteúdo.

Barker (2016, p. 42-45) apresenta as características que um sistema gerenciador de conteúdo deve apresentar, que são:

a) Controlar conteúdo;

O SGC é o responsável por saber das informações referentes ao conteúdo, por exemplo: Quem pode ver ou alterar o conteúdo; se o conteúdo já foi finalizado e publicado; quantas vezes se alterou o conteúdo e o que foi alterado; quais influências o mesmo sofre e exerce sobre outros conteúdos.

b) Permitir reutilização do conteúdo;

A possibilidade de se utilizar um conteúdo criado para um lugar em outro local diferente depende das capacidades do SGC. O sistema deve ser capaz de apenas referenciar o conteúdo nos lugares que ele for utilizado, e não recriar todo o conteúdo sempre que necessário.

c) Permitir automatizar e agrupar conteúdo;

Poder visualizar um conjunto de conteúdos em um só lugar facilita a busca e a manipulação dos mesmos, Isso se torna possível pelo fato do SGC ser a fonte única das informações, podendo assim operar mais facilmente sobre o que está sob seu comando.

- d) Aumentar a eficiência de seus usuários;

Um SGC deve ser capaz de facilitar o processo de geração de conteúdo para quem o utiliza, possibilitando uma geração maior de conteúdo em menos tempo, não sendo nem criando barreiras no processo.

Todo SGC visa implementar, a sua maneira, estas características. É dessa maneira que surgem os diferentes sistemas, cada um tentando trazer seu diferencial em algum ponto.

4.4.3 Exemplos de Softwares Gerenciadores de Conteúdo

Esta seção apresenta alguns sistemas gerenciadores de conteúdo já existentes e utilizados no mercado, que segundo Ciriaco (2019) são os melhores:

- a) Wordpress

Provavelmente a solução mais conhecida pelo público em geral, o autor menciona que “Ele é gratuito, está repleto de recursos extras (como plugins e templates prontos e também gratuitos disponíveis na web) e é utilizado por inúmeros sites grandes” (CIRIACO, 2019), mas também comenta que este SGC necessita de incrementos no quesito segurança.

- b) Joomla

Sendo uma alternativa a altura para o Wordpress “também se destaca pelas inúmeras aplicações que podem incrementar o funcionamento e os recursos de uma página, além de armazenar conteúdos em diversas linguagens distintas” (CIRIACO, 2019). O autor também comenta o fato deste SGC ser baseado na mesma linguagem do Wordpress, que é PHP.

- c) Drupal

O autor menciona que é a escolha da seção do Twitter para desenvolvedores, do site do Instituto de Tecnologia de Massachusetts e também da página do governo dos Estados Unidos. Também “funciona com base em vários módulos que podem ser plugados entre si para colocar uma página e seu conteúdo no ar. Além disso [...] gerencia múltiplos níveis de conteúdo, permitindo um alto nível de personalização” (CIRIACO, 2019)

É possível notar que todos estes possuem destacado em comum a possibilidade de se utilizar de módulos prontos que outros usuários já desenvolveram, agilizando a implementação de alguns recursos.

Os itens vistos até o momento nos dão a base para efetuar a criação de uma solução tecnológica, porém tal criação só tem valor se aplicada a algum objetivo. O desenvolvimento de software é responsável por atuar nas necessidades de inúmeras áreas, dentre essas está a área da saúde e também o seu ramo da fisioterapia. O capítulo a seguir mostra uma visão a respeito de cuidados fisioterápicos e de como a tecnologia atua nessa área.

4.5 FISIOTERAPIA

Ao se efetuar uma pesquisa sobre o significado do termo pode-se encontrar a seguinte definição para fisioterapia: “Tratamento médico por meio de agentes físicos: luz, calor, frio, eletricidade, exercício etc.” (FISIOTERAPIA, 2019).

Esse tratamento tem por objetivo ajudar as “[...] pessoas afetadas por lesões, doenças ou deficiências através de movimentos e exercícios, terapia manual, educação e aconselhamento” (CHARTERED SOCIETY OF PHYSIOTHERAPY, 2018, tradução nossa), e observa-se que esse objetivo já existe de longa data pois “Desde a sua origem, a fisioterapia tem um caráter essencialmente curativo e reabilitador” (JÚNIOR, 2007).

Pode-se observar em Shaik e Shemjaz (2014) que o surgimento da necessidade por um ramo como esse na medicina surgiu nos anos de guerra, pois se fez necessário recuperar soldados que voltavam para casa feridos, amputados, com ossos quebrados, queimaduras e também outras sequelas. Crefito-3 (2018) também mostra que o objetivo dos fisioterapeutas era efetuar uma recuperação para que os soldados pudessem voltar ao mercado de trabalho ou se readaptassem para alguma outra função.

No Brasil, como se vê em Crefito-3 (2018), a fisioterapia iniciou em 1929 na Santa Casa de Misericórdia de São Paulo. Os atendimentos pelos fisioterapeutas não são mais realizados em campos de batalha, agora são efetuados em consultórios mais preparados para o exercer da função, e nesses consultórios é que se realizam exercícios de fisioterapia que são detalhados na próxima sessão.

4.5.1 Exercícios fisioterápicos

Para tentar alcançar seus objetivos a fisioterapia usa de exercícios fisioterápicos, que visam “oferecer qualidade de vida às pessoas, por meio da prevenção e reabilitação física” (ROCKCONTENT, 2019), mas também “faz parte da valorização da prevenção e promoção da saúde como um todo, servindo como veículo para uma série de cuidados e programas” (ROCKCONTENT, 2019).

Os exercícios são indicados para quem sente dores, possui problemas de postura ou queira se prevenir de problemas osteomusculares por exemplo, como se vê em ROCKCONTENT (2019).

Tendo como foco a reabilitação do paciente, foi pesquisado se a realização de exercícios após se ter uma dor lombar reduziria a chance da dor voltar, e se evidenciou que “[...] programas de exercícios pós-tratamento podem prevenir a recorrência de dores nas costas e, portanto, pode ser benéfico para os pacientes seguirem programas de exercícios adicionais após o tratamento [...] ter sido concluído” (CHOI et al., 2010, tradução nossa).

Porém os exercícios não são recomendados de maneira geral para todos os problemas. O estudo feito por Maher (2004) reforça que no caso de dor lombar os exercícios possuem alta eficiência na melhora do paciente, mas outros métodos como acupuntura e ultrassom já não são eficientes por exemplo. Com os diferentes tipos de exercícios fisioterapêuticos que podem ser realizados ocorre a mesma situação, por isso é importante a instrução de um profissional, como detalhado na próxima sessão.

4.5.2 Acompanhamento do Paciente

Uma sessão convencional em uma consulta de fisioterapia consiste no paciente estar efetuando os exercícios e o fisioterapeuta estar ao seu lado lhe auxiliando. Um estudo feito sobre a interação entre o paciente e o fisioterapeuta concluiu, na visão dos fisioterapeutas que “A qualidade da interação teve um grande efeito no desfecho do paciente. Reflexões sobre as experiências e os resultados dos pacientes levaram a um maior entendimento da importância da interação no tratamento do paciente [...]” (GYLLENSTEN, 1999, tradução nossa).

Nesse estudo citado os fisioterapeutas “concentraram-se em um diálogo construtivo com os pacientes, com empatia, respeito, engajamento, sensibilidade e capacidade de serem vistos como ferramentas importantes para ensinar aos pacientes [...] assumir a responsabilidade por sua [...] saúde.” (GYLLENSTEN, 1999, tradução nossa). Nessa visão o fisioterapeuta parece demonstrar ser essa figura humana que se interessa e trabalha pelo paciente e por sua evolução.

Há outro estudo, um pouco mais recente, sobre um programa de fisioterapia feito em casa para pessoas com a doença de Parkinson, esse estudo constata que “os pacientes tiveram escores significativamente mais altos em uma escala de atividade funcional após o tratamento no ambiente doméstico e em menor grau no hospital, um resultado que foi parcialmente sustentado no acompanhamento” (NIEUWBOER et al., 2001, tradução nossa)

Inclusive menciona que “O presente estudo apoia o uso desse conceito de reabilitação, já que implementar esse tratamento em casa levou a um desempenho mais efetivo das atividades funcionais em pacientes com uma duração média da doença de 12 anos” (NIEUWBOER et al., 2001, tradução nossa), expondo assim um resultado promissor para a realização da fisioterapia em casa, mesmo que neste caso apenas no tratamento do Parkinson.

