

FaleMed:

**Um sistema para armazenar documentos médicos e
melhorar a interação entre o profissional da saúde e
seu paciente**

Leandro Paiva Andrade



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa - PB

Dezembro - 2016

Leandro Paiva Andrade

FaleMed:

Um sistema para armazenar documentos médicos e
melhorar a interação entre o profissional da saúde e
seu paciente

Monografia apresentada ao curso Ciência da Computação
do Centro de Informática, da Universidade Federal da Paraíba,
como requisito para a obtenção do grau de Bacharel em Ciências da Computação

Orientador: Eudisley Gomes dos Anjos

Dezembro de 2016

Ficha Catalográfica elaborada por
Rogério Ferreira Marques CRB15/690

A553f

Andrade, Leandro Paiva.

FaleMed: um sistema para armazenar documentos médicos e
melhorar a interação entre o profissional da saúde e seu paciente /
Leandro Paiva Andrade. – João Pessoa, 2017.

61p. : il.

Monografia (Bacharelado em Ciência da Computação) –
Universidade Federal da Paraíba - UFPB.

Orientador: Prof^o. Dr. Eudisley Gomes dos Anjos.

1. Sistemas abertos. 2. Saúde. 3. Multiplataforma. 4. Armazenamento
de dados. I. Título.

UFPB/BSCI

CDU: 004.057.8 (043.2)



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Ciência da Computação intitulado ***FaleMed*** de autoria de Leandro Paiva Andrade, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Me. Eudisley Gomes dos Anjos
Universidade Federal da Paraíba

Prof. Dr. Danielle Rousy Dias da Silva
Universidade Federal da Paraíba

Prof. Dr. Thaís Gaudêncio do Rêgo
Universidade Federal da Paraíba

Coordenador(a) do Departamento de Informática
Daniela Coelho Batista Guedes Pereira
CI/UFPB

João Pessoa, 12 de Dezembro de 2016

*"Vencer é nunca desistir."
(Albert Einstein)*

DEDICATÓRIA

Dedico este trabalho aos meus amados pais, Geraldo e Betania, à minha amiga e companheira Karina e a todos aqueles que buscam através da criatividade a solução de problemas.

AGRADECIMENTOS

A Deus, por me fazer acreditar que nada é impossível quando se trabalha com foco, fé e determinação.

Aos meus pais, pelo amor, confiança, apoio e todo o suporte durante minha formação.

Ao meu orientador, professor Me. Eudisley Gomes dos Anjos, por todo o conhecimento que me foi repassado, pela paciência, disponibilidade e incentivo.

A todos os professores do curso de Ciências da Computação da UFPB, que foram tão importantes na minha vida acadêmica e no desenvolvimento desta monografia.

Aos familiares, namorada, amigos e a todos que direta ou indiretamente fizeram parte da minha formação.

RESUMO

O uso de tecnologias no nosso cotidiano está cada vez mais frequente, principalmente o uso de *smartphones* e *tablets*, que podem servir simultaneamente como dispositivos de comunicação e de armazenamento de informações. A indústria da saúde é uma das que mais crescem nesse setor de dispositivos móveis, tentando tornar cada vez mais fácil e prático o acesso de conteúdos e processos relacionados a essa área. Este trabalho descreve a criação de um sistema multiplataforma que facilite procedimentos médicos comuns, como consultas. O FaleMed traz a proposta de tornar mais prática e ágil a integração entre o profissional da saúde e seu paciente através da criação de um sistema capaz de gerenciar conversas entre profissional-paciente, armazenar dados médicos de forma organizada e cronológica, que podem ser compartilhados entre ambos, e que possui uma rede social onde é possível adicionar usuários e ver conteúdos com informações educativas, gerados pelos profissionais registrados.

Palavras-chave: Saúde. Multiplataforma. Armazenamento de dados. Web. Mobile.

ABSTRACT

The use of technologies in our daily lives is increasingly frequent, especially the use of smartphones and tablets that can be used simultaneously as communication and information storage devices. The health industry is one of the fastest growing in this sector of mobile devices, trying to make it easier and more practical to access content and processes related to this area. This paper describes the creation of a multi platform system that facilitates common medical procedures, such as consultations. FaleMed brings the proposal of making the integration between the health professional and his patient more practical and agile through the creation of a system capable of managing professional-patient conversations, storing medical data in an organized and chronological manner, which can be shared between both, and which has a social network where it is possible to add users and view content with educational information generated by professionals registered in the system.

Key-words: Health. Multi platform Storage data. Web. Mobile

LISTA DE FIGURAS

Figura 1	<i>Xcode Interface</i>	22
Figura 2	Interface do sistema Clínica nas Nuvens.	26
Figura 3	Sistema Saúde Controle. (A) Versão <i>web</i> e (B) Versão <i>Mobile</i>	27
Figura 4	Interfaces do sistema Consultório na Nuvem. (A) Versão <i>web</i> , (B) Versão <i>Mobile</i>	28
Figura 5	Interface do Sistema Saúde na Nuvem.	29
Figura 6	Fases e Estágios do <i>Mobile-D</i>	32
Figura 7	Organograma das Etapas da Metodologia	34
Figura 8	Diagrama de Classes	36
Figura 9	Arquitetura do FaleMed	37
Figura 10	Prototipo de Telas. (A) Primeira Sequência de telas, (B) Segunda Sequência de telas	38
Figura 11	Tela de <i>login</i>	39
Figura 12	Tela de Cadastro	39
Figura 13	Tela de Home	40
Figura 14	Tela de lista de Profissionais da Saúde	41
Figura 15	Tela de Perfil do Profissional	41
Figura 16	Tela de lista de Conversas	42
Figura 17	Tela de lista do Histórico Médico. (A) Tela de Histórico, (B) Tela de Exames e (C) Tela de Documentos Salvos	43
Figura 18	Tela de Configurações	44
Figura 19	Tela de <i>login web</i>	45
Figura 20	Tela Inicial <i>web</i>	45
Figura 21	Tela de Pacientes <i>web</i>	46
Figura 22	Tela de Detalhes de Documentos <i>web</i>	46

LISTA DE TABELAS

Tabela 1	Comparativo entre os Sistemas	29
Tabela 2	Cronograma de Atividades	34

LISTA DE ABREVIATURAS

- 3G – Terceira Geração
- AGPL - *Affero General Public License* (Licença Pública Geral Affero)
- API - *Application Program Interface* (Interface do Programa de Aplicação)
- ASCII - *American Standard Code for Information Interchange* (Código Padrão Americano para Intercâmbio de Informações)
- BaaS - *Backend as a Service* (Backend como um serviço)
- BSON - *Binary Structured Object Notation* (Notação de Objeto Estruturado Binário)
- CSS - *Cascading Style Sheet* (Folha de Estilo em Cascata)
- HTML - *HyperText Markup Language* (Linguagem de marcação de hipertexto)
- HTTPS - *Hypertext Transfer Protocol Secure* (Seguro de Protocolo de Transferência de Hipertexto)
- IaaS - *Infrastructure as a Service* (Infra-estrutura como Serviço)
- IBGE - Instituto Brasileiro de Geografia e Estatística
- ID - *Identification* (Identificação)
- IDE - *Integrated Development Environment* (Ambiente de Desenvolvimento Integrado)
- IaaS - *Infrastructure as a Service* (Infraestrutura como serviço)
- iOS - *iPhone Operating System* (Sistema operacional do iPhone)
- JSON - *JavaScript Object Notation* (Notação de Objetos JavaScript)
- LLVM - *Low Level Virtual Machine* (Máquina virtual de baixo nível)
- MBaaS - *Mobile Backend as a Service* (Backend móvel como um serviço)
- MVC - *Model View Control* (Controle de exibição de modelo)
- NoSQL - *Not Only Structured Query Language* (Não só linguagem de consulta estruturada)
- OSX - *Operating System X* (Sistema Operacional X)
- PaaS - *Platform as a Service* (Plataforma como Serviço)
- RC4 - *Rivest Cipher 4*
- REPL - *Read-Eval-Print-Loop*
- REST - *Representational State Transfer* (Transferência Representacional de Es-

tado)

SaaS - *Software as a Service* (Software como serviço)

SDK - *Software Development Kit* (Kit de desenvolvimento de Software)

TDD - *Test-Driven Development* (Desenvolvimento Test-Driven)

UFPB - Universidade Federal da Paraíba

W3C - *World Wide Web Consortium* (Consórcio da World Wide Web)

WWDC14 - *Worldwide Developer Conference 2014* (Conferência Mundial de Desenvolvedores 2014)

Sumário

1	INTRODUÇÃO	18
1.0.1	Objetivo geral	19
1.0.2	Objetivos específicos	19
1.1	Estrutura da monografia	19
2	CONCEITOS GERAIS E REVISÃO DA LITERATURA	21
2.1	Referencial Teórico	21
2.1.1	<i>Swift</i>	21
2.1.2	<i>Xcode</i> (IDE)	22
2.1.3	<i>Parse Server</i>	22
2.1.4	MongoDB	23
2.1.5	HTML	24
2.1.6	CSS	25
2.1.7	Angular 2	25
2.2	Estado da Arte	26
2.2.1	Clínica nas Nuvens	26
2.2.2	Saúde Controle	27
2.2.3	Consultório na Nuvem	27
2.2.4	Saúde na Nuvem	28
2.3	Análise do Estado da Arte	29
3	MATERIAS E MÉTODOS	31
3.1	<i>Mobile-D</i>	31
3.2	<i>Scrum</i>	32
3.3	Etapas da Metodologia	33
4	DESENVOLVIMENTO	35
4.1	Resultados	37
4.1.1	Interfaces	37
5	CONCLUSÃO E TRABALHOS FUTUROS	47
	REFERÊNCIAS	47
	APÊNDICE A - Interfaces aplicação mobile versão profissional	51
	APÊNDICE B - Questionário de Levatamento de Requisitos do Sistema	53
	APÊNDICE C - Documento de Requisitos	53

1 INTRODUÇÃO

Atualmente observa-se uma crescente imersão da população no ambiente digital. Este aumento está relacionado à grande acessibilidade aos dispositivos móveis e também à facilidade de acesso à internet por meio de tecnologias como 3G ou *WiFi*. Segundo uma pesquisa divulgada pelo Instituto Brasileiro de Geografia e Estatística (IBGE) em 2014, o uso do telefone celular para acesso à internet no país ultrapassou o por computadores [1].

A utilização de dispositivos como *smartphones* e *tablets* vem se popularizando cada vez mais, o que implica na crescente demanda por conteúdo móvel. Hoje em dia, existem no mercado, aplicações voltadas para as mais diversas áreas, como transporte, educação, saúde, entre outras. Na área da saúde, as lojas para aplicativos *App Store* e *Google Play* oferecem uma ampla variedade de opções, que englobam desde aplicações voltadas para nutrição e prática de esportes, até aquelas que auxiliam para o controle da ingestão de remédios, reúne o histórico médico do paciente, e até mesmo que ajudam no controle de doenças.

De acordo com uma pesquisa realizada pela Flurry Insights, pertencente ao Yahoo!, no ano de 2013, enquanto a indústria global de aplicativos para celular cresceu 115% em termos de uso médio diário, a categoria de saúde e fitness só cresceu 49%. Mas durante os seis primeiros meses de 2014, foi visto um aumento de 62% no uso de aplicativos de saúde e fitness. Isso se compara ao aumento de 33% no uso, medido em sessões, para a indústria de aplicativos móveis em geral. O crescimento em saúde e fitness é 87% mais rápido do que outras indústrias. O estudo foi realizado em mais de 6.800 aplicativos para iPhone e iPad listados na categoria de saúde e fitness na plataforma da Flurry[2]

Por outro lado, o mundo atual vivencia cada vez mais a cultura da pressa. Existe, portanto, uma crescente busca por ferramentas que valorizem o tempo, e tornem o dia-a-dia da população mais prático. Neste contexto, as grandes filas de espera e a demora de atendimento em consultórios e postos médicos pode ser encarada como um grande problema. Em toda consulta, é exigido que o profissional da saúde siga um protocolo, que envolve a realização de perguntas como todo o histórico de saúde do paciente e exames já realizados anteriormente, o que acarreta um grande consumo de tempo por consulta.

Outro problema muito frequente é a dificuldade de comunicação entre o profissional da saúde e o paciente no pós consulta. Muitas vezes, após o paciente deixar o consultório, surgem dúvidas relacionadas à pedidos de exames, receitas para a compra de medicamentos, entre outros. Neste caso, o comum é que o paciente entre em contato com a clínica, repasse suas dúvidas aos funcionários e peça para que eles perguntem ao médico num momento conveniente, o que pode demorar muito ou até mesmo não acontecer.

Uma forma de tornar todos estes procedimentos mais práticos e rápidos é por meio da utilização de uma aplicação que permita a comunicação direta entre o paciente e seu médico, e onde o profissional já possua o acesso prévio à todas as informações do paciente, de forma que ele possa acessar os exames e outros documentos, bem como estudar o caso de cada paciente quando e onde achar mais conveniente.

O termo mHealth é uma abreviação para “*Mobile Health*” ou “saúde móvel”, e representa a prática da medicina e da saúde pública por meio de dispositivos móveis [3]. Este termo foi designado por Robert Istepanian em 2006, e refere-se à utilização de qualquer tecnologia móvel que é voltada para os desafios da área da saúde, como acessibilidade, qualidade, correspondência de recursos e normas comportamentais [4, 5]. A aplicação proposta por este trabalho, chamada FaleMed aborda todas as possibilidades mencionadas.

1.0.1 Objetivo geral

O objetivo deste trabalho consiste no desenvolvimento de um sistema multiplataforma que interage com uma Interface do Programa de Aplicação (do inglês *Application Program Interface* - API) *open-source*, e tem como finalidade, facilitar a troca de informações e a interação entre profissionais da saúde e seus pacientes.

1.0.2 Objetivos específicos

Entender, na prática, o que é ser um desenvolvedor *full-stack*, por meio de toda a criação de um sistema, desde a concepção das interfaces até o desenvolvimento do aplicativo e interação com o banco de dados.