Com essas visões pode-se dizer que ambas as abordagens conseguem obter a evolução do paciente no tratamento, cada uma com seu enfoque específico. Visto isso, a próxima sessão detalha sobre o que pode ser automatizado ou simplificado no processo da fisioterapia, sem que se prejudique a evolução que se alcança hoje.

4.5.3 Tecnologização na Fisioterapia

Ao se pensar em tecnologia aplicada a fisioterapia pode-se pensar tanto em algum sistema mais analítico, visando apenas entregar melhores informações aos fisioterapeutas, quanto em dispositivos e aparelhos mais tangíveis que facilitem o trabalho diário do profissional.

Um caso brasileiro de dispositivo tangível é o sensor híbrido para robótica e fisioterapia, que pode ser usado para o controle de uma cadeira de rodas robótica, por exemplo, ou também para avaliar o senso posicional do joelho, como se vê em Bôrtol e Filho (2011). Nesse caso da avaliação do joelho “Os equipamentos utilizados recentemente para esta tarefa são usualmente sistemas optoeletrônicos com câmeras de alta velocidade (fotogrametria), que tem custo muito elevado” (BÔRTOLE; FILHO, 2011), ou seja, o dispositivo traz benefícios inclusive no custo do tratamento.

Se for pensado em algo mais virtual, temos a plataforma *PhysioTherapy eXercises*, que consiste em site e um aplicativo. O site possui uma lista com mais de 1000 exercícios focados em deficiências e atividades, está disponível em 13 idiomas, e pode ser usado pelos fisioterapeutas para criar programas de exercícios para seus pacientes, como visto em Flynn (2018).

O aplicativo da plataforma *PhysioTherapy eXercises* é feito com o foco no paciente, pois é no aplicativo que ele consegue ver os exercícios que o fisioterapeuta recomendou, marcar cada um como concluído e avaliar o programa para o fisioterapeuta, como em Flynn (2018). No site o fisioterapeuta pode monitorar o progresso do paciente mesmo a distância e ajustar o programa de exercícios conforme necessário, também considerando as avaliações feitas pelos pacientes, também exposto em Flynn (2018).

A ideia e execução mostram-se promissoras, em geral “o *Physiotherapy Exercises App* é um recurso clínico excelente e fácil de usar” (FLYNN, 2018, tradução nossa). Suas vantagens são que “Fornece aos pacientes acesso ao seu programa de exercícios em casa em todos os momentos, facilita o autogerenciamento e, mais importante, aumenta a comunicação entre o paciente e o terapeuta.” (FLYNN, 2018, tradução nossa).

Após o surgimento e implantação das automatizações e dos sistemas, é necessário aferir se essas ferramentas colaboram de fato ou não com a evolução do paciente no tratamento. A sessão seguinte aborda mais essa questão sobre as métricas.

4.5.4 Avaliação de Resultados de Tratamentos

A aferição da evolução de um paciente pode ser um pouco complicada para o fisioterapeuta, principalmente se o item em questão a ser observado é algo que está no paciente e não é visível ao fisioterapeuta, como a dor por exemplo. Jensen, Karoly e Braver (1986) realizaram um estudo com 6 diferentes formas para se medir o nível de dor de um paciente:

- a) Escala Visual Analógica (VAS);
- b) Escala de Classificação Numérica de 101 pontos (NRS-101);
- c) Escala de Caixa de 11 pontos (BS-11);
- d) Escala de Classificação Comportamental de 6 pontos (BRS-6);
- e) Escala de Classificação Verbal de 4 pontos (VRS-4);
- f) Escala de Classificação Verbal de 5 pontos (VRS-5).

Essas escalas são simplesmente formas diferentes de questionar ao paciente sobre a dor que ele sentia. Os pacientes, dadas algumas condições, respondiam em todas as escalas, e o que se concluiu é que “[...] as diferenças obtidas não foram significativas, indicando que os indivíduos nesta amostra de pacientes com dor crônica geralmente não apresentaram respostas mais incorretas para uma escala do que para qualquer outra” (JENSEN; KAROLY; BRAVER, 1986).

Mas esse desafio não é visto apenas tratando-se de dor, ao se tratar de maneira geral sobre como avaliar a fisioterapia através da medição dos resultados se vê que “O tipo de informação que os fisioterapeutas precisam registrar é extremamente difícil de capturar.” (HAMMOND, 2000, tradução nossa) pois como o autor explica “Não podemos confiar apenas nos raios X, nos exames de sangue e nas varreduras. Registramos informações principalmente do que os pacientes nos dizem (“minha dor já passou”) ou do que observamos” (HAMMOND, 2000, tradução nossa).

A tentativa de se obter o melhor no quesito avaliação do paciente está sempre em execução, tanto que Hammond (2000) comenta que há várias mensurações padronizadas já publicadas. Porém o desafio parece que sempre estará presente, pois não há como fugir da subjetividade nesse momento, o que demanda também uma colaboração do paciente, que necessita sempre ser sincero quanto a suas sensações e experiências em relação ao tratamento.

5 PROPOSTA DE SOLUÇÃO

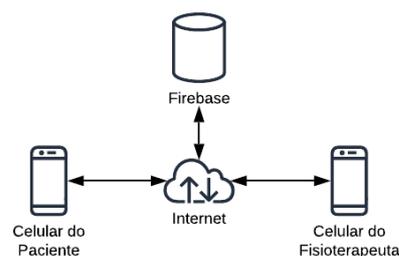
O projeto consiste no desenvolvimento de um aplicativo para dispositivos móveis, que permitirá aos fisioterapeutas realizar a inserção de exercícios no sistema, detalhando como executar cada um através de imagens, textos e também vídeos. Essas imagens e vídeos podem ser inseridos diretamente do celular, possibilitando também sua captura mesmo no momento do cadastro. Dependendo do nível de acesso do fisioterapeuta, ele será capaz de cadastrar novos fisioterapeutas no aplicativo, que podem ser seus colegas ou seus funcionários, dependendo do cargo que o fisioterapeuta com acesso privilegiado exercer na sua organização.

O fisioterapeuta possui acesso a lista de pacientes, podendo assim encontrá-los através de seu nome e ver detalhes de cada um. O profissional também poderá agrupar os exercícios em listas, chamadas protocolos, e então recomendá-las ao seus pacientes, para que esses recebam os protocolos no seu celular e consigam visualizá-los. Os detalhes do paciente exibidos ao fisioterapeuta contemplam também os protocolos que já lhe foram recomendados, para o profissional também consiga ver um histórico de quem ele está tratando.

Para o paciente o aplicativo exibirá apenas os protocolos que lhe foram recomendados pelos fisioterapeutas, assim ele poderá visualizar no detalhe cada exercício da lista para poder entendê-lo e executá-lo da forma correta, também não se distraindo com exercícios não adequados, visto que o aplicativo já fará esse filtro para ele. O paciente também receberá periodicamente um questionário através do aplicativo, para que se possa entender se o aplicativo está lhe beneficiando de alguma forma.

O aplicativo só necessitará de conexão com a internet para acessar os protocolos na primeira vez que o paciente visualizar cada protocolo, os próximos acessos aos protocolos poderão ser realizados *offline*, possibilitando ao paciente realizar os exercícios onde e quando for mais adequado para si, sem a necessidade de que o mesmo se desloque até a clínica. A Figura 3 ilustra o ecossistema da solução.

Figura 3 – Arquitetura da Solução



6 PROJETO

Pelas suas lideranças no ranking de participação no mercado, conforme citado na seção 4.1, optou-se por desenvolver o aplicativo para os 2 sistemas operacionais mais usados: Android e iOS. Para a implementação da solução proposta optou-se por desenvolver um aplicativo híbrido, possibilitando assim a criação mais rápida de um aplicativo para os 2 sistemas operacionais.

O *framework* base escolhido para se usar no projeto será o Google Flutter, visto que o mesmo é capaz de gerar aplicativos tanto para Android quanto para iOS com o mesmo código fonte. Além disso confia-se na empresa Google que é a empresa mantenedora do *framework*, visto que é uma solução nova e são previstos desafios ao longo do trabalho de desenvolvimento.

Como solução de armazenamento será utilizado um serviço do Firebase, o Cloud Firestore, por ser um banco de dados em nuvem flexível e escalonável, gerenciar a alteração dos dados em tempo real Google (2019a), e também ser capaz de controlar a inserção e recuperação de dados mesmo se estando *offline* Google (2019b). O Firebase foi escolhido também pelo fato de não possuir custos para projetos com um uso baixo de seus serviços.

Este aplicativo diferencia-se do *PhysioTherapy eXercises* principalmente por já possuir dentro da própria solução um módulo para inserção de novos exercícios, que é sua parte de gerenciamento de conteúdo, possibilitando realizar todas as ações necessárias através do aplicativo, sem precisar acessar nenhum outro sistema externo.

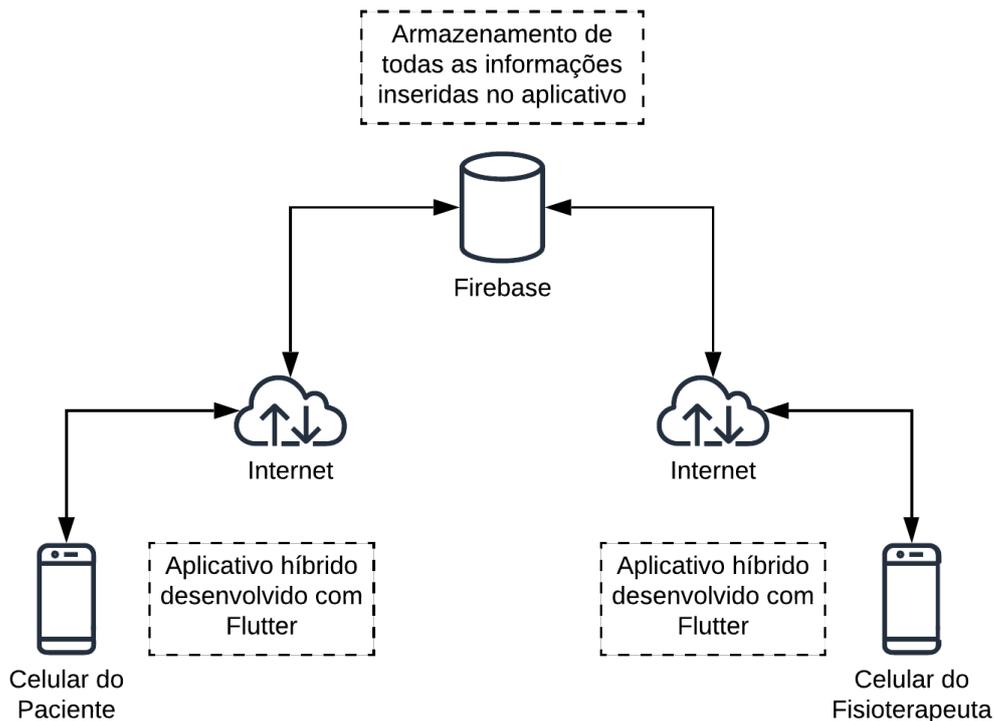
A metodologia de desenvolvimento escolhida para nortear o projeto é a metodologia Iconix, por não gerar muitos artefatos e documentação que poderiam acabar sendo desnecessários, mas ao mesmo tempo exigindo artefatos relevantes para o processo de desenvolvimento, que conseguem de antemão encontrar alguma possível falha na parte conceitual do projeto. Utiliza-se a Linguagem de Modelagem Unificada (UML) para a elaboração da estrutura do projeto.

Nas seguintes seções é possível observar os artefatos escolhidos para utilização, que foram gerados através da metodologia Iconix.

6.1 TOPOLOGIA DO SOFTWARE PROPOSTO

A partir do estudo dos requisitos, propõe-se a seguinte topologia para o *software*:

Figura 4 – Topologia



Fonte: Autor.

O aplicativo, que será desenvolvido com o *framework* Flutter, será gerado tanto para Android quanto para iOS, para possibilitar uma alta compatibilidade com os dispositivos que as pessoas envolvidas possuem. O aplicativo será o mesmo para pacientes e fisioterapeutas, os controles, acessos e permissões serão realizados internamente de maneira lógica, com base no sistema de usuários que o Firebase proporciona.

Quando um fisioterapeuta inserir uma informação nova no seu aplicativo ou editar uma já existente, será gerada uma alteração, essa alteração que ele gerou será enviada via internet para o Firebase, para que os dados lá armazenados sejam atualizados de acordo com seu mais novo estado. E de forma reversa, quando o fisioterapeuta quiser buscar as informações existentes, o aplicativo irá se comunicar com o Firebase através da internet para resgatar a informação mais atualizada e exibi-la na tela.

Da mesma forma isso ocorrerá para o paciente, quando ele responder o questionário, por exemplo, suas respostas serão enviadas ao Firebase, através da internet, e lá ficarão armazenadas. Quando o paciente visualizar pela primeira vez um protocolo, os dados do protocolo virão do Firebase através da internet. Porém, caso ele não esteja mais com acesso a internet e acessar novamente o protocolo, os dados do protocolo virão do armazenamento local do dispositivo, eliminando assim a necessidade de se estar sempre conectado a internet.

6.2 REQUISITOS

As seções 6.2.1 e 6.2.2 a seguir apresentam os requisitos levantados para o desenvolvimento da solução do projeto.

6.2.1 Requisitos Funcionais

Na Tabela 1 a seguir estão descritos os requisitos funcionais que “São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações” (SOMMERVILLE, 2011).

Tabela 1 – Requisitos Funcionais

Id	Requisito
RF-01	Cadastro e edição de fisioterapeutas, conforme nível de usuário
RF-02	Cadastro e edição de pacientes
RF-03	Cadastro e edição de exercícios, conforme nível de usuário
RF-04	Cadastro e edição dos protocolos de exercícios, conforme nível de usuário
RF-05	Busca de pacientes por nome
RF-06	Exibição dos protocolos, para pacientes e fisioterapeutas
RF-07	Exibição detalhada de exercício, através de imagens, vídeos e texto
RF-08	Autenticação dos usuários
RF-09	Exibição dos exercícios, para pacientes e fisioterapeutas
RF-10	Questionário sobre desempenho do aplicativo
RF-11	Mecanismo de recuperação de senha
RF-12	Recomendação de protocolos aos pacientes pelos fisioterapeutas

Fonte: Autor.

6.2.2 Requisitos Não Funcionais

A Tabela 2 abaixo descreve os requisitos não funcionais que “São restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de *timing*, restrições no processo de desenvolvimento e restrições impostas pelas normas.” (SOMMERVILLE, 2011).

Tabela 2 – Requisitos Não Funcionais

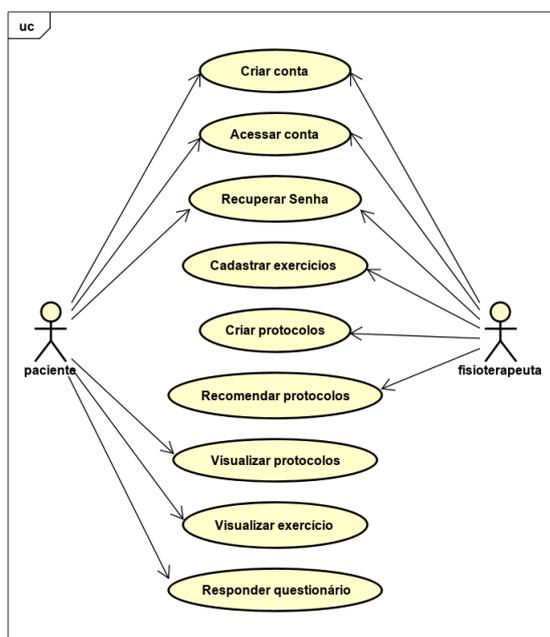
Id	Requisito
RNF-01	Trafegar os dados de modo criptografado
RNF-02	Ser fácil de usar
RNF-03	Fornecer uma experiência fluída de uso
RNF-04	Funcionar nas plataformas Android e iOS
RNF-05	Permitir acesso <i>offline</i> para pacientes

Fonte: Autor.

6.3 CASOS DE USO

Nesta seção estão detalhados os casos de uso levantados para atender a solução proposta. A Figura 5 representa o diagrama de casos de uso, onde se pode ver de maneira resumida cada caso de uso e seus respectivos relacionamentos.

Figura 5 – Casos de Uso



Fonte: Autor.

Na sequência pode-se observar os quadros dos casos de uso de forma detalhada.