Melhorar o aprendizado sobre gerência e desenvolvimento de projetos através da utilização das metodologias Scrum e Mobile-D.

Conhecer a fundo as tecnologias utilizadas no desenvolvimento de um sistema, visando a preparação para o mercado de trabalho.

Colocar em prática os principais tópicos vistos nas disciplinas ao longo do curso de Ciência da Computação.

1.1 Estrutura da monografia

Este trabalho é dividido nos seguintes capítulos: conceitos gerais e revisão da literatura, metodologia, desenvolvimento e conclusão. O Capítulo 2, conceitos gerais e revisão da literatura, agrupa todo o conteúdo referente ao estado da arte do projeto e a revisão das tecnologias envolvidas. A metodologia descreve todos os passos e métodos

utilizados para o desenvolvimento do projeto e pode ser encontrada no Capítulo 3. O Capítulo 4 trata do desenvolvimento do sistema e refere-se aos artefatos que foram gerados durante este trabalho. Por fim, a conclusão apresentada no Capítulo 5, apresenta uma avaliação dos objetivos que foram alcançados, os conhecimentos aprimorados durante todo o processo e propostas para trabalhos futuros.

2 CONCEITOS GERAIS E REVISÃO DA LITERATURA

2.1 Referencial Teórico

Nesta seção, serão brevemente descritos os conceitos relacionados às tecnologias envolvidas no desenvolvimento do aplicativo FaleMed.

2.1.1 *Swift*

Lançado em junho de 2014 pela Apple, *Swift* é uma linguagem de programação multiparadigma criada para o desenvolvimento de programas para iOS e Mac OSX, e que utiliza a infra-estrutura do compilador LLVM e o tempo de execução *Objective-C* [6]. Como o *Swift* usa o mesmo tempo de execução do *Objective-C*, as duas linguagens podem ser combinadas em um único programa ou projeto, de forma que uma delas é usada como *framework* para a outra, sendo então compiladas juntamente. A linguagem *Swift* pode acessar classes em *Objective-C*, tipos, funções e variáveis através de um “*header bridge*”, bem como por extensão C e código C++. *Swift* é fortemente influenciada por muitas outras linguagens de programação, como *Rust*, *Haskell*, *Ruby*, *Python* e *C#*, e oferece muitos recursos voltados a objetos e funcionalidades encontrados nelas. *Swift* também inclui o *read-eval-print-loop* (REPL), ambiente simples e interativo que pode ser acessado no Xcode ou na linha de comando, e onde é possível executar códigos facilmente[7].

Durante a Conferência Mundial de Desenvolvedores de 2014 (do inglês Worldwide Developer Conference 2014 - WWDC14), a Apple afirmou que o *Swift* é capaz de executar aplicativos de forma “ultra-rápida”. No decorrer da apresentação, foram mencionados dados relativos à velocidade do *Swift*, e os mesmos foram comparados com os dados das linguagens Python e *Objective-C* para a ordenação de objetos complexos e para criptografia RC4.

Os dados exibidos pela Apple durante a Conferência, mostraram que *Objective-C* e *Swift* foram, respectivamente, 2,8 e 3,9 vezes mais rápidos que o Python para classificar objetos complexos. Desta forma, pode-se afirmar também que *Swift* é aproximadamente 1,4 vezes mais rápido do que *Objective-C*. Posteriormente, realizou-se a comparação destas linguagens para criptografia RC4. Segundo os dados apresentados, *Objective-C* e *Swift* foram, respectivamente, 127 e 220 vezes mais rápidos do que Python, o que significa que *Swift* é 1,73 vezes mais rápido do que *Objective-C* para criptografia RC4 [6].

Atualmente a linguagem *Swift* pode ser encontrada na versão 3.0.1, a qual se tornou *open source*¹ em dezembro de 2015. Como esta é a linguagem de desenvolvimento nativa para o sistema iOS, ela foi escolhida para a realização deste trabalho. Isto porque para

¹<https://github.com/apple/swift>

realização dos objetivos mencionados fez-se necessário o desenvolvimento de um aplicativo voltado para o sistema iOS. A versão *Swift* 3.0 foi utilizada devido a experiências prévias com a mesma.

2.1.2 Xcode(IDE)

Xcode é o ambiente de desenvolvimento integrado da Apple (IDE), utilizado na criação de aplicativos para produtos Apple, incluindo o iPad, iPhone, Apple Watch e Mac. O *Xcode* fornece diversas ferramentas de gerenciamento que podem ser utilizadas na criação das aplicações, e também nos testes, otimização e submissão na App Store[8]. Como pode ser visto na 1, na interface do *Xcode* é possível, em uma única janela, trabalhar com a edição de código, design de interface, gerenciamento de arquivos, teste e depuração de forma personalizada. O *Xcode* possibilita a criação das interfaces das aplicações de forma simplificada, de maneira que é necessário apenas clicar e arrastar os objetos para a posição desejada, e definir suas restrições de posições.

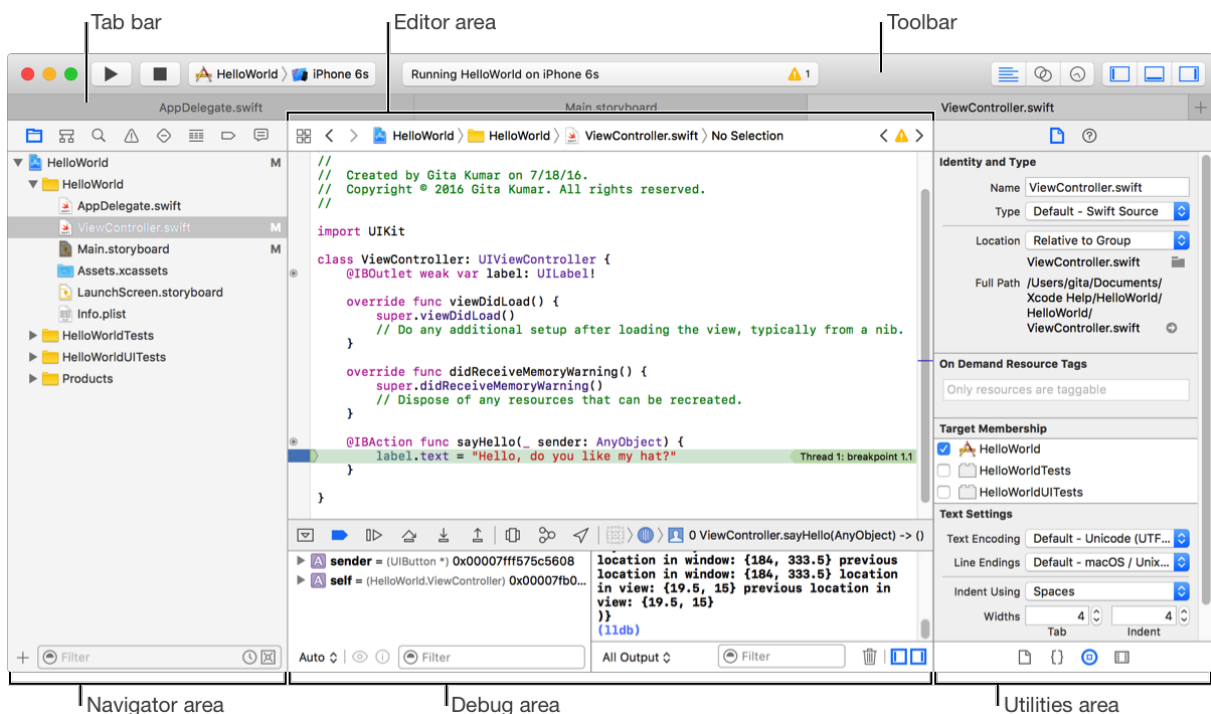


Figura 1: *Xcode* Interface. Fonte: [8]

2.1.3 Parse Server

O *Parse*, nome original do *Parse Server*, foi um Backend como Serviço (do inglês *Backend-as-a-Service* - BaaS) que foi adquirida pelo Facebook em 2013 [9]. *Backend-as-a-Service* (BaaS) é um modelo de serviço de computação em nuvem que funciona como o *middleware*, fornecendo aos desenvolvedores maneiras de conectar aplicações *web*

e *mobile* para serviços por meio de interfaces de programação de aplicativos (API) e kits de desenvolvedores de software (SDK) [10]. O BaaS também é conhecido como *Mobile-Backend-as-a-Service* (MBaaS). O modelo BaaS permite que o desenvolvedor se concentre no *frontend*, enquanto o *backend* está sendo cuidado pelo provedor BaaS.

As características de um BaaS incluem armazenamento na nuvem, notificações remotas, codificação do servidor, gerenciamento de usuários e arquivos, integração com redes sociais, serviços de localização, bem como muitos outros serviços de *backend*. Algumas características do BaaS são semelhantes aos de outros modelos de serviços, como SaaS, IaaS e PaaS. No entanto, BaaS é o único que satisfaz as necessidades de aplicações móveis e *web*. Em janeiro de 2016, um ano após a data de seu anúncio de encerramento, o Parse divulgou o fechamento de suas atividades e tornou-se *open source*: o *Parse Server* [11]. Este é construído em Node.js² e trabalha com o *framework* de aplicações *web* Express. O *Parse Server* é uma *RESTful* API e fornece o SDK para iOS e Android, o que facilita a forma de acesso às informações do *backend* no desenvolvimento da aplicação.

Para o desenvolvimento do presente trabalho, utilizou-se o *Parse Server*, o qual foi hospedado no Heroku (www.heroku.com), uma plataforma de aplicações na nuvem, ou *Platform-as-a-Service* (PaaS), que suporta diversas linguagens de programação e vem sendo utilizada como um modelo de implantação de aplicativos *web* [12]. O Heroku funciona por meio de planos que são categorizados de acordo com as necessidades de cada aplicação, e possui também uma plano grátis que é ideal para testes de aplicações em nuvem, o qual fez-se uso neste projeto. Escolheu-se esta plataforma devido à sua boa reputação e a experiências prévias com a mesma.

2.1.4 MongoDB

O MongoDB é um banco de dados orientado a documentos, livre de esquema, escrito em C++, sob a licença AGPL versão 3.0. Foi desenvolvido em um projeto *open source*, impulsionado principalmente pela empresa 10gen. Atualmente, o MongoDB continua sendo mantido pela mesma empresa, está na versão 2.2.2 e em constante desenvolvimento. O seu nome, Mongo, deriva do adjetivo *humongous*, monstruoso em português [13].

De acordo com seus desenvolvedores, o objetivo principal do MongoDB é preencher a lacuna entre os bancos de dados de armazenamento chave/valor, rápidos e altamente escaláveis, e os tradicionais sistemas de gerenciamento de banco de dados relacionais, que possuem grande quantidade de recursos. Assim, o MongoDB foi projetado para otimização de desempenho, facilidade de escalabilidade horizontal, alta disponibilidade e capacidade de consultas avançadas [14].

²Plataforma para desenvolvimento de aplicações no lado do servido baseadas em rede utilizando JavaScript e o V8 JavaScript Engine.

Quanto ao armazenamento, de forma simplificada pode-se dizer que o servidor MongoDB contém um ou mais bancos de dados, independentes, armazenados separadamente. Cada banco de dados é composto por uma ou mais coleções constituídas de documentos BSON (Binary JSON), que são estruturados como documentos 36 JSON, com esquema dinâmico, fazendo com que a integração de dados em certos tipos de aplicações seja mais fácil e rápida. Como já mencionado, os documentos no MongoDB são organizados em coleções. Cada coleção pode conter qualquer tipo de documento, entretanto aconselha-se utilizar uma coleção para cada tipo de documento. Cada documento tem um campo de ID, que é usado como uma chave primária ObjectID. Estes índices são armazenados em estruturas como árvore B. Para permitir consultas rápidas, o desenvolvedor pode criar um índice para cada campo de consulta possível um documento[13].

2.1.5 HTML

O HTML, Linguagem de Marcação de Hipertexto (do inglês *HyperText Markup Language*), é uma linguagem de marcação declarativa que permite criar páginas e aplicações *web*. Um arquivo HTML é formado por um conjunto de marcações (do inglês *tags*) que definem a estrutura de uma página, e para a sua criação necessita-se de apenas um editor de texto Código Padrão Americano para Intercâmbio de Informações (do inglês *American Standard Code for Information Interchange - ASCII*). A linguagem possibilita a produção de documentos estruturados por meio de uma notação semântica estrutural para o texto, como cabeçalhos, parágrafos, listas, links, citações e outros itens. Para verificar a aparência da página gerada pelo código de marcação durante a fase de desenvolvimento, basta salvar o arquivo com a extensão *.html* ou *.htm* e usar um browser *web* como *Mozilla*, *Chrome* ou *Safari* para abrir o arquivo. Os browsers não mostram as *tags* HTML, mas as utilizam para interpretar seu conteúdo[15].

Atualmente, o crescente uso de equipamentos que possuem capacidade de se conectar à internet, tais como smartphones, computadores, televisores, entre outros, implica na necessidade de evolução contínua da linguagem HTML. A última atualização da linguagem HTML, chamada de HTML5, contou com um vasto conjunto de novas *tags* e APIs, como, por exemplo, a validação de formulários, para aumentar ainda mais a potencialidade da linguagem[16]. Muitos recursos do HTML5 têm sido produzidos visando a sua execução em dispositivos de baixa potência, como smartphones e tablets. HTML pode incorporar programas escritos em linguagem de *script*, como *JavaScript*, que afetam o comportamento e o conteúdo das páginas da *web*, e também pode haver a inclusão de CSS, o qual define a aparência e o *layout* do conteúdo. O *World Wide Web Consortium* (W3C) é o órgão responsável por manter os padrões do HTML e CSS.

Escolheu-se o HTML5 para o desenvolvimento deste trabalho, porque a maioria dos navegadores atuais dão suporte à esta linguagem, e também por sua integração com

o *framework* Angular 2.