Quadro 1 – Caso de Uso 01 - Criar Conta

Caso de Uso 01 - Criar Conta	
Requisitos Atendidos	RF-01 e RF-02
Atores	Paciente ou Fisioterapeuta.
Pré-condição	Paciente com o aplicativo e sem conta criada, ou fisioterapeuta com permissão. Dispositivo com acesso a internet.
Cenário principal	<ul style="list-style-type: none"> a) Usuário acessa tela de criação de conta. b) Usuário preenche todos os campos obrigatórios: nome, sexo, data de nascimento, usuário e senha. c) Usuário confirma a criação da conta. d) Aplicativo cria a conta do usuário no sistema. e) Aplicativo exibe mensagem de sucesso.
Cenários alternativos	<ul style="list-style-type: none"> a) (c) Usuário confirma a criação mas não preencheu todos os campos obrigatórios. - Aplicativo exibe mensagem de erro, solicitando preenchimento dos campos.
Pós-condição	Usuário criado no sistema.

Fonte: Autor.

Quadro 2 – Caso de Uso 02 - Acessar Conta

Caso de Uso 02 - Acessar Conta	
Requisitos Atendidos	RF-08
Atores	Paciente ou Fisioterapeuta.
Pré-condição	Paciente ou fisioterapeuta com conta já criada. Dispositivo com acesso a internet.
Cenário principal	<ul style="list-style-type: none"> a) Usuário acessa tela de acesso a conta. b) Usuário preenche usuário e senha. c) Usuário manda entrar no sistema. d) Aplicativo direciona o usuário para sua tela principal.
Cenários alternativos	<ul style="list-style-type: none"> a) (c) Usuário manda entrar no sistema mas informou o usuário ou a senha incorretamente. - Aplicativo exibe mensagem de erro, solicitando revisão das informações inseridas. b) (b) Usuário não lembra sua senha. - Usuário inicia o processo de recuperação de senha clicando em “Esqueci minha senha”. c) (b) Usuário não possui usuário e senha. - Usuário inicia o processo de criação de conta clicando em “Cadastrar-se”.
Pós-condição	Usuário possibilitado de acessar funções específicas no aplicativo.

Fonte: Autor.

Quadro 3 – Caso de Uso 03 - Recuperar senha

Caso de Uso 03 - Recuperar senha	
Requisitos Atendidos	RF-11
Atores	Paciente ou Fisioterapeuta.
Pré-condição	Usuário já cadastrado no aplicativo. Dispositivo com acesso a internet.
Cenário principal	<ul style="list-style-type: none"> a) Usuário acessa a tela de recuperação de senha. b) Usuário informa seu usuário de acesso ao sistema. c) Usuário solicita recuperação de senha. d) Aplicativo envia e-mail com as instruções de recuperação para o usuário. e) Fisioterapeuta visualiza as respostas do questionário do protocolo.
Cenários alternativos	<ul style="list-style-type: none"> a) (c) Usuário solicita recuperação mas informou um usuário não cadastrado. - Aplicativo informa que não existe conta com aquele usuário.
Pós-condição	Usuário apto a recuperar sua senha.

Fonte: Autor.

Quadro 4 – Caso de Uso 04 - Cadastrar Exercícios

Caso de Uso 04 - Cadastrar Exercícios	
Requisitos Atendidos	RF-03
Atores	Fisioterapeuta.
Pré-condição	Fisioterapeuta já logado no aplicativo.
Cenário principal	<ul style="list-style-type: none"> a) Fisioterapeuta acessa tela de cadastro de exercícios. b) Fisioterapeuta preenche nome, descrição, tempo aproximado, imagens e vídeos do exercício. c) Fisioterapeuta manda salvar o exercício no sistema. d) Aplicativo exibe mensagem de sucesso.
Cenários alternativos	<ul style="list-style-type: none"> a) (c) Fisioterapeuta manda salvar mas não informou nome ou descrição. - Aplicativo exibe mensagem de erro, solicitando inserção das informações obrigatórias.
Pós-condição	Exercício disponível para uso e visualização dos fisioterapeutas no aplicativo.

Fonte: Autor.

Quadro 5 – Caso de Uso 05 - Criar Protocolos

Caso de Uso 05 - Criar Protocolos	
Requisitos Atendidos	RF-04
Atores	Fisioterapeuta.
Pré-condição	Fisioterapeuta já logado no aplicativo.
Cenário principal	<ul style="list-style-type: none"> a) Fisioterapeuta acessa tela de criação de protocolos. b) Fisioterapeuta seleciona os exercícios que deverão compor o protocolo. c) Fisioterapeuta manda salvar o protocolo no sistema. d) Aplicativo exibe mensagem de sucesso.
Cenários alternativos	<ul style="list-style-type: none"> a) (c) Fisioterapeuta manda salvar mas não selecionou nenhum exercício. - Aplicativo exibe mensagem de erro, solicitando inserção de um exercício.
Pós-condição	Protocolo disponível para uso e visualização dos fisioterapeutas no aplicativo.

Fonte: Autor.

Quadro 6 – Caso de Uso 06 - Recomendar Protocolos

Caso de Uso 06 - Recomendar Protocolos	
Requisitos Atendidos	RF-05 e RF-12
Atores	Fisioterapeuta.
Pré-condição	Fisioterapeuta já logado no aplicativo.
Cenário principal	a) Fisioterapeuta acessa tela de busca de pacientes e procura um paciente. b) Fisioterapeuta seleciona o paciente que terá um novo protocolo sendo recomendado. c) Fisioterapeuta seleciona o protocolo que deseja recomendar. d) Fisioterapeuta manda salvar a recomendação no sistema. e) Aplicativo exibe mensagem de sucesso.
Cenários alternativos	Não apresenta
Pós-condição	Protocolo disponível para visualização pelo paciente no aplicativo.

Fonte: Autor.

Quadro 7 – Caso de Uso 07 - Visualizar protocolos

Caso de Uso 07 - Visualizar protocolos	
Requisitos Atendidos	RF-06 e RNF-05
Atores	Paciente.
Pré-condição	Paciente já logado no aplicativo.
Cenário principal	a) Paciente entra no aplicativo. b) Paciente visualiza sua lista de protocolos recomendados.
Cenários alternativos	Não apresenta
Pós-condição	Paciente ciente dos protocolos que lhe foram recomendados.

Fonte: Autor.

Quadro 8 – Caso de Uso 08 - Visualizar exercício

Caso de Uso 08 - Visualizar exercício	
Requisitos Atendidos	RF-07, RF-09 e RNF-05
Atores	Paciente.
Pré-condição	Paciente já logado no aplicativo.
Cenário principal	a) Paciente entra no aplicativo. b) Paciente seleciona um protocolo na sua lista de protocolos recomendados. c) Paciente seleciona o exercício do qual deseja ver detalhes. d) Aplicativo exibe detalhes do exercício. e) Paciente visualiza as informações do exercício.
Cenários alternativos	Não apresenta
Pós-condição	Paciente instruído para realizar o exercício aonde estiver.

Fonte: Autor.

Quadro 9 – Caso de Uso 09 - Responder questionário

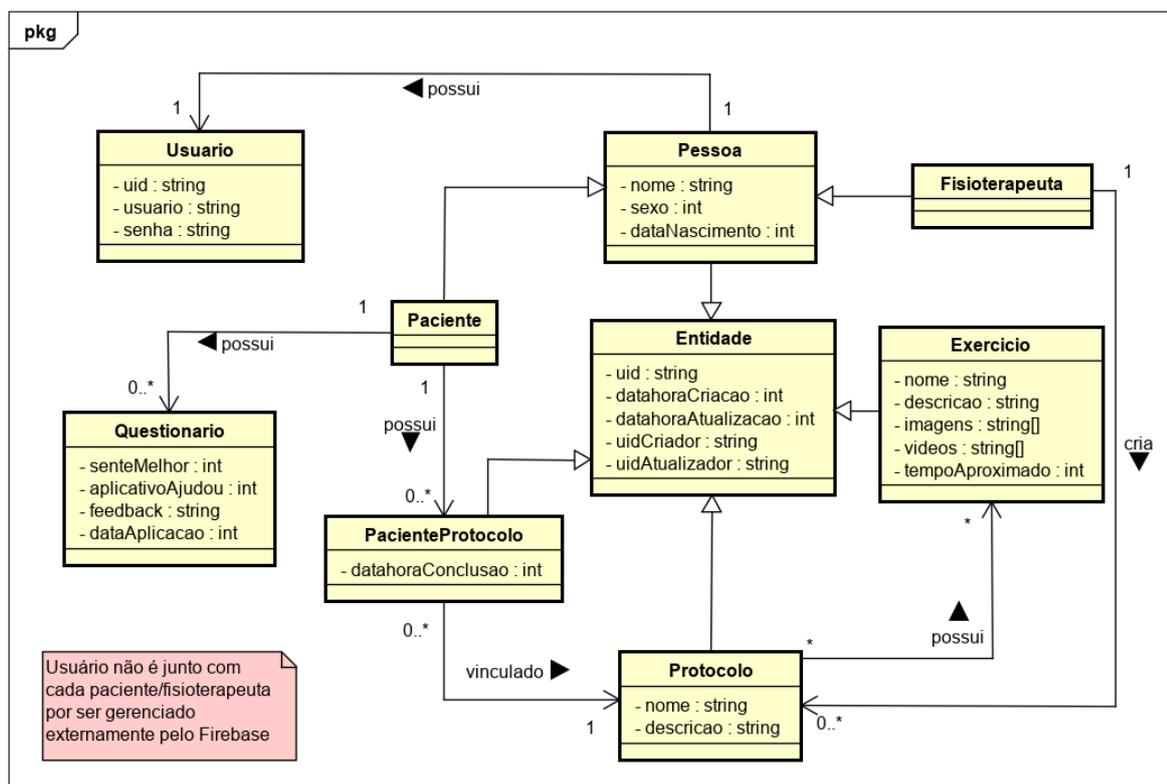
Caso de Uso 09 - Responder questionário	
Requisitos Atendidos	RF-10
Atores	Paciente.
Pré-condição	Paciente já logado no aplicativo.
Cenário principal	a) Aplicativo exibe o questionário para o paciente. b) Paciente responde as perguntas do questionário. c) Aplicativo salva as respostas do questionário.
Cenários alternativos	a) (b) Paciente opta por não responder as perguntas. - Aplicativo volta para a tela anterior.
Pós-condição	Respostas do questionário disponíveis no sistema para visualização e análise.