2.1.6 CSS

O CSS (*Cascading Style Sheets*), que no português significa folhas de estilo em cascata, é uma linguagem usada para estilizar as páginas *web* criadas com HTML[17], e é projetado principalmente para permitir a separação entre o conteúdo do documento e sua forma de apresentação, incluindo aspectos como o *layout*, cores e fontes. Esta separação pode melhorar a acessibilidade do conteúdo, fornecer mais flexibilidade e controle na especificação das características de apresentação, permitindo que múltiplas páginas HTML possam compartilhar a mesma formatação, que deve ser especificada em um arquivo *.css* em separado, reduzindo a complexidade e repetição no conteúdo estrutural[18].

Atualmente, o CSS encontra-se em sua terceira versão, sendo denominado CSS3, o qual trouxe diversas novas funcionalidades em comparação com a sua versão anterior. Uma das inovações mais importantes é a facilidade de criação de uma aplicação responsiva, atividade auxiliada pela definição de *Media Queries*. As *Media Queries* permitem que o CSS seja capaz de consultar o navegador e verificar qual é o tipo de dispositivo que o usuário está utilizando para acessar um site, permitindo que estilos diferentes sejam especificados para cada tipo de tela. Basicamente, a responsividade busca adequar os elementos e os comportamentos da aplicação para diferentes cenários conforme as dimensões disponíveis no aparelho do usuário[19].

A linguagem CSS3 foi escolhida para o desenvolvimento deste projeto por se tratar de um complemento indispensável para a HTML5. Sua utilização tem como principal finalidade auxiliar na apresentação do website para o usuário, buscando melhorar o visual e a usabilidade do sistema.

2.1.7 Angular 2

Angular 2 é um *framework* de aplicações *web front-end* e *open source*, que visa simplificar o desenvolvimento e testes de aplicações *web*, fornecendo uma estrutura *Model-View-Control*(MVC) no lado cliente. O Angular 2 é mantido principalmente pelo Google e por uma comunidade online, composta por voluntários que buscam soluções para os desafios encontrados no desenvolvimento de aplicações *web*[20].

O Angular 2 foi anunciado durante a conferência ng-Europe, que ocorreu em Paris entre os dias 22 e 23 de Setembro de 2014, mas sua versão final foi lançada apenas em 14 de setembro de 2016. Angular 2 não é uma atualização do AngularJS, mas sim uma reescrita completa deste *framework*, sendo que ambos não possuem compatibilidade. A linguagem nativa para versão Angular 2 é *TypeScript*.

Diferentemente do AngularJS, o Angular 2 possibilita o desenvolvimento *mobile* com auxílio de ferramentas como o *Ionic Framework*, *NativeScript*, e *React Native*, a redução dos problemas de compatibilidade com os browsers modernos, modularização dos componentes, entre outros[21].

2.2 Estado da Arte

Antes de se desenvolver qualquer tipo de sistema, é necessário verificar o que já existe de similar em funcionamento. Com as frequentes inovações na área da saúde, já existem no mercado vários sistemas que buscam ajudar no gerenciamento das informações médicas, e alguns deles serão apresentados nas subseções a seguir.

2.2.1 Clínica nas Nuvens

O Clínica nas Nuvens é um sistema online voltado para o controle interno de clínicas e consultórios. O sistema possui como principais funcionalidades a interação com pacientes, multiclínicas e multiusuários, galeria de imagens e documentos, prontuário eletrônico, multi agendas, compatibilidade com dispositivos móveis, dentre outras funcionalidades[22]. A Figura 2 apresenta a interface deste sistema.

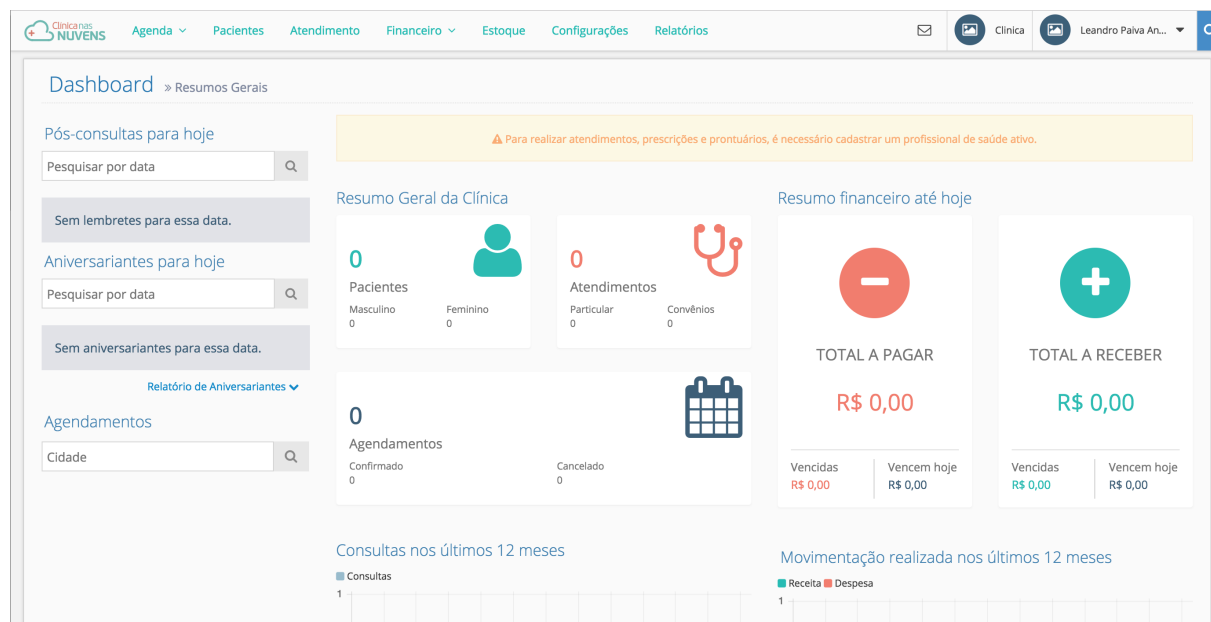


Figura 2: Interface do sistema Clínica nas Nuvens. Fonte: [22]

Até o presente momento, o sistema possui uma única interface, a qual é direcionada somente para a utilização de clínicas, ou seja, apenas o médico e seus funcionários possuem acesso aos documentos dos pacientes e aos registros de consultas.

2.2.2 Saúde Controle

O Saúde Controle é um sistema médico de armazenamento de dados que não é voltado para profissionais da área de saúde, e sim para pacientes. Ele oferece uma plataforma de soluções *web* e *mobile*, e suas principais funções são o armazenamento de informações como exames, fichas médicas, vacinas, medicamentos, linha do tempo, entre outros[23]. A Figura 3 mostra a interface deste sistema.

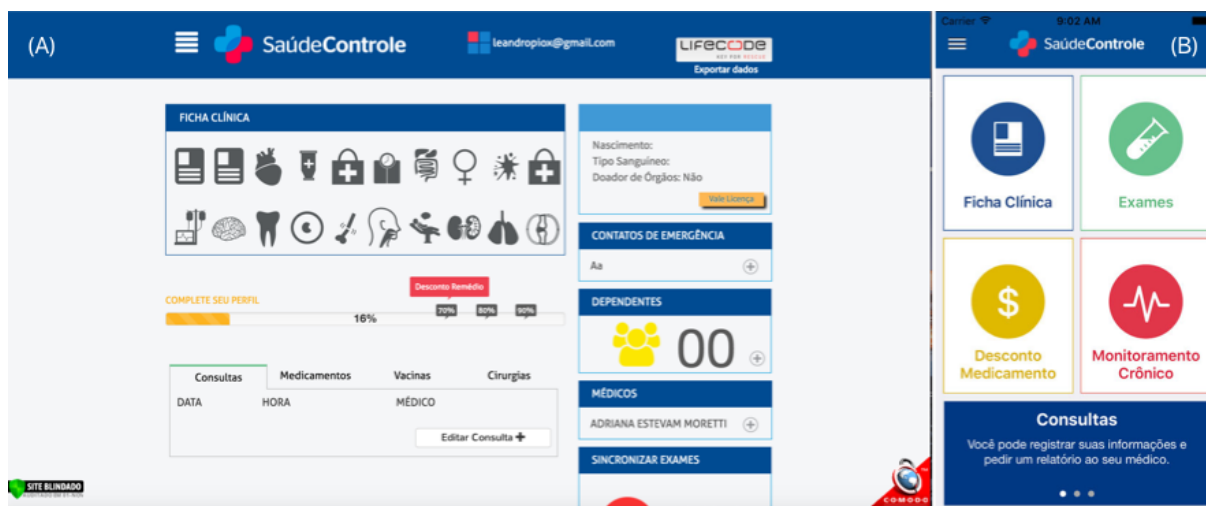


Figura 3: Sistema Saúde Controle. (A) Versão *web* e (B) Versão *Mobile*. Fonte: [23]

Na aplicação *mobile*, especificamente a versão iOS, o sistema não segue um padrão de interface para aplicações iOS. Tanto na App Store quanto na Google Play existe uma série de reclamações sobre erros, tais como problemas no armazenamento de dados, imagens salvas com baixa resolução, entre outros[24, 25].

2.2.3 Consultório na Nuvem

O Consultório na Nuvem é um sistema inteiramente online para registro e arquivo de informações médicas de consultórios. Possui as plataformas *web* e *mobile*. As principais funcionalidades são o acesso aos dados do agendamento, atendimentos, registros de exames, solicitação de exames, prescrições e atestados[26].

Este sistema é voltado para a utilização de clínicas, médicos e pacientes. Como pode ser visualizado na Figura 4, o foco do sistema é o cadastro de pacientes e o agendamento de consultas para os mesmos. Foi verificado que existem alguns erros na versão *mobile* deste sistema para iOS, como por exemplo, a dificuldade de acesso à área do paciente.

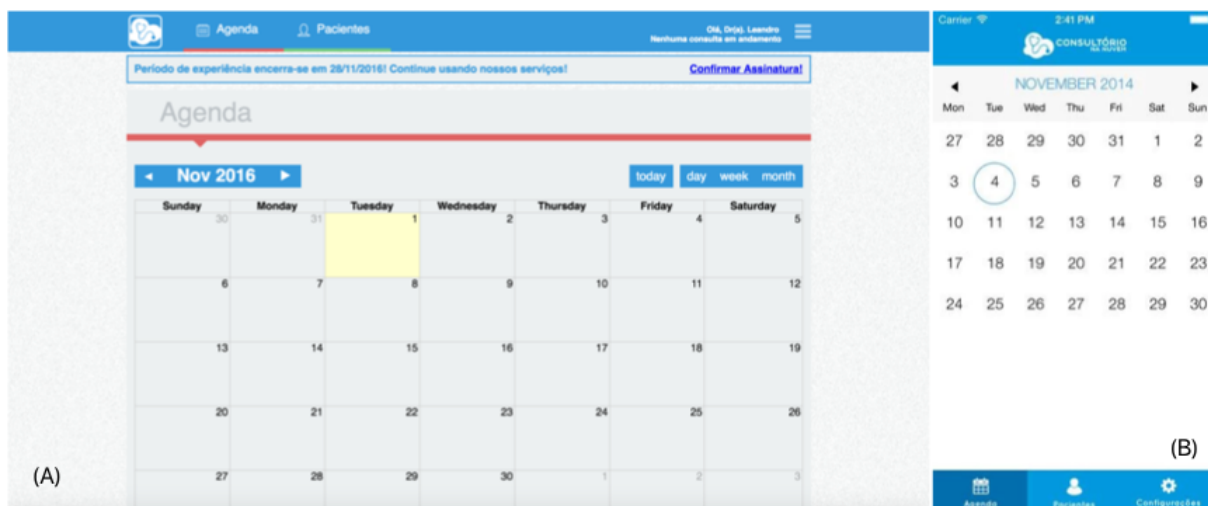


Figura 4: Interfaces do sistema Consultório na Nuvem. (A) Versão *web*, (B) Versão *Mobile*. Fonte: [26]

2.2.4 Saúde na Nuvem

O Saúde na Nuvem é um sistema para armazenamento online de informações médicas. Por meio desta aplicação, é possível agendar consultas, acessar uma linha do tempo com todas as informações adicionadas ao sistema, como histórico de exames e vacinas, e também listar os médicos consultados.

Atualmente o sistema possui apenas a versão *web*, a qual aparenta ainda estar em fase de desenvolvimento, pois apresenta deficiência de informações e alguns links que não funcionam. Este sistema também dispõe de uma área de comunicação entre médico e paciente a qual não foi possível o acesso ao realizar testes. A interface desta aplicação é apresentada pela Figura 5.

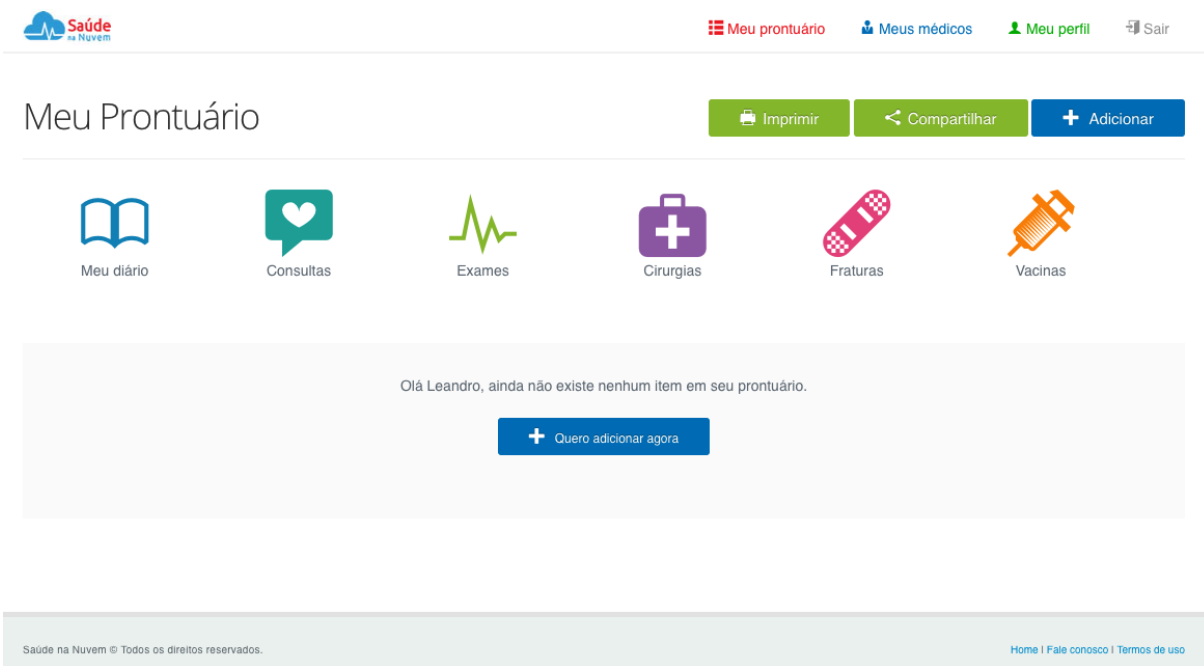


Figura 5: Interface do Sistema Saúde na Nuvem. Fonte: [27]

2.3 Análise do Estado da Arte

A Tabela 1 apresenta um comparativo entre todos os sistemas descritos na seção 2.1. A partir dos dados previamente expostos e da análise da Tabela 1, conclui-se que não existe um sistema que foque na interação médico-paciente, ou seja, que possui duas interfaces distintas, sendo uma para profissionais da área de saúde e a outra para pacientes. Também pode-se concluir que estes sistemas não possuem uma ferramenta ou função que facilite o contato entre o profissional e o paciente, como um chat voltado para a direta comunicação entre ambos.