Fonte: Autor.

6.4 MODELO DE DOMÍNIO

A Figura 6 abaixo apresenta o modelo de domínio da solução, com suas classes e respectivas relações.

Figura 6 – Modelo de Domínio



Fonte: Autor.

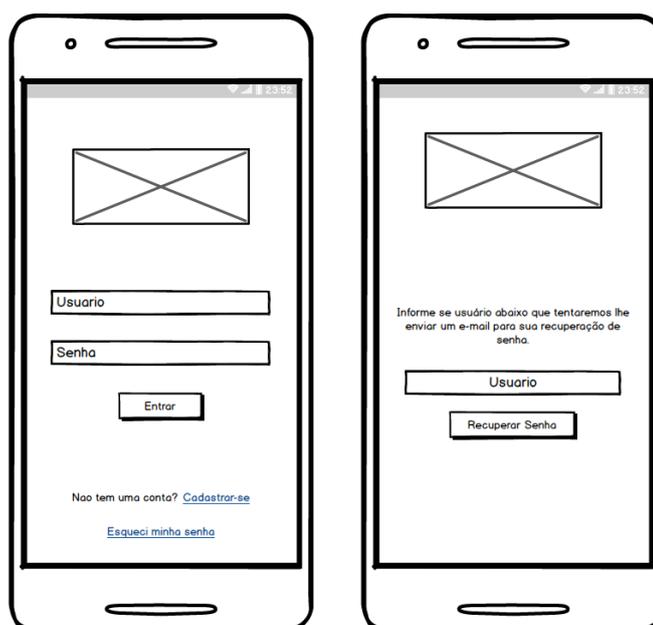
6.5 PROTÓTIPOS DE TELA

A seguir serão descritos os protótipos de tela que são esboços de como deve ser a forma final do aplicativo que será desenvolvido para atender aos requisitos anteriormente citados:

6.5.1 Acesso

Cada pessoa possuirá um usuário e senha para acessar o aplicativo. Na Figura 7 no lado esquerdo está a tela onde a pessoa inserirá seus dados de acesso para poder usar as funcionalidades do aplicativo. Caso a pessoa não lembre de sua senha ela pode clicar no link “Esqueci minha senha”, que a levará até a tela da direita da mesma Figura 7, onde ela poderá inserir seu usuário para então recuperar sua senha e poder acessar o aplicativo.

Figura 7 – Acesso ao aplicativo

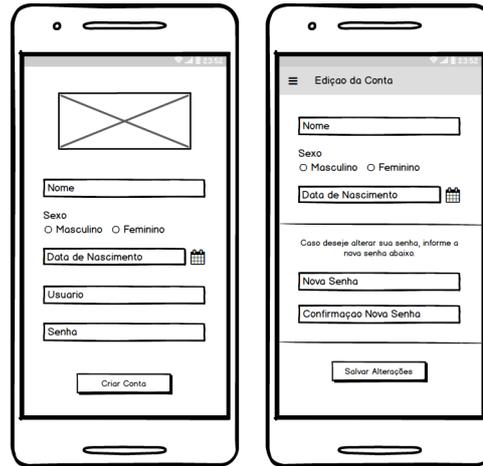


Fonte: Autor.

6.5.2 Dados Pessoais

Tanto paciente quanto fisioterapeuta possuem informações pessoais salvas no aplicativo. A tela da esquerda na Figura 8 representa onde são informados pela primeira vez, que é na criação de uma nova conta, e a tela da direita na mesma figura é acessada já com um usuário autenticado no sistema, para caso haja necessidade de se atualizar algum dado.

Figura 8 – Criação e edição da conta

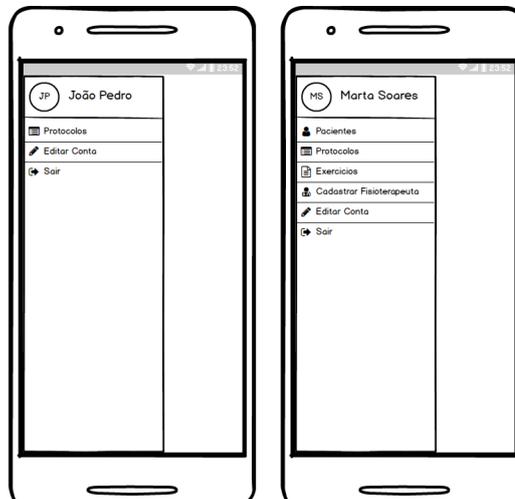


Fonte: Autor.

6.5.3 Menus

Os menus são os elementos que dão liberdade para o usuário navegar no aplicativo, e a Figura 9 apresenta os menus com as visões de paciente e de fisioterapeuta respectivamente, cada um com acesso as telas que lhe são devidas.

Figura 9 – Menus do aplicativo

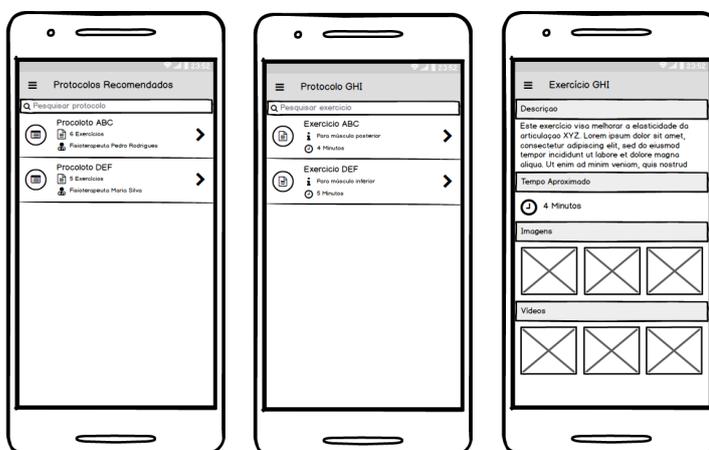


Fonte: Autor.

6.5.4 Fluxo do Paciente

Visando facilitar o uso para os pacientes, o aplicativo limita seu acesso a poucas telas. Ao entrar no aplicativo lhe é exibido a tela da esquerda da Figura 10, que é a lista de todos os protocolos que lhe foram recomendados. Ao tocar em um protocolo o paciente é redirecionado para a tela do meio presente também na Figura 10, que é a listagem de todos os exercícios que estão presentes no protocolo selecionado. Ao tocar em um exercício o aplicativo lhe leva para a tela da direita da mesma figura, que exibe todos os detalhes de como se realizar um exercício, para que o paciente possa reproduzi-lo.

Figura 10 – Fluxo para um paciente

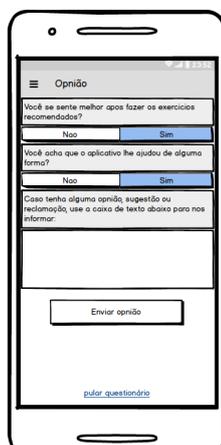


Fonte: Autor.