Atualmente, encontra-se disponível um número superior de sistemas voltados aos profissionais da saúde do que aos pacientes, sendo que os sistemas de maior qualidade são focados para clínicas ou médicos. Este fato está relacionado com a monetização dos produtos, visto que os sistemas na área da saúde são mais facilmente adquiridos por profissionais da área do que por pacientes.

Sistemas	Plataforma <i>web</i>	Plataforma <i>Mobile</i>	Interface Paciente	Interface Médico	Chat Médico Paciente
Clínica nas Nuvem	X			X	
Saúde Controle	X	X	X		
Consultório na Nuvem	X	X		X	
Saúde na Nuvem	X		X		
FaleMed	X	X	X	X	X

Tabela 1: Comparativo entre os Sistemas

Todas as tecnologias descritas na Seção 2.2 foram previamente analisadas com a finalidade de se escolher as mais adequadas para o desenvolvimento do sistema proposto

neste trabalho. Para isto, considerou-se o grau de desenvolvimento e atualização destas tecnologias, e também toda a documentação já existente, sendo ela formal ou informal.

3 MATERIAS E MÉTODOS

Este capítulo apresenta os materiais e métodos utilizados para o desenvolvimento do sistema proposto. Realizou-se a combinação das metodologias ágeis *Scrum* e *Mobile-D* com a finalidade de organizar e facilitar o desenvolvimento do projeto.

3.1 *Mobile-D*

Mobile-D é um conceito de metodologia ágil desenvolvido para aplicações mobile[28], e é definido em cinco fases: *Explore*, *Initialize*, *Productionize*, *Stabilize* e *System Test & Fix*. Cada fase possui um número associado de estágios, tarefas e práticas. As especificações completas do método estão disponíveis em *VTT Eletronics*³[29].

Na primeira fase, *Explore*, a equipe de desenvolvimento deve gerar um plano e definir as características do projeto, o que é feito em três etapas: estabelecimento do cliente (os clientes que tomam parte ativa no processo de desenvolvimento), o planejamento inicial do projeto e coleta de requisitos, e o estabelecimento de processos.

Na próxima fase, *Initialize*, a equipe de desenvolvimento e todas as partes interessadas finalizam o conceito do produto que está em desenvolvimento e prepararam os recursos essenciais necessários para as atividades de produção, tais como recursos físicos, tecnológicos e de comunicação. Esta fase é dividida em três etapas: projeto de set-up, planejamento inicial e dia de teste[30].

A fase *Productionize* compreende principalmente as atividades de implementação. Ao final desta fase, a maior parte da aplicação deve estar completa. Esta fase é dividida em dias de planejamento, dias de trabalho e dias de lançamento. Os dias de planejamento destinam-se a reforçar o processo de desenvolvimento, priorizando a análise de requisitos, planejamento dos conteúdos de iteração, e criação de testes de aceitação que serão executados no dia do lançamento.

O *Test-Driven Development* (TDD) é um processo de desenvolvimento de software que se baseia na repetição de um ciclo de desenvolvimento muito curto onde os requisitos são transformados em casos de teste muito específicos, e em dias atuais ele é usado para implementar funcionalidades, de acordo com o plano pré-estabelecido para a iteração atual. Usando TDD junto com Integração Contínua, que é a prática de juntar todo o código desenvolvido em uma linha principal compartilhada várias vezes ao dia, os desenvolvedores criam testes de unidade, escrevem códigos que passam os testes, e integram o novo código com a versão já existente do produto, abordando quaisquer erros que possam surgir no processo de integração. Finalmente, em dias de lançamento, uma versão de trabalho do sistema é produzido e validado através de testes de aceitação[30].

³<http://agile.vtt.fi/mobiled.html>

As duas últimas fases, *Stabilize e System Test & Fix*, são utilizados para a finalização do produto e testes, respectivamente. Eles compreendem estágios semelhantes à fase *Productionize*, com algumas modificações para acomodar a criação da documentação e teste do sistema[30].

Na Figura 6 é possível visualizar todos os processos envolvidos na metodologia *Mobile-D*.

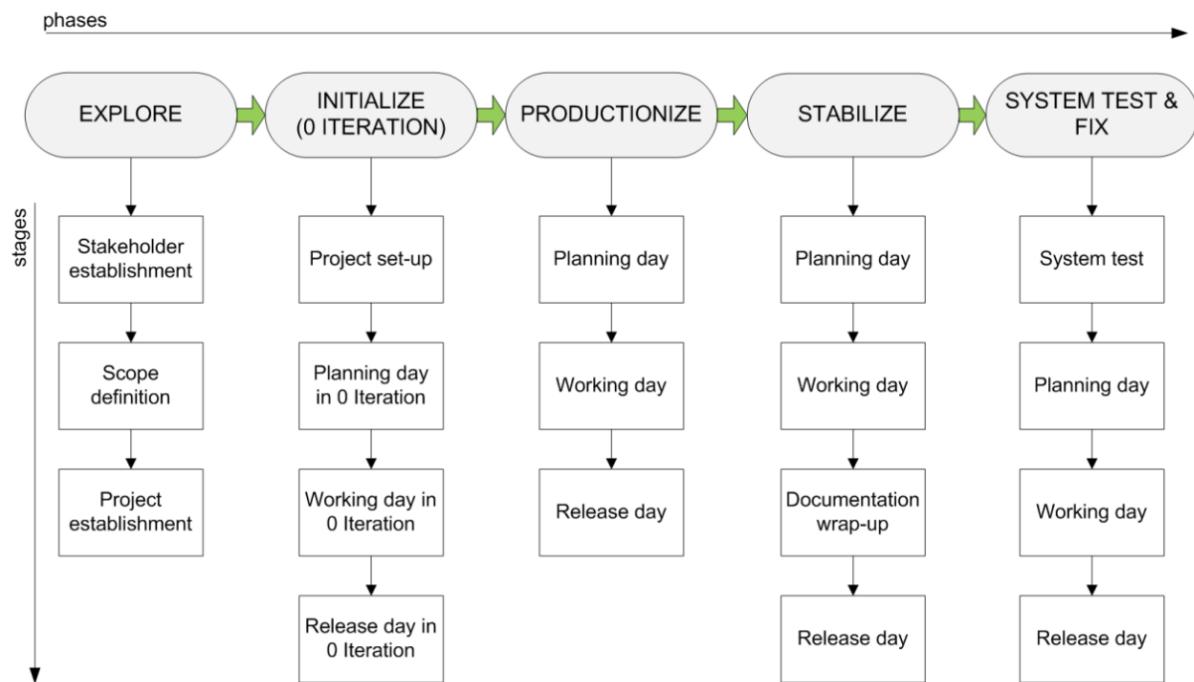


Figura 6: Fases e Estágios do *Mobile-D*. Fonte: [30]

3.2 Scrum

Scrum é uma metodologia ágil para gestão e planejamento de projetos de software que busca diminuir a complexidade dos processos de desenvolvimento, para facilitar a construção de softwares que atendam às necessidades dos negócios e das empresas[31]. No *Scrum*, os projetos são divididos em ciclos de tamanhos fixos denominados *Sprints*, que possuem conjuntos de atividades que devem ser executadas em um determinado tempo[32].

Inicialmente os requisitos são obtidos em reuniões com toda a equipe de desenvolvimento e o cliente, este conjunto de requisitos criado é denominado *Product Backlog*. A partir daí são feitas reuniões para priorizar os itens do *Product Backlog* e dividir as atividades entre as equipes, que serão realizadas durante o *Sprint*. As tarefas alocadas em um *Sprint* são transferidas do *Product Backlog* para o *Sprint Backlog*. A cada dia de uma *Sprint*, a equipe realiza breves reuniões com o objetivo de disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que

se inicia. Ao final de um *Sprint*, a equipe apresenta as funcionalidades implementadas, e então faz-se uma retrospectiva do que foi feito e a equipe parte para o planejamento do próximo *Sprint*. Desta forma, reinicia-se o ciclo[32].

3.3 Etapas da Metodologia

E1 - Levantamento de Requisitos do Sistema: A princípio promoveu-se o levantamento dos requisitos funcionais que o sistema deve conter. Para isto, realizou-se entrevistas com profissionais da área da saúde e pacientes, sendo 2 profissionais, um otorrinolaringologista e um endocrinologista (ambos possuem clínicas na cidade de João Pessoa), e 5 pacientes, entre homens e mulheres com faixa etária de 25 à 55 anos, utilizando um questionário com possíveis funções para a aplicação. Por meio das respostas dos questionários foi criado um *Product Backlog* que satisfaz a metodologia *Scrum* e a primeira fase da *Mobile-D*.

E2 - Estudo e definição das tecnologias a serem utilizadas: Estudo das linguagens que foram aplicadas, dos *frameworks*, definição do banco de dados e estabelecimento das ferramentas utilizadas para o desenvolvimento.

E3 - Idealização e Prototipagem da aplicação: Realizou-se o projeto da aplicação por meio do desenvolvimento de artefatos que representam as telas do sistema. Para isto, a prototipagem das telas foi realizada através da utilização do software *Sketch 3*, a arquitetura da aplicação foi desenvolvida com o auxílio de um editor de desenho, e a modelagem do banco de dados foi feita por meio da ferramenta *DbSchema*.

E4 - Desenvolvimento e Testes da aplicação: Após a idealização do projeto com a criação dos artefatos, deu-se início à fase de implementação e testes do sistema. O desenvolvimento foi realizado em *Product Backlog*, considerando a prioridade de cada funcionalidade. Os testes ocorreram constantemente.

E5 - Validação dos requisitos implementados e correção de erros: Os requisitos funcionais presentes no sistema foram validados pelos próprios usuários que os propuseram na fase de entrevistas, todos os erros encontrados por eles foram corrigidos, e todas as sugestões foram acatadas.

Na Figura 7 é possível visualizar o organograma das etapas da metodologia.

As atividades podem ser visualizadas na Tabela 2, o qual apresenta o cronograma de execução do corrente projeto.

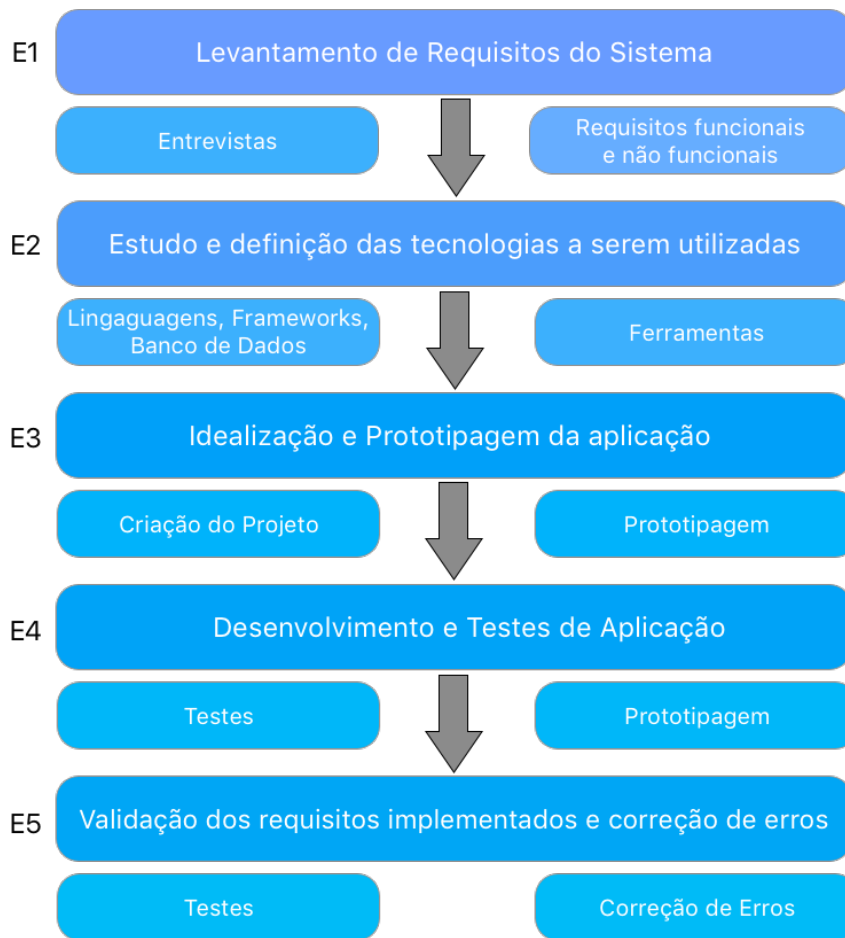


Figura 7: Organograma das Etapas da Metodologia

Atividades	Jul	Ago	Set	Out	Nov
Revisão Bibliográfica	X	X	X	X	X
Levantamento de requisitos	X	X			
Idealização e Prototipagem		X	X		
Desenvolvimento e Testes		X	X	X	X
Validação dos Requisitos			X	X	X
Análise dos Resultados				X	X
Revisão da Monografia				X	X
Defesa do TCC					X

Tabela 2: Cronograma de Atividades

4 DESENVOLVIMENTO

Este capítulo apresenta as funcionalidades levantadas e os artefatos de projeto gerados para a materialização do sistema desenvolvido, o FaleMed.

O sistema desenvolvido, FaleMed, é voltado para a área de saúde, e tem o intuito de facilitar a comunicação entre médico e paciente, além de armazenar todo o acervo médico de um usuário em uma nuvem, criando assim um histórico de exames e médicos visitados. Para isto, é necessário que os usuários, pacientes e/ou profissionais da área de saúde, realizem o cadastro no sistema, via aplicação *mobile*, com *e-mail*, usuário e senha. Portanto, após a realização do cadastro, o usuário tem acesso ao sistema por meio de seu usuário e senha.

O sistema possui dois perfis, sendo um deles voltado para os profissionais da área de saúde e o outro voltado para pacientes. No caso da versão *mobile* deste sistema, com o intuito de diminuir o tamanho do arquivo final gerado e evitar erros, criou-se duas aplicações, uma para os médicos e a outra para os pacientes. Durante a primeira utilização do aplicativo voltado aos profissionais da saúde, além do cadastro, deve ocorrer o registro das suas informações profissionais, como, por exemplo, especialidade, credencial médica, clínicas de atendimento, entre outros. No aplicativo para pacientes, não é obrigatória a adição de informações extras, apenas recomendada.

Após o cadastro de todas as informações, ambos os perfis terão acesso ao *feed* de notícias que tem o propósito de ser utilizado como forma de divulgação de novidades na área da saúde e/ou informações médicas importantes, como, por exemplo, a alteração em horários de atendimento. A inserção de informações ao *feed* de notícias apenas poderá ser feito por profissionais da área de saúde e administradores do sistema.

O perfil do paciente tem acesso a funções que permitem salvar informações adicionais sobre sua saúde em uma ficha clínica, adicionar e visualizar exames, consultas, receitas médicas, vacinas e seu acervo médico, procurar e adicionar médicos à sua conta, iniciar conversas com os profissionais já adicionados, compartilhar informações como documentos ou sua ficha clínica com médicos da sua rede, agendar consultas e realizar pesquisas de médicos por proximidade.

Por outro lado, o perfil dos profissionais da saúde apresenta funções que permitem cadastrar funcionários da sua clínica, enviar documentos para pacientes já adicionados à rede e visualizar exames compartilhados pelos mesmos, além de enviar convite para cadastro de novos pacientes e adicionar informações no *feed* de notícias.

A aplicação *web* é voltada apenas para o perfil dos profissionais da saúde, com o motivo de facilitar o acesso e melhor visualização dos documentos e imagens do paciente durante a consulta. Esta versão mantém apenas as funções de *login*, cadastro de pacientes

e funcionários e cadastro de documentos.

Como o sistema a ser desenvolvido necessita do armazenamento dos dados, foi feita então a persistência dos mesmo no *MongoDB*, como citado na Seção 2.2.4. *MongoDB* é um sistema *NoSQL* que é um termo genérico para uma classe definida de banco de dados não-relacionais, assim não é possível criar um diagrama entidade-relacionamento para representar esse banco de dados, ao invés disso, é recomendado utilizar o diagrama de classes que pode ser visto na Figura 8.

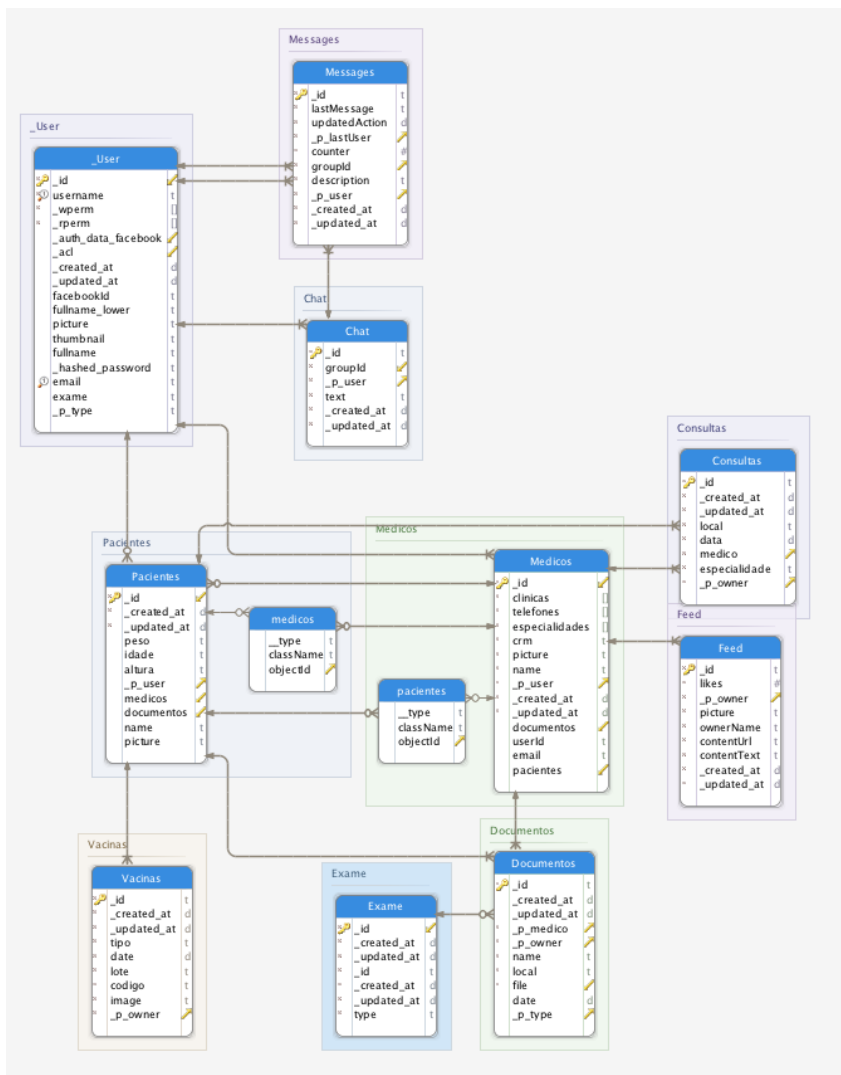


Figura 8: Diagrama de Classes

A arquitetura do sistema é definida na Figura 9, onde podemos observar o funcionamento da comunicação entre o cliente e o servidor, que ocorre através do uso das SDKs disponibilizadas pelo Parse[33]. Os clientes do sistema acessam a *web* API do Parse através de *login* existente no Parse SDKs que necessita de um usuário e senha cadastrados no sistema.

As requisições HTTP (*POST*, *PUT*, *GET*, *DELETE*) enviadas pelos cliente, via

SDK, são tratadas pelo *Parse Server* e retornam um objeto em formato JSON.

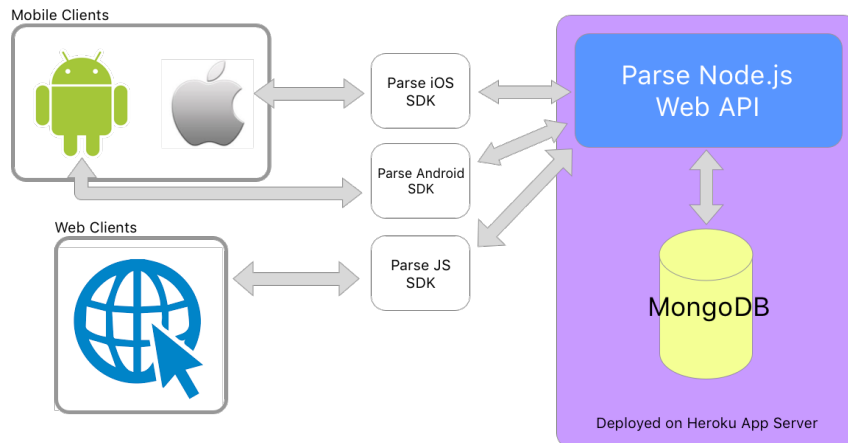


Figura 9: Arquitetura do FaleMed

4.1 Resultados

Nesta Seção serão apresentadas as funcionalidades implementadas na aplicação e as interfaces produzidas durante o processo de desenvolvimento do sistema e suas interações.

4.1.1 Interfaces

A partir do esboço apresentado na Figura 10, toda a interface foi criada através do *Sketch 3*, software voltado para designers e exclusivo para sistemas Mac OSX [34].

Todas as interfaces da versão iOS foram criadas de acordo com o *iOS Human Interface Guidelines*, que reúne todos os princípios e padrões de design que devem existir em toda aplicação iOS[35]. Das Figuras 15 à 21 são apresentadas as telas da aplicação para pacientes e as interfaces *web*. No Apêndice A é possível encontrar as interfaces da aplicação para o profissional. As duas aplicações diferem apenas no cadastro, no histórico médico, o qual o paciente possui e o profissional não, e na lista de médicos, que para o profissional será lista de pacientes.

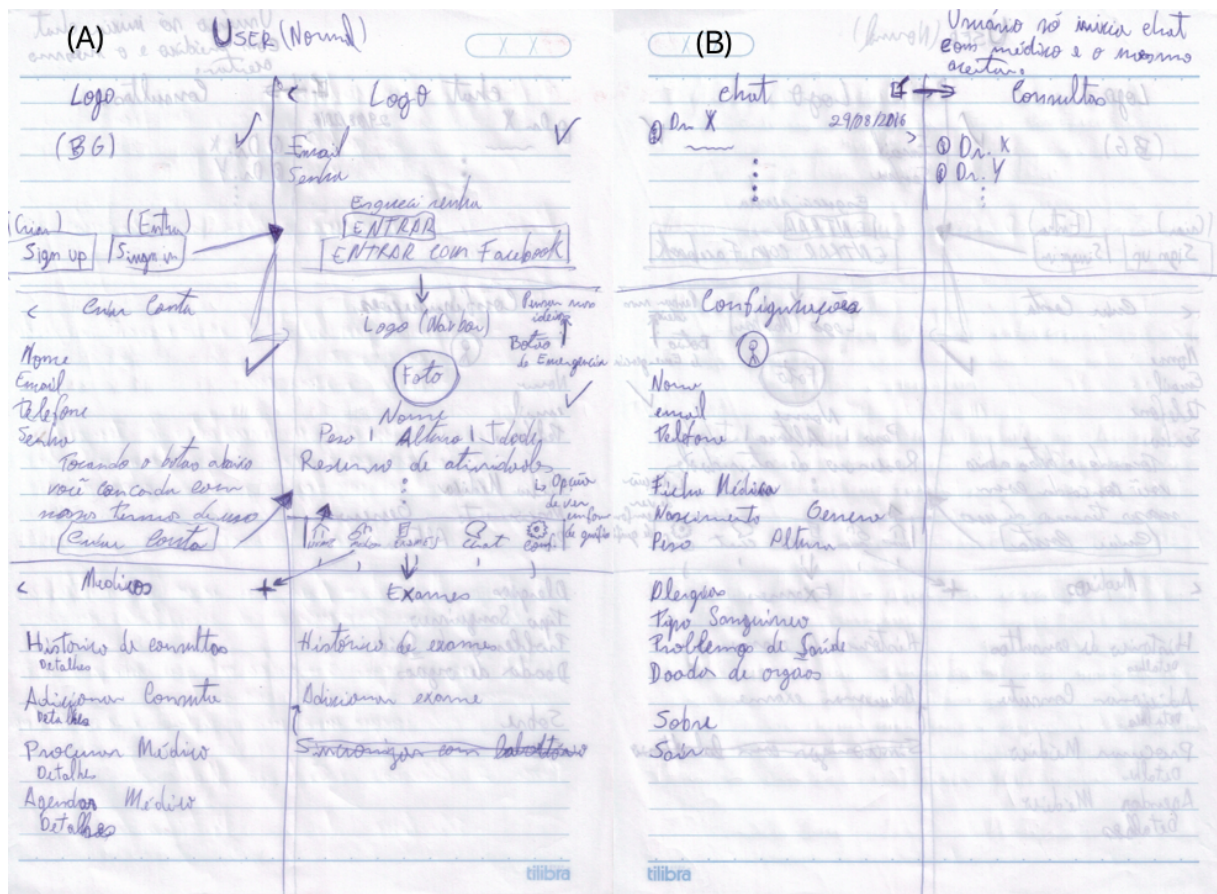


Figura 10: Prototipo de Telas. (A) Primeira Sequência de telas, (B) Segunda Sequência de telas

Na Figura 11, é possível visualizar a tela de *login* para pacientes, a qual possui apenas os dois campos necessários para acessar o sistema: usuário e senha. Nesta mesma tela também existem as opções para recuperação de senha e para cadastro. O botão de cadastro redireciona o novo usuário para uma tela onde ele poderá efetuar o seu registro, por meio do preenchimento dos itens “Nome completo”, “email” e “senha”, como é mostrado na Figura 12. É importante ressaltar que a tela de cadastro foi projetada para ser relativamente simples, com a finalidade de facilitar o acesso ao sistema por novos usuários.