6.5.5 Questionário

Com o objetivo de medir como o aplicativo está desempenhando, periodicamente um questionário é liberado para o paciente responder. A Figura 11 ilustra esse questionário.

Figura 11 – Questionário

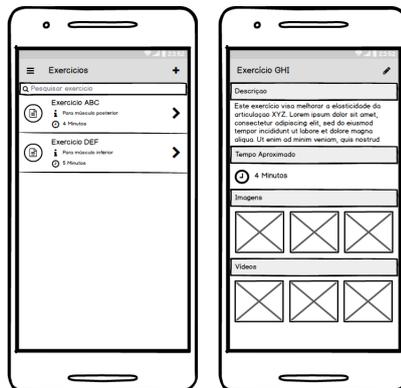


Fonte: Autor.

6.5.6 Exercícios na visão de um fisioterapeuta

O fisioterapeuta é capaz de visualizar todos os exercícios já inseridos no aplicativo, como vemos na tela da esquerda na Figura 12. Caso ele clique no exercício lhe é exibida uma tela semelhante a exibida para um paciente, apenas com a adição de um botão no canto superior direito, como se vê na tela da direita também na Figura 12. Ao se clicar no botão o aplicativo redireciona o fisioterapeuta para a tela da esquerda da Figura 13, o mesmo comportamento ocorre ao se clicar no botão de mesma posição mas na tela da esquerda.

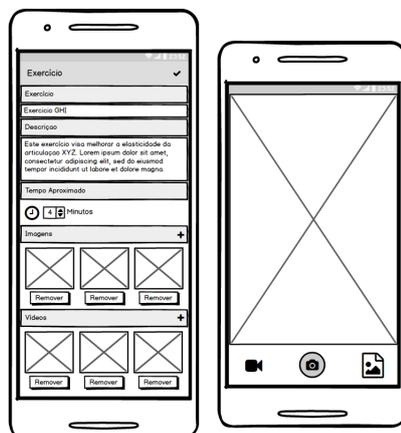
Figura 12 – Visualização de exercícios para um fisioterapeuta



Fonte: Autor.

O fisioterapeuta poderá editar as informações de exercícios já existentes, e fará isso na tela da esquerda presente na Figura 13. Essa mesma tela é utilizada para a inserção de novos exercícios no sistema. Dentro dessa tela, ao se clicar no botão com o símbolo de adição, que serve para adicionar uma nova mídia, o aplicativo possibilitará a escolha de adicionar um material já salvo na galeria do dispositivo, ou adicionar um material que será gravado ou fotografado em seguida, como vemos na tela da direita na mesma Figura 13.

Figura 13 – Edição de exercícios para um fisioterapeuta

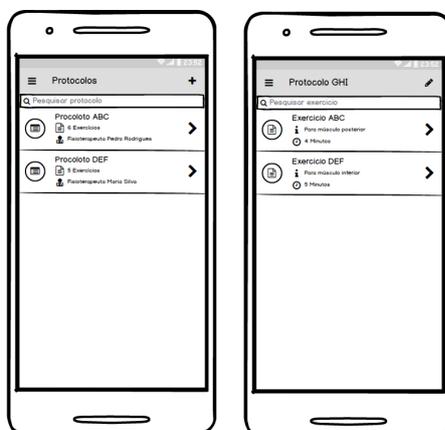


Fonte: Autor.

6.5.7 Protocolos na visão de um fisioterapeuta

O fisioterapeuta também é capaz de visualizar todos os protocolos já inseridos no aplicativo, como vemos na tela da esquerda na Figura 14. Caso ele clique no protocolo lhe é exibida uma tela semelhante a exibida para um paciente, apenas com a adição de um botão no canto superior direito, como se vê na tela da direita da mesma Figura 14. Ao se clicar no botão o aplicativo redireciona o fisioterapeuta para a tela de protocolo, exibida na esquerda da Figura 15, caso o fisioterapeuta clique no botão de mesma posição mas na tela da esquerda, ele também é redirecionado para a tela da esquerda da Figura 15, mas agora para a inserção de um novo protocolo no aplicativo.

Figura 14 – Protocolos para um fisioterapeuta



Fonte: Autor.

A tela da esquerda da Figura 15 possibilita editar um protocolo existente ou inserir um novo no aplicativo. Ao se clicar no botão ao lado direito da seção “Exercícios” o aplicativo exibe a tela da direita presente na mesma figura, que é uma busca de todos os exercícios do sistema, para possibilitar a associação deles ao protocolo que está sendo editado/criado.

Figura 15 – Edição de protocolos para um fisioterapeuta

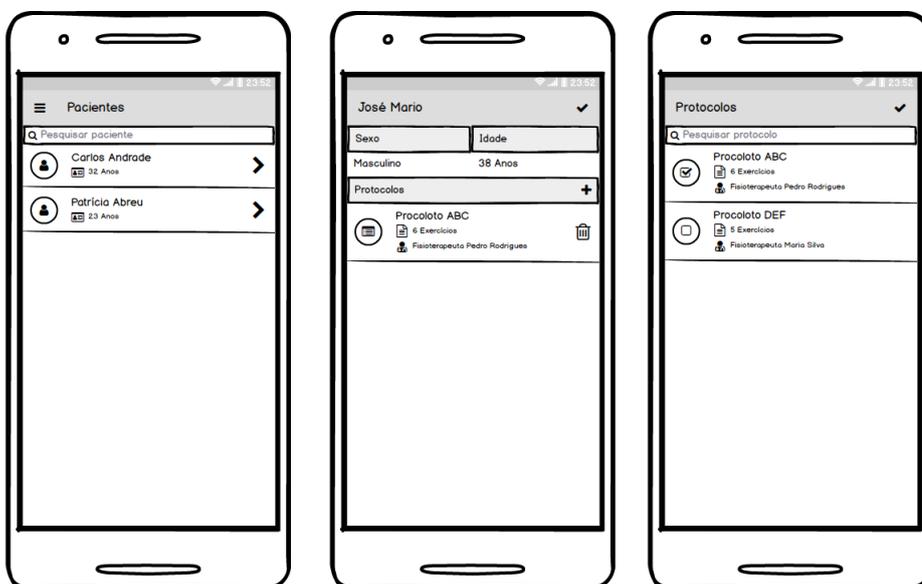


Fonte: Autor.

6.5.8 Busca de Pacientes

O fisioterapeuta também consegue visualizar todos os paciente que estão cadastrados no aplicativo, como vemos na tela da esquerda na Figura 16. Caso ele clique na pessoa lhe é exibida uma tela com mais detalhes sobre o paciente selecionado, como se vê na tela do meio da Figura 16. A tela do meio possibilita vincular protocolos àquela pessoa, clicando-se no botão ao lado direito da seção “Protocolos”, o que redireciona o fisioterapeuta para a tela da direita na mesma figura, onde ele visualiza os protocolos que podem ser associados ao paciente e os seleciona.

Figura 16 – Pacientes

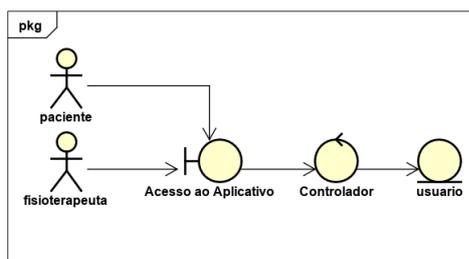


Fonte: Autor.

6.6 DIAGRAMAS DE ROBUSTEZ

Nesta seção são apresentados os diagramas de robustez referente ao aplicativo. Na Figura 17 se encontra o diagrama referente ao processo de acessar o aplicativo e também recuperar a senha.

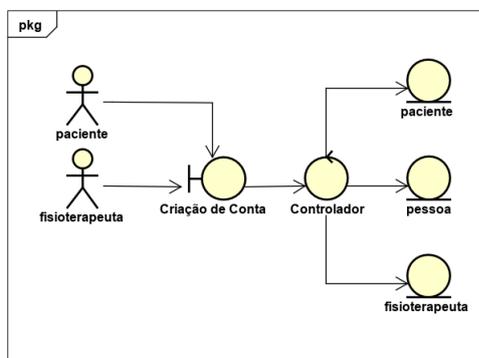
Figura 17 – Acesso ao aplicativo



Fonte: Autor.

Na Figura 18 está o diagrama do processo de criação e edição de uma conta.

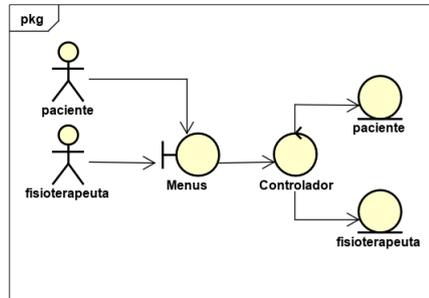
Figura 18 – Criação e edição da conta



Fonte: Autor.

A Figura 19 representa o diagrama relacionado aos menus do aplicativo.

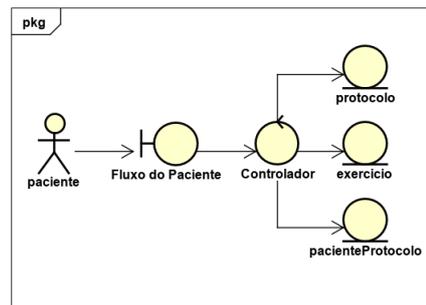
Figura 19 – Menus



Fonte: Autor.

A Figura 20 mostra o diagrama do fluxo de navegação que um paciente possui dentro do aplicativo.

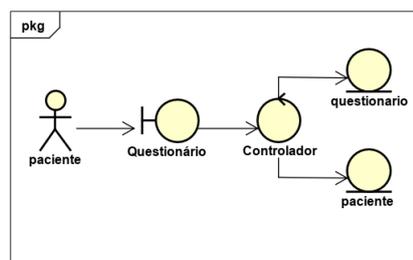
Figura 20 – Fluxo para um paciente



Fonte: Autor.

A Figura 21 representa o diagrama do questionário que é apresentado no aplicativo.

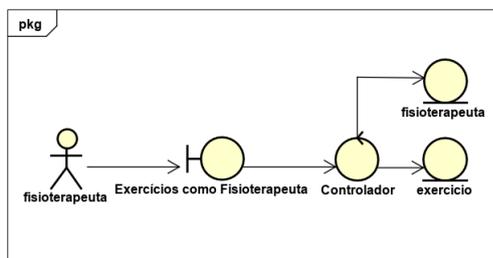
Figura 21 – Questionário



Fonte: Autor.

Na Figura 22 está o diagrama do fluxo de exercícios que um fisioterapeuta vê.

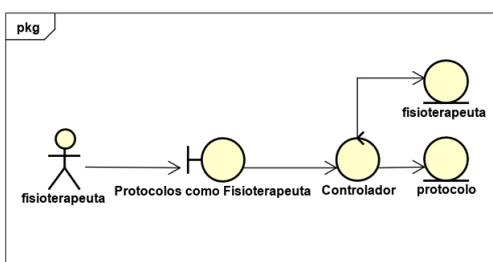
Figura 22 – Exercícios para um fisioterapeuta



Fonte: Autor.

A Figura 23 representa o diagrama do fluxo de protocolos que um fisioterapeuta entrega.

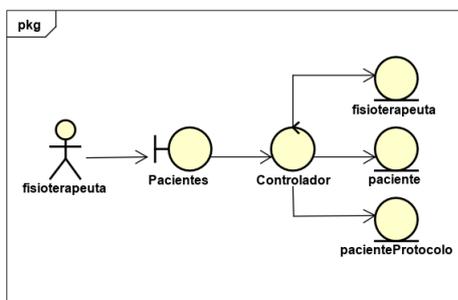
Figura 23 – Protocolos para um fisioterapeuta



Fonte: Autor.

Na Figura 24 está o diagrama da busca de pacientes e atribuição de protocolos aos mesmos.

Figura 24 – Pacientes



Fonte: Autor.

7 CONSIDERAÇÕES PARCIAIS

Neste trabalho foi realizada a modelagem de um sistema que auxilia o tratamento fisioterápico para os pacientes, também auxiliando o trabalho dos fisioterapeutas.

Espera-se que com o uso do *framework* Flutter seja obtida uma navegação muito fluida no aplicativo, sendo assim de fácil aceitação por seus usuários que possuirão uma boa experiência de criação e visualização dos exercícios.

Para o planejamento do projeto utilizou-se a metodologia ágil de desenvolvimento Ico-nix, observando as classes necessárias, suas telas, casos de uso e também requisitos. Visto que os artefatos necessários foram identificados, preparou-se assim o caminho para o desenvolvimento do trabalho durante a execução do Trabalho de Conclusão II.

7.1 CRONOGRAMA

A Figura 25 mostra o cronograma a ser seguido para o Trabalho de Conclusão II.

Figura 25 – Cronograma

	Ago/19	Set/19	Out/19	Nov/19	Dez/19
Desenvolvimento da documentação					
Desenvolvimento					
Testes					
Implantação					

Fonte: Autor.

REFERÊNCIAS

- ADOBE SYSTEMS INC. *PhoneGap*. 2016. Disponível em: <<https://phonegap.com/>>. Acesso em: 22 abr. 2019.
- APPBRAIN. *Android and Google Play statistics*. 2019. Disponível em: <<https://www.appbrain.com/stats>>. Acesso em: 21 abr. 2019.
- APPLE. *NSUserDefaults*. 2019. Disponível em: <<https://developer.apple.com/documentation/foundation/nsuserdefaults>>. Acesso em: 6 jun. 2019.
- BACK4APP. *Back4App Pricing*. 2019. Disponível em: <<https://www.back4app.com/pricing>>. Acesso em: 1 mai. 2019.
- BACK4APP. *Parse Server Made Simple | Back4App*. 2019. Disponível em: <<https://www.back4app.com/>>. Acesso em: 1 mai. 2019.
- BARKER, D. *Web Content Management*. Sebastopol, Califórnia: O'Reilly Media, 2016. 550 p. ISBN 978-1-4919-0807-5. Disponível em: <<http://booksdl.org/get.php?md5=416ac9fc0f64531ea27e47f226787d08>>. Acesso em: 1 mai. 2019.
- BELLINASSO, M. *Flutter: the good, the bad and the ugly*. 2018. Disponível em: <<https://medium.com/asos-techblog/flutter-vs-react-native-for-ios-android-app-development-c41b4e038db9>>. Acesso em: 24 abr. 2019.
- BÔRTOLE, M.; FILHO, T. F. B. Desenvolvimento de um sensor híbrido para aplicações em robótica e fisioterapia. Universidade Federal do Espírito Santo, 2011. Disponível em: <[doi:10.1002/pri.15](https://doi.org/10.1002/pri.15)>.
- CHARTERED SOCIETY OF PHYSIOTHERAPY. *What is physiotherapy?* 2018. Disponível em: <<https://www.csp.org.uk/careers-jobs/what-physiotherapy>>. Acesso em: 5 mai. 2019.
- CHOI, B. K. L. et al. Exercises for prevention of recurrences of low-back pain. *Occupational and Environmental Medicine*, BMJ Publishing Group Ltd, v. 67, n. 11, p. 795–796, 2010. ISSN 1351-0711. Disponível em: <[doi:10.1136/oem.2010.059873](https://doi.org/10.1136/oem.2010.059873)>.
- CIRIACO, D. *Os 5 melhores sistemas de gerenciamento de conteúdo gratuitos*. 2019. Disponível em: <<https://canaltech.com.br/internet/os-5-melhores-sistemas-de-gerenciamento-de-conteudo-gratuitos/>>. Acesso em: 10 jun. 2019.
- CREFITO-3. *Fisioterapia - Definição*. 2018. Disponível em: <<http://www.crefito3.org.br/dsn/fisioterapia.asp>>. Acesso em: 5 mai. 2019.
- EISENMAN, B. *Learning React Native: Building Native Mobile Apps with JavaScript*. Sebastopol, Califórnia: O'Reilly Media, 2015. 272 p.
- ESPLIN, C. *What is Firebase?* 2016. Disponível em: <<https://howtofirebase.com/what-is-firebase-fcb8614ba442>>. Acesso em: 1 mai. 2019.
- FACEBOOK INC. *React Native - A framework for building native apps using React*. 2019. Disponível em: <<https://facebook.github.io/react-native/>>. Acesso em: 22 abr. 2019.
- FISIOTERAPIA. In: DICIONÁRIO Online de Português. Porto: 7Graus, 2019. Disponível em: <<https://www.dicio.com.br/fisioterapia/>>. Acesso em: 5 mai. 2019.

FLYNN, A. Home exercise programs made effortless using the physiotherapy exercises patient app. *Journal of Physiotherapy*, Elsevier BV, v. 64, n. 2, p. 134–135, 2018. Disponível em: <doi:10.1016/j.jphys.2018.01.002>.