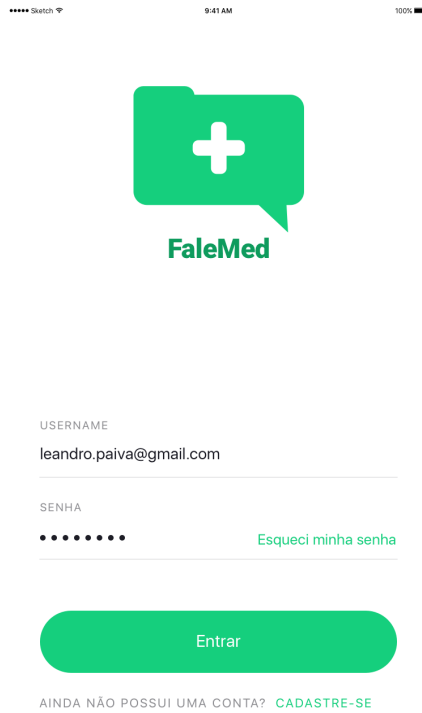


Figura 11: Tela de *login*

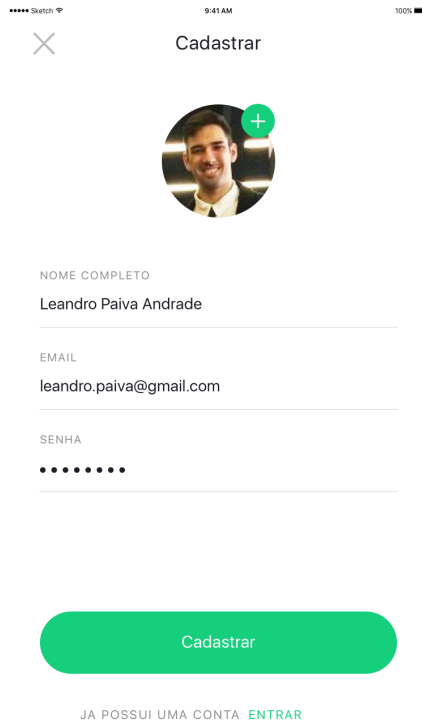


Figura 12: Tela de Cadastro

A Figura 13 apresenta a tela principal da aplicação na versão paciente, na qual existe um *feed* de notícias onde os profissionais da saúde e administradores podem postar

notícias e informações importantes. No canto superior esquerdo da tela há um botão que permite o acesso a um menu lateral, enquanto que na parte inferior existe uma *tab bar* com opções de acesso à área de médicos, mensagens, histórico e também às configurações.



Figura 13: Tela de Home

Na Figura 14, é possível visualizar a tela da aplicação que lista os profissionais amigos do usuário. Nesta mesma tela também é possível realizar uma busca por outros profissionais que já são cadastrados no sistema. Caso o usuário não obtenha sucesso em sua busca, o mesmo pode enviar um convite para que o médico de interesse realize seu registro no sistema. O envio do convite é feito clicando-se no botão localizado no canto superior direito da tela.

Caso o usuário encontre o médico de interesse por meio do recurso de busca, ele poderá acessar o perfil do profissional, que é apresentado como na Figura 15. No perfil do profissional deverão estar contidas informações como locais de atendimento, telefones e email para contato, além da opção de se iniciar um chat no próprio aplicativo.

Na Figura 16 é possível verificar a área de mensagens, onde se encontram todas as conversas iniciadas entre profissionais e seus pacientes dentro da aplicação. As conversas são ordenadas de acordo com a data de envio das mensagens, de forma que a conversa que contém a mensagem enviada mais recentemente fica localizada no topo da lista, e assim sucessivamente.

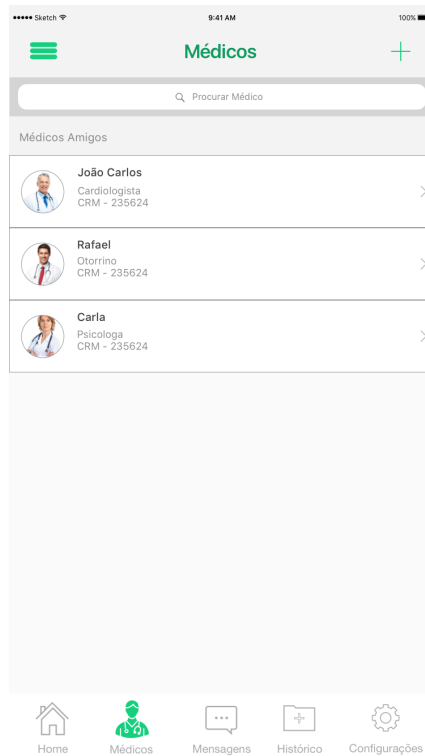


Figura 14: Tela de lista de Profissionais da Saúde

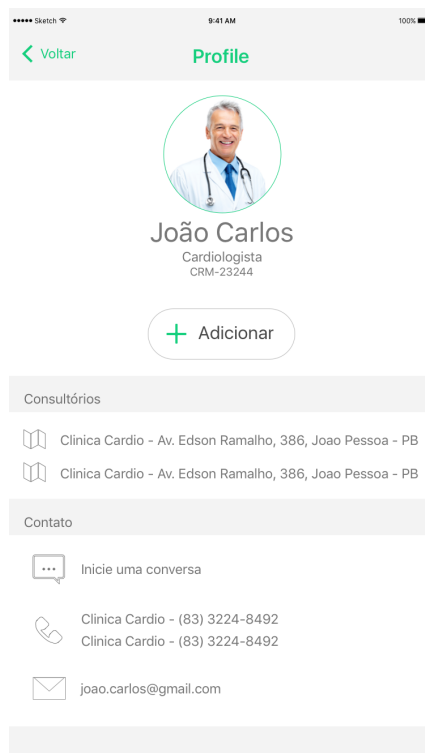


Figura 15: Tela de Perfil do Profissional

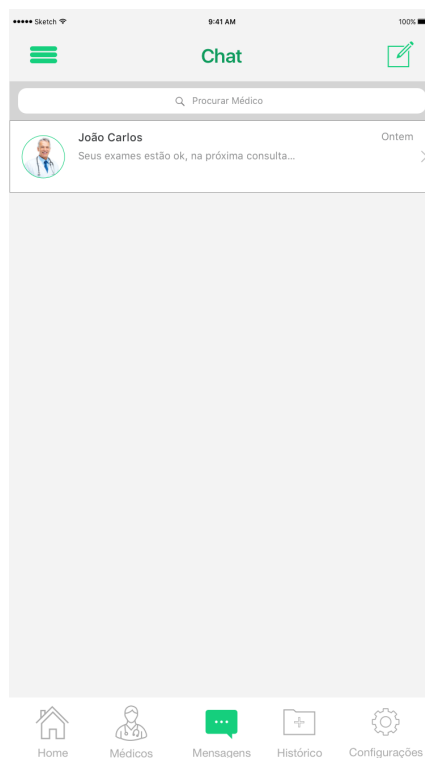


Figura 16: Tela de lista de Conversas

A Figura 17 contém as telas (A), (B) e (C), que mostram o histórico médico do paciente. Quando o botão “Histórico” da *tab bar* é acionado, o usuário é direcionado para a tela (A), a qual possui o seu acervo de documentos médicos, que estão subdivididos em categorias: “Exames”, “Consultas”, “Receitas Médicas” e “Vacinas”. Ao selecionar a categoria “Exames”, o paciente é então redirecionado para a tela (B), onde é possível escolher o tipo de exame que se deseja acessar, como, por exemplo, “Exames de Sangue”, “Radiografias”, “Ultrassonografias”, entre outros. Após a escolha do tipo de exame, o usuário terá acesso à tela (C), na qual estão contidos todos os exames do tipo selecionado que foram previamente anexados ao sistema, ordenados cronologicamente. É possível anexar novos documentos médicos ao histórico por meio do botão de adição, localizado no canto superior direito.

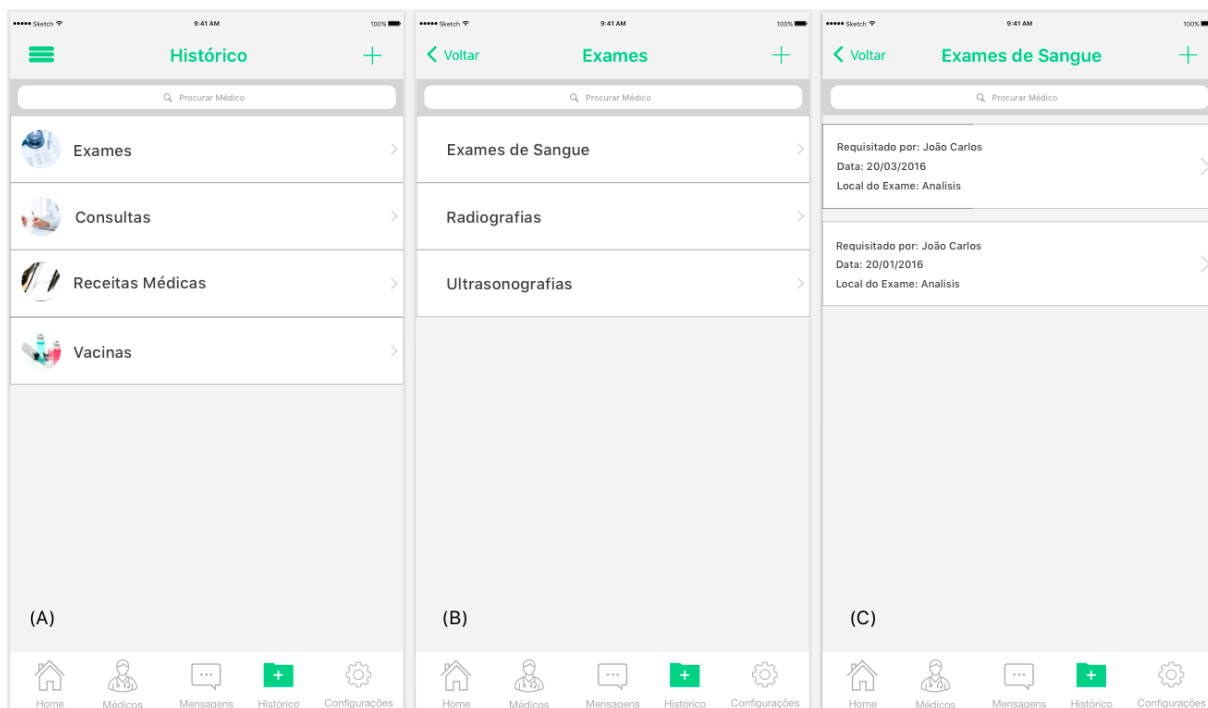


Figura 17: Tela de lista do Histórico Médico. (A) Tela de Histórico, (B) Tela de Exames e (C) Tela de Documentos Salvos

A Figura 18 corresponde à área de configurações da aplicação. O acesso à esta tela possibilita a edição de informações pessoais como nome, peso, idade e altura. Além das funções de alteração de cadastro, o usuário também é capaz de reportar qualquer problema encontrado durante a utilização do aplicativo, e solicitar ajuda quanto ao seu uso, além de verificar informações sobre o sistema, e também sair da conta.

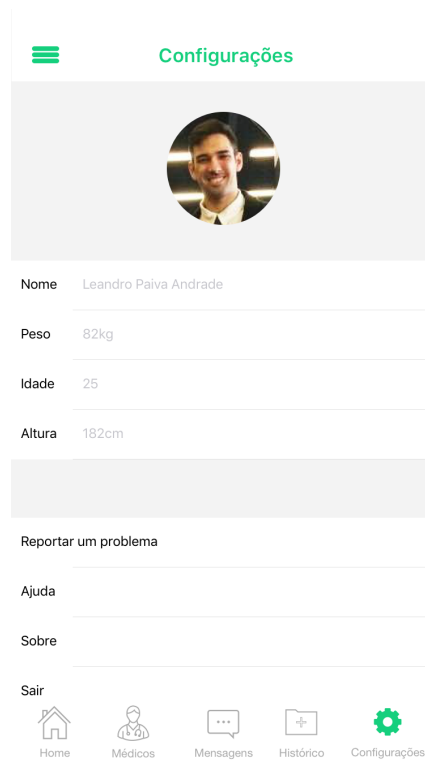


Figura 18: Tela de Configurações

A interface *web* foi desenvolvida por meio da utilização do *Bootstrap*, que é um *framework front-end* responsivo baseado e *open source*. Escolheu-se o *Bootstrap* como *front-end* pois ele permite a responsividade em aplicações, ou seja, a capacidade do site ou sistema se adaptar a diferentes tipos de resolução existentes, sejam elas de celulares, tablets ou computadores e sua grande documentação formal e informal.

Esta interface foi desenvolvida apenas para o perfil dos profissionais da saúde, como já dito anteriormente.

A Figura 19 apresenta a tela de *login web*, onde não é possível se registrar no sistema, pois o registro é apenas realizado na aplicação *mobile*.

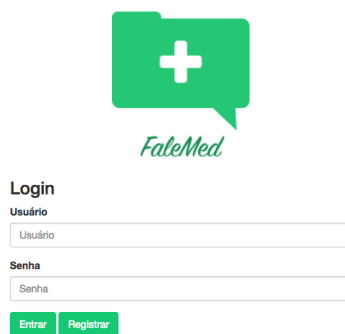


Figura 19: Tela de *login web*

A Figura 20 mostra a tela inicial da versão *web*. Esta interface é simples e com funções específicas para facilitar o acesso às informações, como cadastrar pacientes e funcionários, buscar pacientes, deletar pacientes e acessar informações do paciente.

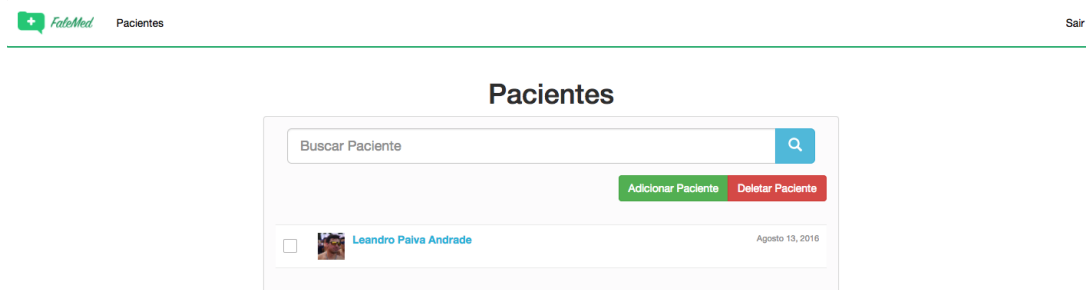


Figura 20: Tela Inicial *web*

A Figura 21 contém as informações do paciente selecionado na Figura 20, como nome, endereço, *email* e telefone. Também é possível visualizar nesta interface o histórico

de documentos médicos adicionados pelo profissional ou compartilhados pelo paciente. Apenas é possível deletar exames que foram adicionados pelo próprio profissional.

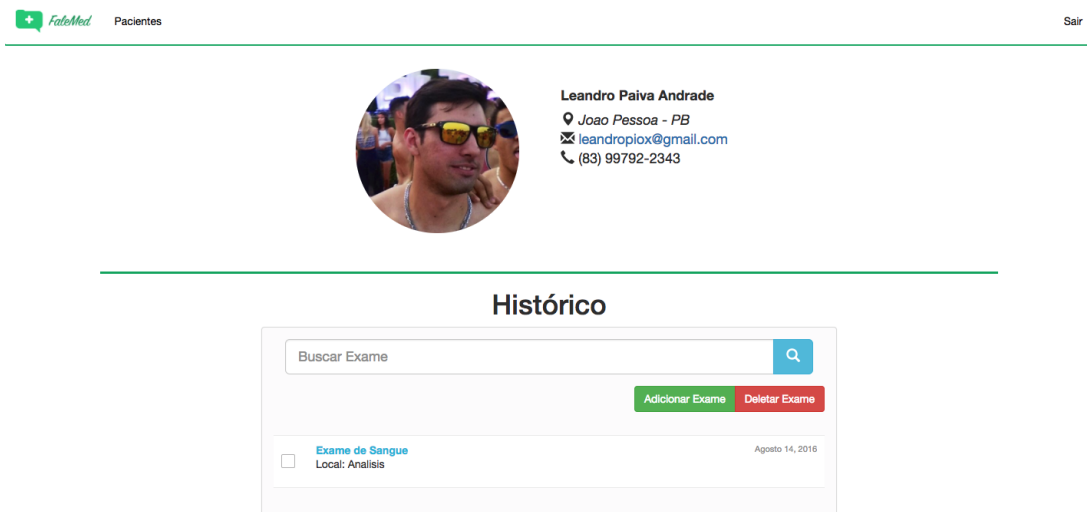


Figura 21: Tela de Pacientes *web*

A Figura 22 corresponde às informações do documento selecionado na tela anterior. É possível visualizar informações mais detalhadas do documento, como data, local onde foi realizado, médico que requisitou o exame e o próprio exame em si.

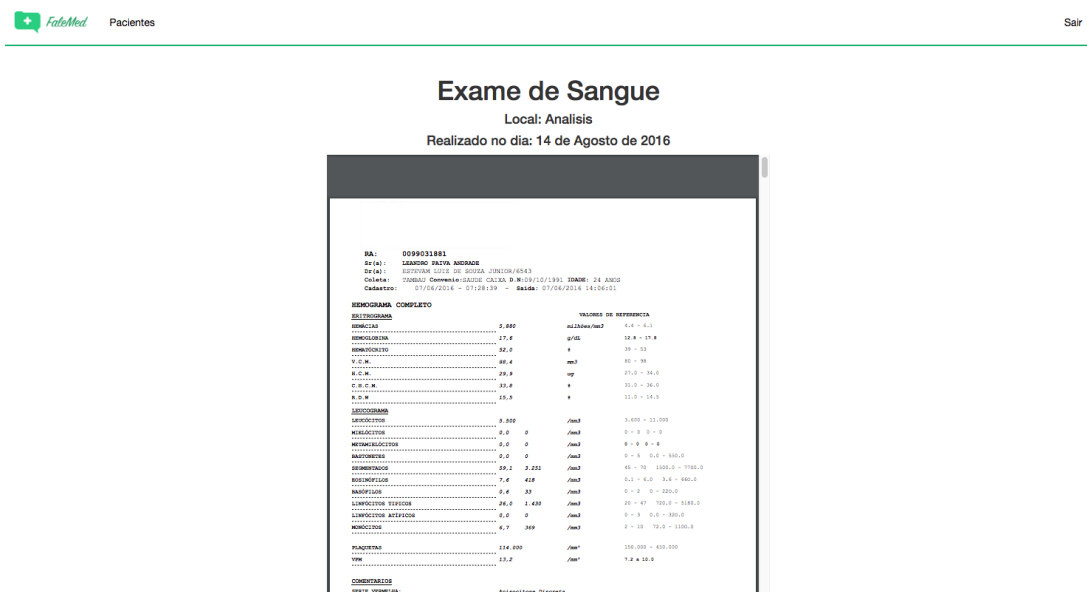


Figura 22: Tela de Detalhes de Documentos *web*

5 CONCLUSÃO E TRABALHOS FUTUROS

As tecnologias usadas para o desenvolvimento do sistema atenderam às expectativas. Devido a fatores como a documentação pesquisada e estudada, a comunidade online das tecnologias escolhidas para este projeto e a escolha das duas metodologias ágeis SCRUM e Mobile-D, foi possível gerenciar e desenvolver o sistema proposto utilizando as melhores práticas, desde a criação das interfaces até a codificação e hospedagem da API.

A aplicação FaleMed foi criada e desenvolvida com sucesso como um sistema para facilitar a interação entre profissionais da saúde e seus pacientes. Além de oferecer uma série de vantagens no sentido de tornar mais prática e ágil esta interação, a aplicação também oferece um sistema de rede social, com o objetivo de distribuir mais informações importantes relacionadas à saúde. Através desta rede é possível adicionar profissionais como amigos e acompanhar um *feed* de notícias alimentado por estes profissionais cadastrados que queiram compartilhar informações importantes como dicas de saúde e avisos sobre consultas.

Uma proposta para trabalhos futuros é o desenvolvimento de novas versões da aplicação para diferentes tipos de dispositivos móveis, como também a adição de mais funções, como o agendamento de consultas médicas pelo próprio aplicativo, e a adição de exames diretamente por laboratórios e hospitais. Outra proposta é expandir a versão web para o perfil de pacientes, com todas as funções que a versão mobile possui. Também para trabalhos futuros será feita a adição dos termos de uso e de privacidade dos usuários, que será criado no código de ética médica brasileiro e políticas padrões de privacidade para gerenciamento de dados.

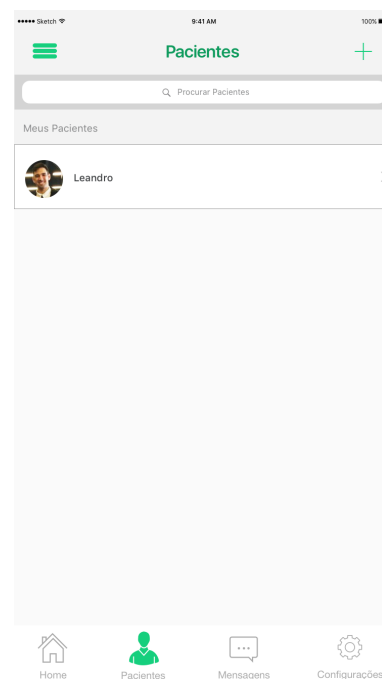
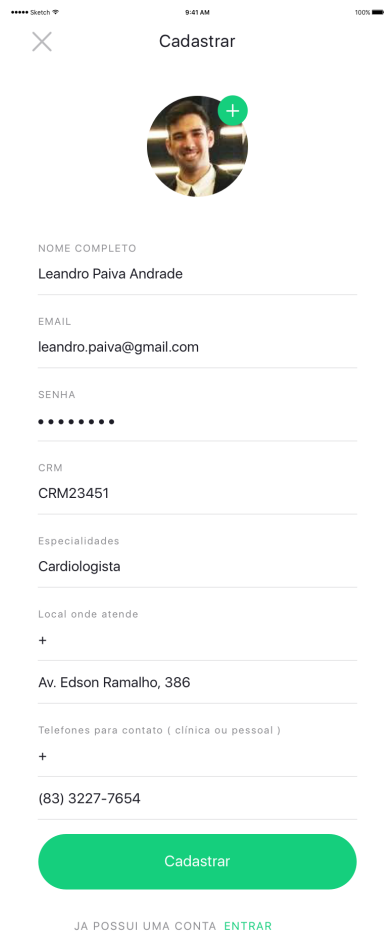
REFERÊNCIAS

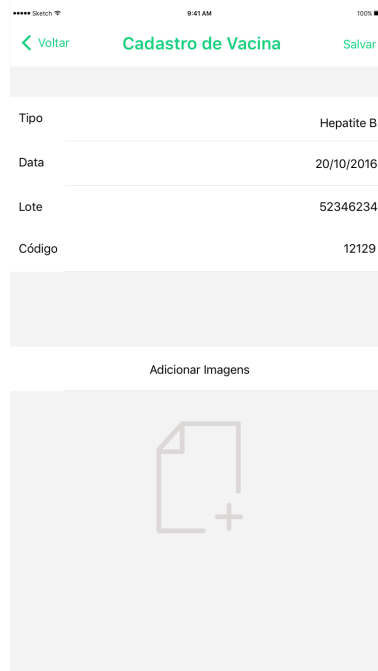
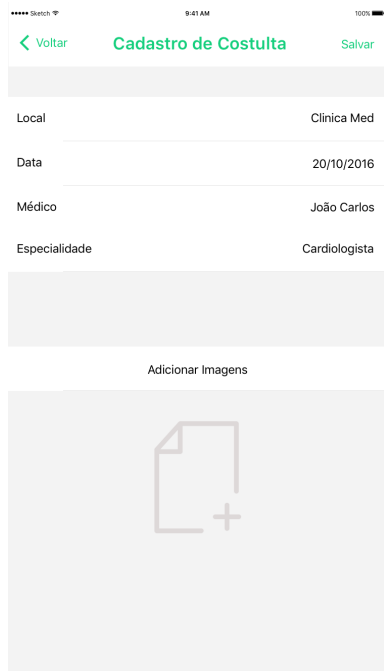
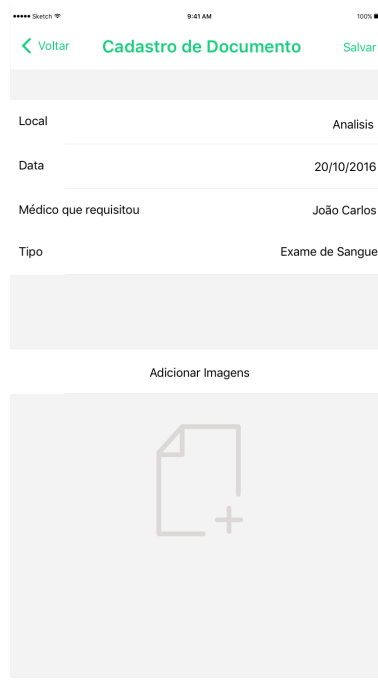
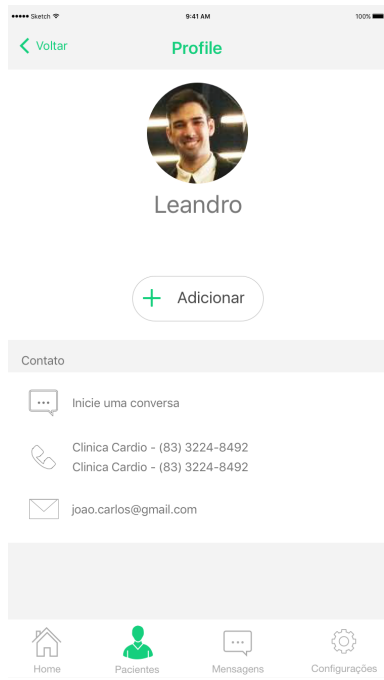
- [1] Coordenação de Trabalho e Rendimento IBGE. *Acesso à Internet e à Televisão e posse de Telefone Móvel Celular para Uso Pessoal*. IBGE, 2016.
- [2] Simon Khalaf. **Health and Fitness Apps Finally Take Off, Fueled by Fitness Fanatics**. Disponível em: <<http://flurrymobile.tumblr.com/post/115192181465/health-and-fitness-apps-finally-take-off-fueled>>, Acesso em: 23 de Novembro de 2016.
- [3] Sasan Adibi. *Mobile health: a technology road map*, volume 5. Springer, 2015.
- [4] Robert Istepanian, Swamy Laxminarayan, and Constantinos S Pattichis. *M-health*. Springer, 2006.
- [5] Robert SH Istepanian and Jose C Lacal. **Emerging mobile communication technologies for health: some imperative notes on m-health**. In *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, volume 2, pages 1414–1416. IEEE, 2003.
- [6] WWDC. **APPLE EVENTS - SPECIAL EVENT JUNE 2014**. Disponível em: <<http://www.apple.com/apple-events/june-2014/>>, Acesso em: 28 de Outubro de 2016.
- [7] Apple Inc. **The Swift Programming Language**. Apple Inc, 2014.
- [8] Apple Inc. **Xcode Overview**. Disponível em: <https://developer.apple.com/library/prerelease/content/documentation/Tools-Languages/Conceptual/Xcode_Overview/index.html>, Acesso em: 28 de Outubro de 2016.
- [9] Fosco Marott. **What is Parse Server?** Disponível em: <<http://blog.parse.com/announcements/what-is-parse-server/>>, Acesso em: 28 de Outubro de 2016.
- [10] Techopedia. **What is Backend-as-a-Service (BaaS)? - Definition from Techopedia**. Disponível em: <<https://www.techopedia.com/definition/29428/backend-as-a-service-baas>>, Acesso em: 28 de Outubro de 2016.
- [11] Kevin Lacker. **Moving On**. Disponível em: <<http://blog.parse.com/announcements/moving-on/>>, Acesso em: 28 de Outubro de 2016.
- [12] Heroku Inc. **What is Heroku**. Disponível em: <<https://www.heroku.com/what>>, Acesso em: 28 de Outubro de 2016.