GARBADE, D. M. J. *Native vs. cross-platform app development: pros and cons*. 2018. Disponível em: <https://codeburst.io/native-vs-cross-platform-app-development-pros-and-cons-49f397bb38ac>. Acesso em: 15 abr. 2019.

GARGENTA, M. *Learning Android*. Sebastopol, Califórnia: O'Reilly Media, 2010. 320 p.

GOOGLE. *Ativar dados off-line*. 2019. Disponível em: <https://firebase.google.com/docs/firestore>. Acesso em: 11 jun. 2019.

GOOGLE. *Ativar dados off-line*. 2019. Disponível em: <https://firebase.google.com/docs/firestore/manage-data/enable-offline>. Acesso em: 11 jun. 2019.

GOOGLE. *FAQ - Flutter*. 2019. Disponível em: <https://flutter.dev/docs/resources/faq>. Acesso em: 24 abr. 2019.

GOOGLE. *Firebase Pricing*. 2019. Disponível em: <https://firebase.google.com/pricing>. Acesso em: 1 mai. 2019.

GOOGLE. *Firebase Products*. 2019. Disponível em: <https://firebase.google.com/products>. Acesso em: 1 mai. 2019.

GOOGLE. *Flutter - Beautiful native apps in record time*. 2019. Disponível em: <https://flutter.dev/>. Acesso em: 31 mar. 2019.

GOOGLE. *Opções de armazenamento*. 2019. Disponível em: <https://developer.android.com/guide/topics/data/data-storage>. Acesso em: 6 jun. 2019.

GYLLENSTEN, A. L. Interaction between patient and physiotherapist: a qualitative study reflecting the physiotherapists perspective. *Physiotherapy Research International*, Wiley, v. 4, n. 2, p. 89–109, 1999. Disponível em: <doi:10.1002/pri.156>.

HAMMOND, R. Evaluation of physiotherapy by measuring the outcome. *Physiotherapy*, Elsevier BV, v. 86, n. 4, p. 170–172, 2000. Disponível em: <doi:10.1016/s0031-9406(05)60959-5>.

IONIC. *Ionic - Cross-Platform Mobile App Development*. 2019. Disponível em: <https://ionicframework.com/>. Acesso em: 22 abr. 2019.

JAMIN, B. *Reasons Not To Use Firebase*. 2016. Disponível em: <https://crisp.chat/blog/why-you-should-never-use-firebase-realtime-database/>. Acesso em: 1 mai. 2019.

JENSEN, M. P.; KAROLY, P.; BRAVER, S. The measurement of clinical pain intensity: a comparison of six methods. *Pain*, Ovid Technologies (Wolters Kluwer Health), v. 27, n. 1, p. 117–126, 1986. Disponível em: <doi:10.1016/0304-3959(86)90228-9>.

JÚNIOR, J. P. B. Fisioterapia e saúde coletiva: desafios e novas responsabilidades profissionais. *Ciência & Saúde Coletiva*, Rio de Janeiro, 2007. ISSN 1678-4561. Disponível em: <https://www.scielo.org/scielo.php?pid=S1413-81232010000700074&script=sci_arttext>. Acesso em: 5 mai. 2019.

LELER, W. *Whats Revolutionary about Flutter*. 2017. Disponível em: <https://hackernoon.com/whats-revolutionary-about-flutter-946915b09514>. Acesso em: 31 mar. 2019.

- MAHARANA, N. *Cross Platform Mobile Apps and Its Pros and Cons*. 2017. Disponível em: <<https://medium.com/@netranandamaharana/cross-platform-mobile-apps-and-its-pros-and-cons-9c257ec64e94>>. Acesso em: 18 abr. 2019.
- MAHER, C. Effective physical treatment for chronic low back pain. *Orthopedic Clinics of North America*, Elsevier BV, v. 35, n. 1, p. 57–64, 2004. Disponível em: <doi:10.1016/S0030-5898(03)00088-9>.
- MALIK, N. *React Native: Pros and Cons*. 2018. Disponível em: <<https://citrusbits.com/react-native-pros-and-cons/>>. Acesso em: 24 abr. 2019.
- MELO, A. *Firestore vs. Parse Server*. 2016. Disponível em: <<https://blog.back4app.com/2016/06/15/firebase-parse/>>. Acesso em: 1 mai. 2019.
- MICROSOFT. *Desenvolvimento de Aplicativos Xamarin com Visual Studio | Visual Studio*. 2019. Disponível em: <<https://visualstudio.microsoft.com/pt-br/xamarin/>>. Acesso em: 22 abr. 2019.
- NEWGENAPPS. *Flutter vs React Native vs Ionic - the best tool for cross-platform apps*. 2018. Disponível em: <<https://www.newgenapps.com/blog/flutter-react-native-ionic-cross-platform-apps>>. Acesso em: 22 abr. 2019.
- NIEUWBOER, A. et al. The effect of a home physiotherapy program for persons with parkinson's disease. *Journal of Rehabilitation Medicine*, v. 33, n. 6, p. 266–272, 2001. Disponível em: <doi:10.1080/165019701753236455>.
- OCCHINO, T. *React Native: Bringing modern web techniques to mobile*. 2015. Disponível em: <<https://code.fb.com/android/react-native-bringing-modern-web-techniques-to-mobile/>>. Acesso em: 24 abr. 2019.
- PRITCHETT, A. *6 Pros and Cons of Cloud Storage for Business*. 2018. Disponível em: <<https://www.comparethecloud.net/articles/6-pros-and-cons-of-cloud-storage-for-business/>>. Acesso em: 1 mai. 2019.
- RAJPUT, M. *Top Mobile App Development Framework in 2019*. 2018. Disponível em: <<https://www.mindinventory.com/blog/mobile-app-development-framework-2019/>>. Acesso em: 18 abr. 2019.
- ROCKCONTENT. *Você sabe para que serve a fisioterapia?* 2019. Disponível em: <<https://blog.fisiotrauma.com.br/para-que-serve-a-fisioterapia/>>. Acesso em: 5 mai. 2019.
- ROSENBERG, D. *Agile development with ICONIX process : people, process, and pragmatism*. Berkeley, CA: Apress, 2005. 261 p. ISBN 9781430200093.
- ROUSE, M. *What is cross-platform mobile development?* 2018. Disponível em: <<https://searchmobilecomputing.techtarget.com/definition/cross-platform-mobile-development>>. Acesso em: 18 abr. 2019.
- ROUSE, M. *What is native app?* 2018. Disponível em: <<https://searchsoftwarequality.techtarget.com/definition/native-application-native-app>>. Acesso em: 15 abr. 2019.
- SHAIK, A. R.; SHEMJAZ, A. M. The rise of physical therapy: A history in footsteps. *Arch Med Health*, Karnataka, v. 2, p. 257–260, 2014. ISSN 2321-6085. Disponível em: <doi:10.4103/2321-4848.144367>. Acesso em: 5 mai. 2019.

SINICKI, A. *Developing for Android vs developing for iOS in 5 rounds*. 2016. Disponível em: <<https://www.androidauthority.com/developing-for-android-vs-ios-697304/>>. Acesso em: 22 abr. 2019.

SOMMERVILLE, I. *Engenharia de Software*. Upper Saddle River, Nova Jersey: Pearson Prentice Hall, 2011. 544 p. ISBN 978-85-7936-108-1.

STATCOUNTER. *Mobile Operating System Market Share Worldwide*. 2019. Disponível em: <<http://gs.statcounter.com/os-market-share/mobile/worldwide#monthly-201803-201903>>. Acesso em: 21 abr. 2019.

STEEL MEDIA. *App Store Metrics*. 2019. Disponível em: <<https://www.pocketgamer.biz/metrics/app-store/>>. Acesso em: 21 abr. 2019.

SYSGEN. *The Pros and Cons of Cloud vs in House Servers*. 2018. Disponível em: <<https://sysgen.ca/cloud-vs-in-house-servers/>>. Acesso em: 29 abr. 2019.

TAMPLIN, J. *Firebase is Joining Google!* 2014. Disponível em: <<https://firebase.googleblog.com/2014/10/firebase-is-joining-google.html>>. Acesso em: 1 mai. 2019.

TRIKHA, B. A journey from floppy disk to cloud storage. (*IJCSE*) *International Journal on Computer Science and Engineering*, v. 2, n. 4, p. 1449–1452, 2010. ISSN 0975-3397. Disponível em: <<http://www.enggjournals.com/ijcse/doc/IJCSE10-02-05-24.pdf>>. Acesso em: 1 mai. 2019.