- [13] MongoDB Inc. **The MongoDB 3.2 Manual**. Disponível em: <<https://docs.mongodb.com/manual/>>, Acesso em: 28 de Outubro de 2016.
- [14] Christof Strauch, Ultra-Large Scale Sites, and Walter Kriha. **NoSQL databases. Lecture Notes, Stuttgart Media University**, 2011.
- [15] Maurício Samy Silva. **Construindo sites com CSS e (X) HTML: sites controlados por folhas de estilo em cascata**. Novatec Editora, 2007.
- [16] W3C. **HTML5 Differences from HTML4**. Disponível em: <<https://www.w3.org/TR/html5-diff/>>, Acesso em: 28 de Outubro de 2016.
- [17] W3C. **HTML & CSS**. Disponível em: <<https://www.w3.org/standards/webdesign/htmlcss#whatcss>>, Acesso em: 28 de Outubro de 2016.
- [18] Maurício Samy Silva. **CSS3: desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3**. Novatec Editora, 2011.
- [19] Lucas Mazza. **HTML5 e CSS3: Domine a web do futuro**. Editora Casa do Código, 2014.
- [20] Google Inc. **Angular 2 Documentation**. Disponível em: <<https://angular.io/about/>>, Acesso em: 28 de Outubro de 2016.
- [21] Mohit Bumb. **7 key differences between Angular 1 and Angular 2**. Disponível em: <<http://bindasmonkeys.com/7-key-differences-angular-1-angular-2-code-examples/>>, Acesso em: 28 de Outubro de 2016.
- [22] CAMTWO SISTEMAS. **Clínica nas Nuvem**. Disponível em: <<https://clinicanasnuvens.com.br/>>, Acesso em: 28 de Outubro de 2016.
- [23] SaúdeControle. **Saúde Controle**. Disponível em: <<http://saudecontrole.com.br/>>, Acesso em: 28 de Outubro de 2016.
- [24] Google Play. **Saúde Controle App - Google Play**. Disponível em: <https://play.google.com/store/apps/details?id=br.com.xds.saudecontrole&hl=pt_BR>, Acesso em: 28 de Outubro de 2016.
- [25] iTunes. **Saúde Controle App - iTunes**. Disponível em: <<https://itunes.apple.com/br/app/saudecontrole/id909107562?mt=8>>, Acesso em: 28 de Outubro de 2016.
- [26] AGÊNCIA ÓPERA. **Consultório na Nuvem**. Disponível em: <<http://www.consultorionanuvem.com/>>, Acesso em: 28 de Outubro de 2016.

- [27] Saúde na Nuvem. **Saúde na Nuvem.** Disponível em: <<http://www.saudenanuvem.com.br/>>, Acesso em: 28 de Outubro de 2016.
- [28] Pekka Abrahamsson, Antti Hanhineva, Hanna Hulkko, Tuomas Ihme, Juho Jääli-noja, Mikko Korkala, Juha Koskela, Pekka Kyllönen, and Outi Salo. **Mobile-D: an agile approach for mobile application development.** In *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, pages 174–175. ACM, 2004.
- [29] Andrei Cristian Spataru. **Agile development methods for mobile applicati-ons.** *Master of Science Thesis submitted to Computer Science School of Informatics, University of Edinburgh*, 2010.
- [30] VTT Eletronics. **Modile-D.** Disponível em: <<http://agile.vtt.fi/mobiled.html>>, Acesso em: 28 de Outubro de 2016.
- [31] scrum.org. **What is Scrum?** Disponível em: <<https://www.scrum.org/Resources/What-is-Scrum>>, Acesso em: 28 de Outubro de 2016.
- [32] Desenvolvimentoagil.com.br. **Scrum: metodologia ágil para gestão e planejamento de projetos.** Disponível em: <<http://www.desenvolvimentoagil.com.br/scrum/>>, Acesso em: 28 de Outubro de 2016.
- [33] Parse. **Parse SDKs.** Disponível em: <<https://parseplatform.github.io/#sdks>>, Acesso em: 28 de Outubro de 2016.
- [34] Bohemian BV. **SketchApp.** <<https://www.sketchapp.com/>>, 2016.
- [35] Apple Inc. **iOS Human Interface Guidelines.** Disponível em: <<https://developer.apple.com/ios/human-interface-guidelines/>>, Acesso em: 28 de Outubro de 2016.

APÊNDICE A - - Interfaces aplicação mobile versão profissional





APÊNDICE B - Questionário de Levatamento de Requisitos do Sistema

QUESTIONÁRIO DE LEVANTAMENTO DE REQUISITOS DO SISTEMA FALEMED

Questões para Profissionais da saúde

- 1- Você exige exames para os pacientes?
- 2- Se exige, que tipos de exames?
- 3- Você costuma perguntar do histórico de consulta do paciente?
- 4- Você costuma perguntar histórico de exames do paciente?
- 5- Como é o pós consulta?
- 6- Em caso de dúvida, como o paciente deve proceder?
- 7- Receber documentos de forma virtual relacionado a saúde do seu paciente ajudaria no processo de consulta?
- 8- Quanto tempo em média leva cada consulta?
- 9- Você usaria um Sistema para receber informações do histórico do seu paciente em seu consultório?
- 10- Você faz uso de alguma aplicação web ou para celulares na sua vida profissional?
- 11- Você usaria um chat para conversa com seus pacientes em caso de dúvidas?

Questões para Pacientes

- 1- Você armazena seus documentos médicos virtualmente ou fisicamente?
- 2- Você usa ou usaria alguma aplicação para armazenar esses tipos de documentos?
- 3- Que tipos de documentos médicos você gostaria de armazenar?
- 4- Quando você vai ao médico, você costuma levar seus exames antigos?
- 5- Você sabe seus históricos de exames, consultas e vacinas?
- 6- O pós-consulta com os médicos que você frequenta costuma ser bom?
- 7- Você já precisou tirar dúvidas com o médico após uma consulta?
- 8- Se sim, foi fácil contatar o médico?

APÊNDICE C - Documento de Requisitos

TCC 2016
Graduando: Leandro Paiva Andrade
Orientador: Eudisley Gomes dos Anjos

Documento de Requisitos do Sistema

Versão 1.0

Conteúdo

1. [INTRODUÇÃO](#)
 - 1.1 [VISÃO GERAL DO DOCUMENTO](#)
 - 1.2 [CONVENÇÕES, TERMOS E ABREVIACÕES](#)
 - 1.2.1 [Identificação dos requisitos](#)
 - 1.2.2 [Prioridades dos requisitos](#)
2. [DESCRIÇÃO GERAL DO SISTEMA](#)
 - 2.1 [ABRANGÊNCIA E SISTEMAS RELACIONADOS](#)
3. [REQUISITOS FUNCIONAIS](#)
4. [REQUISITOS NÃO-FUNCIONAIS](#)
5. [LIMITAÇÕES](#)
6. [COMENTÁRIOS](#)

1. Introdução

Este documento especifica os requisitos do sistema XXX, fornecendo aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

1.1 Visão geral do documento

Além desta seção introdutória, as seções seguintes estão organizadas como descrito abaixo.

Seção 2 – Descrição geral do sistema: apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.

Seção 3 – Requisitos funcionais: especifica todos os requisitos funcionais do sistema.

Seção 4 – Requisitos não-funcionais: especifica todos os requisitos não funcionais do sistema.

Seção 5 – Limitações: apresenta regras ou limitações do sistema.

Seção 6 – Comentários: apresenta comentários em relação a produção do documento.

Seção 7 – Referências: apresenta referências para outros documentos utilizados para a confecção deste documento.

1.2 Convenções, termos e abreviações

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

1.2.1 Identificação dos requisitos

A referência a requisitos é feita através do nome do identificador, o tipo (prioridade), o nome do requisito em si (descrição e observações e o sub-requisito (descrição e observações).

Por exemplo, o requisito funcional F-1.Login deve estar descrito na primeira linha da tabela presente na seção “Requisitos Funcionais”. Já o requisito não-funcional NF-1.Acessibilidade deve estar descrito na primeira linha da tabela presente na seção “Requisitos Não Funcionais”

Os requisitos devem ser identificados com um identificador único. A numeração inicia com o identificador [F-1] ou [NF-1] e prossegue sendo incrementada à medida que forem surgindo novos requisitos.

1.2.2 Prioridades dos requisitos

Para estabelecer a prioridade dos requisitos, nas seções 3 e 4, foram adotadas as denominações “essencial”, “importante” e “desejável”.

1. **Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.

2. **Importante** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos recomendados devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.

3. **Desejável** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

2. Descrição geral do sistema

2.1 Abrangência e sistemas relacionados

O sistema XXXX é uma aplicação desenvolvida para a área de saúde com o intuito de facilitar a comunicação médico paciente e levar todo o acervo médico de um usuário para a nuvem, criando assim um histórico de exames e médico visitados. O usuário também é capaz de registrar sua agenda médica de forma a visualizar melhor os dias e horários de suas consultas apenas pelo celular. A idéia principal desta aplicação é simplificar as consultas médicas, dando acesso aos médicos ao histórico de seus pacientes, de forma ágil e fácil apenas utilizando um aparelho celular, e assim podendo se verificar doenças diagnosticadas, exames, receitas, entre outros. Do lado paciente é facilitar o acesso a informação da saúde do mesmo, que muitas vezes possui um excesso de documentos médicos e exames em sua casa, e também para criar uma lista de médicos que frequenta sabendo para qual se consultou, sua especialidade e quando foi.

Se a iniciativa ganhar força e o sistema se propagar, será adicionado uma nova funcionalidade, a de realizar a busca por médicos baseada em localização e o agendamento via aplicação.

O sistema possuirá criptografia em todos os arquivos armazenados, de forma que apenas o usuário da conta onde os arquivos estão armazenados poderá acessar. Caso seja necessário compartilhar documentos com Médicos, o usuário deverá gerar uma permissão de compartilhamento e determinar certas restrições ao compartilhamento, como por exemplo, tempo que o Médico poderá visualizar aquele documento. Não será permitido o compartilhamento de documentos entre usuários Pacientes-Paciente ou Médico-Médico, apenas entre usuários Médico-Paciente.

3. Requisitos funcionais

Descrição dos requisitos funcionais do projeto.

Tipo: 1-Essencial, 2-Recomendado, 3-Desejável

Req #	Tipo	Requisito (Descrições e Observações)	Sub-Requisitos (Descrições e Observações)
-------	------	--------------------------------------	---

F-1	1	Login	Logar-se com Facebook, Gmail ou conta criada no sistema
F-2	1	Cadastro de Conta	Cadastrar-se através de email e nome
F-3	1	Permitir acesso a informações mediante autenticação do usuário	Informações serem Exames, histórico médico, entre outros
F-4	1	Permitir ao paciente visualizar agendamentos marcados para ele.	Visualizar em calendário ou em forma de lista
F-5	2	Enviar uma notificação	Com lembretes de consultas
F-6	1	Permitir ao paciente visualizar resultado dos exames.	Exames ordenados por data ou por médico
F-7	1	Permitir médicos a visualizar exames.	Apenas médicos autorizados pelo responsável dos documentos.
F-8	1	Permitir o cadastro de usuário médico e usuário paciente	Dois tipos de contas com diferentes acessos ou sistema.
F-9	2	Permitir médico cadastrar sub-usuário a sua conta	Cadastro de funcionários do médico
F-10	1	Permitir ao médico a postagem de documentos no usuário paciente.	Documentos como Exames, receitas, laudo médico, entre outros
F-11	1	Permitir ao paciente a postagem de documentos na própria conta.	Documentos como Exames, receitas, laudo médico, entre outros
F-12	1	Histórico médico do paciente.	Paciente poderá registrar suas consultas médicas ou automaticamente será criado o histórico a partir das consultas com os médicos registrados
F-13	1	Setor de chat paciente - médico	Baseado em tags para facilitar

F-14	2	Agendar consultas	Realizar agendamento de consulta via aplicação
F-15	2	Buscar por médicos	Realizar buscas por médicos próximos
F-16	3	Cadastro de pacientes por usuários médicos e seus sub-usuários	Cadastro de pacientes que não possuem conta no sistema
F-17	3	Cancelar Agendamento	Médico, sub-usuários médico e pacientes podem cancelarem um agendamento
F-18	3	Visualizar os horários disponíveis para o agendamento	Auto explicativo
F-19	3	Visualizar agendamentos	Médico e seus sub-usuários podem visualizar uma agenda que relaciona todas as consultas agendadas e os horários disponíveis.
F-20	2	Feed de notícias	Atualizações do paciente e dos médicos amigos.
F-21	2	Permitir Médicos e Administradores adicionar informações ao feed de notícias	Médicos e administradores podem realizar postagens no feed de notícias

4. Requisitos não-funcionais

Req #	Tip o	Requisito	Sub-Requisitos
NF-1	2	Acessibilidade	VoiceOver e TalkBack
NF-2	1	Integridade/Segurança	Criptografia em todos os arquivos armazenados.

NF-3	1	Usabilidade	Teste com Diversos Tipos de Usuários
NF-4	1	Documentação	Requisitos, Descrição Geral, Arquitetura, Manual do Usuário, Comentários em Código.
NF-5	1	Compatibilidade	iOS
NF-6	1	Desempenho	Uso mínimo de memória do iPhone

5. Constraints

Constraint #	Detalhes de Constraint
C-1	Conexão com Internet (Deslogar Caso a Internet não Seja Detectada)
C-2	Funcionara para sistemas iOS 9.x
C-3	Funcionara para sistemas Android 6.X

6. Comentários

#	Date	Editor	Comentário
1	02/08/2016	Leandro	Criação do documento
2	03/08/2016	Leandro	Inserção dos requisitos funcionais e não funcionais
3	05/08/2016	Leandro	Melhorias na Seção 2 (Descrição do Sistema)
4	07/09/2016	Leandro	Revião do Documento
5	14/09/2016	Eudisley	Revisão do documento de requisitos e inserção de comentários para possíveis melhorias
6	21/09/2016	Leandro	Adicionado funcionalidade F-20

7	02/11/2016	Leandro	Alteração do tipos das funcionalidades F-14 e F-15
8	02/11/2016	Leandro	Adicionado funcionalidade F-21